DYNAMIC BAYESIAN NETWORKS FOR INFORMATION FUSION WITH APPLICATIONS TO HUMAN–COMPUTER INTERFACES

BY

VLADIMIR IVAN PAVLOVIC

Dipl., University of Novi Sad, 1991
M.S., University of Illinois at Chicago, 1993

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Electrical Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 1999

Urbana, Illinois

Dynamic Bayesian Networks for Information Fusion with Applications to Human–Computer Interfaces

Approved by
Dissertation Committee:

____

____

____

____

____

ABSTRACT

Recent advances in various display and virtual technologies coupled with an explosion in available computing power have given rise to a number of novel human–computer interaction (HCI) modalities–speech, vision-based gesture recognition, eye tracking, EEG, etc. However, despite the abundance of novel interaction devices, the naturalness and efficiency of HCI has remained low. This is due in particular to the lack of robust *sensory data interpretation* techniques. To deal with the task of interpreting single and multiple interaction modalities this dissertation establishes a novel probabilistic approach based on *dynamic Bayesian networks* (DBNs). As a generalization of the successful hidden Markov models, DBNs are a natural basis for the general temporal action interpretation task. The problem of interpretation of single or multiple interacting modalities can then be viewed as a Bayesian inference task. In this work three complex DBN models are introduced: *mixtures of DBNs*, *mixed-state DBNs*, and *coupled HMMs*. In-depth study of these models yields efficient approximate inference and parameter learning techniques applicable to a wide variety of problems. Experimental validation of the proposed approaches in the domains of gesture and speech recognition confirms the model's applicability to both unimodal and multimodal interpretation tasks.

To Karin

ACKNOWLEDGMENTS

I wish to acknowledge all the people who have given me encouragement and helped me in the completion of this study. I would especially like to express my appreciation and gratitude to my advisor, Dr. Thomas S. Huang, for his support, time, and advice throughout my graduate studies. I also wish to thank my other committee members, Dr. Steven Levinson, Dr. Pierre Moulin, Dr. Michelle Perry, and Dr. Rajeev Sharma, for their constructive ideas and comments. Special thanks are extended to Dr. Brendan Frey for hours of valuable discussions that improved this dissertation.

I wish to thank my family for all their love, support, and encouragement through the good times and bad. Finally, but not least, I thank my wife Karin for all her help, sacrifice, and understanding that made this work possible.

TABLE OF CONTENTS

CHAPTER **PAGE**

1 INTRODUCTION 1
- 1.1 Background 1
  - 1.1.1 Human action modalities for HCI 3
  - 1.1.2 Computer sensing modalities for HCI 3
    - 1.1.2.1 Position and motion sensing 4
    - 1.1.2.2 Audio sensing 4
    - 1.1.2.3 Visual sensing 4
    - 1.1.2.4 Tactile and force sensing 5
    - 1.1.2.5 Neural sensing 5
- 1.2 Motivation 6
  - 1.2.1 Unimodal interpretation 6
  - 1.2.2 Multimodal interpretation 7
- 1.3 Organization 8
- 1.4 Contribution 9

2 BAYESIAN NETWORKS 10
- 2.1 Introduction 10
- 2.2 Inference 12
  - 2.2.1 Bayesian rule inference 12
  - 2.2.2 Exact probability propagation in a singly connected network 13
  - 2.2.3 Approximate inference 14
- 2.3 Learning 15
  - 2.3.1 Learning with hidden variables 16
    - 2.3.1.1 Gradient-based learning 16
    - 2.3.1.2 Expectation-maximization learning 17

3 DYNAMIC BAYESIAN NETWORKS 20
- 3.1 Introduction 20
- 3.2 Model 21
- 3.3 Inference 22
  - 3.3.1 Forward propagation 23
  - 3.3.2 Backward propagation 23
  - 3.3.3 Smoothing 24
  - 3.3.4 Prediction 24
  - 3.3.5 Decoding 24
- 3.4 Learning 25
- 3.5 Linear Dynamic Systems and Continuous-State Dynamic Bayesian Networks 26
  - 3.5.1 Inference 27
    - 3.5.1.1 Forward propagation 27

3.5.1.2 Smoothing 28
3.5.1.3 Decoding 29
3.5.2 Learning 29
3.6 Hidden Markov Models and Discrete-State Dynamic Bayesian Networks 30
3.6.1 Notation 30
3.6.2 Discrete observation HMM 31
3.6.3 Gaussian observation HMM 31
3.6.4 Mixture-of-Gaussians observation HMM 32
3.6.5 A different state-space model 33
3.6.6 Inference 33
3.6.6.1 Forward propagation 33
3.6.6.2 Backward propagation 34
3.6.6.3 Smoothing 34
3.6.6.4 Prediction 34
3.6.6.5 Decoding 35
3.6.7 Learning 35
3.6.8 Bayesian network representation and state transition diagrams 36
3.7 Complex Dynamic Bayesian Networks 37
3.8 Hierarchy of Models 38

4 MIXTURES OF DYNAMIC BAYESIAN NETWORKS 42
4.1 Introduction 42
4.2 Model 43
4.3 Inference 44
4.3.1 Mixture of linear dynamic systems 46
4.3.2 Mixture of hidden Markov models 47
4.4 Learning 48

5 MIXED-STATE DYNAMIC BAYESIAN NETWORKS 50
5.1 Introduction 50
5.2 Model 51
5.3 Inference 54
5.3.1 Decoupled inference 55
5.3.2 Variational inference 56
5.4 Decoding 59
5.5 Learning 61

6 COUPLED HIDDEN MARKOV MODELS 63
6.1 Introduction 63
6.2 Model 64
6.3 Inference 66
6.3.1 Naive inference 66
6.3.2 Variational inference 67
6.3.2.1 Fixed weights model 68
6.3.2.2 Adaptive intermodal weights model 73
6.3.2.3 Adaptive time-varying intermodal weights model 76
6.4 Learning 79

7 ANALYSIS AND RECOGNITION OF HAND GESTURES USING DYNAMIC BAYESIAN NETWORKS . . . 82
7.1 Introduction . . . 82
7.1.1 Definition of gestures . . . 82
7.1.2 Gestural taxonomy . . . 84
7.1.3 Temporal modeling of gestures . . . 85
7.1.4 Spatial modeling of gestures . . . 86
7.1.5 Gesture analysis . . . 86
7.1.5.1 Feature detection . . . 87
7.1.5.2 Model state estimation . . . 88
7.1.6 Gesture recognition . . . 88
7.2 General Dynamic Bayesian Network Gesture Model . . . 89
7.3 Directly Observed Hand Gestures . . . 92
7.3.1 Dynamics of hand motion . . . 92
7.3.2 Inference . . . 94
7.3.3 Parameter learning . . . 95
7.3.4 Initialization . . . 95
7.3.5 Experiment: Analysis and recognition of hand gestures acquired by the computer mouse . . . 96
7.4 Visually Observed Hand Gestures . . . 102
7.4.1 Inference . . . 105
7.4.2 Spatially localized analysis . . . 108
7.4.3 Prediction and on-line inference . . . 108
7.4.4 Learning . . . 109
7.4.5 Experiment: Analysis and recognition of visually perceived hand gestures . . . 111
7.4.5.1 Initial Hand Tracking . . . 112
7.4.5.2 Recognition of gestural actions . . . 114

8 MULTIMODAL ANALYSIS AND RECOGNITION OF HAND GESTURES AND SPEECH USING DYNAMIC BAYESIAN NETWORKS . . . 123
8.1 Introduction . . . 123
8.2 When to Integrate Gestures and Speech . . . 123
8.3 How to Integrate Gestures and Speech . . . 125
8.3.1 A general integration model . . . 125
8.3.2 Feature-level coupling . . . 126
8.3.3 Decision-level coupling . . . 126
8.4 Previous Work . . . 127
8.4.1 Feature-level coupling . . . 127
8.4.2 Decision-level coupling . . . 127
8.5 Experiment: Coupled Interpretation of Hand Gestures and Speech for Display Control . . 129

9 CONCLUSIONS . . . 140
9.1 Future Work . . . 142

APPENDIX A VARIATIONAL INFERENCE THEOREM . . . 144

APPENDIX B MIXTURE OF DYNAMIC BAYESIAN NETWORKS . . . 147

APPENDIX C MIXED-STATE DYNAMIC BAYESIAN NETWORKS . . . 148

APPENDIX D COUPLED HIDDEN MARKOV MODELS 149
D.1 Fixed-Weights Coupled Hidden Markov Model 149
D.2 Adaptive-Weights Coupled Hidden Markov Model 149
D.3 Adaptive Time-Varying Coupled Hidden Markov Weights Model 150

REFERENCES 152

VITA 160

LIST OF TABLES

Table **Page**

7.1 Error estimates [%] and error estimate variances ([%]) for mouse symbol classification. . 100
7.2 Error estimates [%] and error estimate variances ([%]) for noisy mouse symbol classification. 100
7.3 Gestural commands for virtual display control. Dynamic stroke gestural actions involve highly dynamic hand motion. Static stroke actions, on the other hand, exhibit limited hand motion during gesture stroke phases. Two-letter codes in parentheses are used to denote respective gestural actions. 112
7.4 Transcript of training (GVS1) and test (GVS2) gesture video sequences. 116
7.5 Misclassification error estimates [%] and error estimate variances ([%]) for classification of 40 gestural actions from GVS2 data set. 120
7.6 Confusion table of gesture classification with coupled mixture of LDS/HMM model. Columns are original actions, rows are classified actions. 120
7.7 Confusion table of gesture classification with decoupled mixture of LDS/HMM model. Columns are original actions, rows are classified actions. 121
7.8 Confusion table of gesture classification with decoupled nonlinear model. Columns are original actions, rows are classified actions. 121

8.1 Spoken commands for virtual display control. Dynamic and static actions are associated with their respective gestural counterparts. 129
8.2 Misclassification error estimates [%] and error estimate variances ([%]) for classification of 40 audio-visual actions from GVS2 data set. Columns correspond to different classification models: **A** - independent audio, **V** - independent video, **AV1** - decision-level coupled, **AV2** - state level coupled, **CAV1** - coupled with fixed weights, **CAV2** - coupled with adapted time-invariant weights, and **CAV3** - coupled with adapted time-varying weights. 135
8.3 Misclassification error estimates [%] and error estimate variances ([%]) for classification of 40 audio-visual actions from GVS2 data set with 5dB SNR on audio. 135
8.4 Confusion table of audio action classification with an HMM. Columns are original actions, rows are classified actions. 136
8.5 Confusion table of audio/video action classification under decision-level fusion. 136
8.6 Confusion table of audio/video action classification with unimodal HMMs. 136
8.7 Confusion table of audio/video action classification with fixed-weight coupled HMM model. Columns are original actions, rows are classified actions. 137
8.8 Confusion table of audio/video action classification using coupled HMM with variable time-invariant weights. 137
8.9 Confusion table of audio/video action classification using coupled HMM with variable time-varying weights. 137
8.10 Confusion table of noisy audio action classification with an HMM. 138
8.11 Confusion table of noisy audio/video action classification under decision-level fusion. . . . 138
8.12 Confusion table of noisy audio/video action classification with unimodal HMMs. 138

8.13 Confusion table of audio/video action classification with fixed-weight coupled HMM model. Columns are original actions, rows are classified actions. 139
8.14 Confusion table of noisy audio/video action classification using coupled HMM with variable time-invariant weights. 139
8.15 Confusion table of noisy audio/video action classification using coupled HMM with variable time-varying weights. 139

LIST OF FIGURES

Figure **Page**

1.1 Human-to-human interaction and human-to-computer interaction. Humans perceive their environment through five basic senses. HCI, on the other hand, need not be bound to typical human senses. 2

1.2 Modalities for human sensing and action. Human beings sense the environment they live in through their senses. They act on the environment using numerous actuators. 2

2.1 Bayesian network—graphical depiction of dependencies in $Pr(z_0, z_1, z_2, z_3, z_4, z_5, z_6)$. Each node in the network represents one random variable. Set $a(z_5)$ denotes ancestor variables of z_5. Pdf $Pr(z_1, z_2, z_3, z_4, z_5, z_6)$ can then be decomposed as $Pr(z_6|z_4)$ $Pr(z_5|z_3, z_4)$ $Pr(z_3|z_2, z_0)$ $Pr(z_2|z_1)$ $Pr(z_4)$ $Pr(z_1)$ $Pr(z_0)$. 11

2.2 Exact probability propagation in singly connected Bayesian networks. Any variable (node) in the network separates the rest of the variables into two disjoint sets. Calculation of conditional probabilities then reduces to message passing between a node and its parents and children. 14

3.1 General notion of dynamic Bayesian networks. DBNs describe evolution of the states of some system in time in terms of their joint probability distribution. At each instance in time t the states of the system depend only on the states at the previous $(t-1)$ and possibly current time instance. 20

3.2 Dependency graph of a dynamic Bayesian network (DBN) with one hidden and one observable state at each time instance. 21

3.3 Inference in DBNs. Given a sequence of observations one needs to estimate the distribution of hidden variables (shaded circles). 22

3.4 Dependency graph of a hidden Markov model with a mixture of observation pdfs. Variable c_t specifies which observation distribution mixture component is active at time t. The observation y_t depends both on the hidden state x_t as well as the mixture variable c_t. The model is often used to define an HMM with a mixture of Gaussian observation pdfs. 33

3.5 State diagram representation of a discrete observation hidden Markov model in text. The model has six hidden states and three observation states with transition and observation probabilities denoted on the diagram. Equivalent DBN representation of the model is still the same as that depicted in Figure 3.2, but must also include the probability tables specified in text. 37

3.6 Dependency graph for an HMM with an observation sequence of length 5. 38

3.7 Complex dynamic Bayesian networks. The terms s and x denote discrete and continuous space variables, respectively. 39

3.8 State diagram of a Markov chain defined over the space of four models. Markov chain allows transitions from model m_0 to model m_1 or model m_3, model m_1 to model m_2, and model m_2 to model m_3. Given this model-to-model transition topology, only $P_{1,0}$, $P_{3,0}$, $P_{2,1}$, and $P_{3,2}$ are nonzero. Each model is assumed to be a specific DBN, as indicated in the lower figure. 40

4.1 A mixture of $N = 2$ dynamic Bayesian networks with observations in the joint observation set of cardinality $M = 5$. At each time instance t an observation $y_t^{(m)}, m = 0, \ldots, 4$ is generated by one of N states $x_t^{(n)}, n = 0, \ldots, 1$. Switching variable $s_t^{(m)}$ determines which of the two dynamic systems generates the observation $y_t^{(m)}$. For convenience, only the observations at $t = 2$ are shown. 43

4.2 Factorization of the mixture of dynamic Bayesian networks from Figure 4.1. For clarity, only the factorization of the mixture subnet $n = 0$ with observations at time $t = 2$ is depicted. Complete network factorization contains N such subnets. 45

5.1 Block diagram of a physical system (LDS) driven by an input generated from a hidden Markov model (HMM). The system has an equivalent representation as a mixed-state dynamic Bayesian network. 50

5.2 Bayesian network representation (dependency graph) of the mixed-state DBN. Term s denotes instances of the discrete valued concept states driving the continuous valued physical system states x and observations y. 52

5.3 Inference in decoupled mixed-state DBN can be easily accomplished once the coupling u. is known. 56

5.4 Factorization of the original mixed-state DBN. Factorization reduces the coupled network into a decoupled pair of an HMM (Q_s) and an LDS (Q_x). 57

5.5 Symbolical depiction of a variational inference loop in a mixed-state DBN. Assume that one is given an estimate of s. Estimate of LDS input u is obtained from s, and estimate of LDS state x is found from u and y. Finally, an updated estimate of s can be found from x. 59

5.6 Variational parameter q_τ as a function of input level d. Shown is a set of six different input levels d_0 through d_5. The function attains maximum for input level d_4, which is closest to the global maximum at $d^* = \langle x_\tau \rangle - A \langle x_{\tau-1} \rangle$. 60

6.1 Block diagram of two coupled systems with different internal dynamics. In this example, each system is modeled as a hidden Markov model. 64

6.2 Dependency graph of a two-subsystem (bimodal) coupled hidden Markov model. 65

6.3 Naive approach to inference in coupled HMMs. The M-modal coupled HMM is reduced to an ordinary HMM whose hidden states σ_t live in the Cartesian product of M state spaces of the original modal HMMs. The dimension of this new state space is $N = N(0) \cdot \ldots \cdot N(M-1)$. 67

6.4 Dependency graph of a coupled HMM with fixed intermodal weights. 69

6.5 Factorization of fixed-weight coupled HMM. 70

6.6 Result of variational inference on one sub-HMM of the approximating distribution Q. Inference in the submodel is performed by assuming that all other submodels have fixed and known hidden states. 71

6.7 Figurative representation of the variational inference algorithm for coupled HMMs. At each iteration, state variables of all submodels except one are assumed constant (subscript "old"). The remaining submodels' states are then re-estimated (s_{new}) as functions of the observations y and the fixed states of other models s_{old}. 72

6.8 Dependency graph of a coupled HMM with adaptive intermodal weights. 74

6.9 Factorized joint pdf for variational inference of a coupled HMM with adaptive intermodal weights. Factorization yields four independent networks that isolate the four sets of hidden variables of the original model. 75
6.10 Dependency graph of a coupled HMM with time-varying adaptive weights. Evolution of intermodal factors is modeled as another dynamic Bayesian network. 77
6.11 Factorized approximation of a coupled HMM with time-varying adaptive intermodal weights. Network consists of four independent subnets related to the four sets of hidden variables of the original model. 78

7.1 Vision-based gesture interpretation system. Visual images of gestures are acquired by one or more video cameras. They are processed in the analysis stage where the gesture model parameters are estimated. Using the estimated parameters and some higher-level knowledge, the observed gestures are inferred in the recognition stage. 83
7.2 Production and perception of gestures. Hand gestures originate as a mental gestural concept G, are expressed (T_{hg}) through arm and hand motion H, and are perceived (T_{vh}) as visual images V. 84
7.3 A taxonomy of hand gestures for HCI. Meaningful gestures are differentiated from unintentional movements. Gestures used for manipulation (examination) of objects are separated from the gestures which possess inherent communicational character. 85
7.4 Analysis of hand gestures. In the analysis stage, features F are extracted from visual images V. Model states $\hat{P}$ are estimated and possibly predicted. 87
7.5 Recognition of hand gestures. States of the physical model H are used to infer the underlying gestural concept G. 89
7.6 Block diagram of a gesture production system. Gestural concept c drives a physical system through an input u. System states x evolve according to some dynamics and are partially observed through intermediate features f as observations y. 90
7.7 Dynamic Bayesian network representation of a gestural action. 91
7.8 Dynamic Bayesian network representation of a directly observable gestural action, such as the ones measured by the computer mouse. 92
7.9 Hand position is measured relative to some stationary origin. In this case hand position (y_x, y_y) is defined with respect to head. 93
7.10 Examples of four symbols produced by computer mouse motions (left-to-right, top-to-bottom): "arrow," "erase," "circle," and "wiggle." 97
7.11 Three ways of modeling mouse-acquired symbols. From top to bottom: completely coupled LDS and HMM (mixed-state HMM), decoupled adapted LDS and HMM, and decoupled fixed LDS and HMM. 98
7.12 Estimates of concept states for "arrow" symbol. Top graph depicts the symbol and the estimated driving force. Bottom graph shows estimates of concept states obtained using variational inference (solid line) and Viterbi decoding (dashed line). 99
7.13 Classification error estimates of four mouse-acquired symbols. Shown are 95 percent confidence intervals for error counts. All three models (mixed, LDS/HMM, and grad/HMM) performed equally well in all cases but one. ("circle"). ($\bigtriangledown$ - mixed-state HMM, □ - fixed LDS/HMM, $\diamond$ - adapted LDS/HMM.) 100
7.14 Samples of four mouse-acquired symbols corrupted by additive zero-mean white noise with standard deviation of 0.01. 101
7.15 Classification error estimates of four noisy mouse symbols. Shown are 95 percent confidence intervals for error counts. Unlike the noise-free case, the coupled mixed-state HMM performed significantly better than the decoupled adapted and fixed LDS/HMM models. ($\bigtriangledown$ - mixed-state HMM, □ - fixed LDS/HMM, $\diamond$ - adapted LDS/HMM.) 101
7.16 Bayesian network model for visual tracking of the human hand in scenes with stationary backgrounds. For convenience, only one time slice of observations and switching states is shown. 103

7.17 Factorized Bayesian network model for visual tracking of the human hand in scenes with stationary backgrounds. For convenience, only one time slice of observations and switching states is shown. . . . 106
7.18 Experimental testbed for integrated multimodal interaction. (a) User interaction with BattleView simulation environment. (b) SmallWall frontal projection setup. . . . 111
7.19 Examples of free hand gestural actions. Shown are selected frames of "move up" (a) and "move left" (b) gestural actions. . . . 113
7.20 Hand and head tracking using DBN. Figure (a) depicts frame 300 of one complete gestural video set. Shown are also hand trajectory estimates and hand ROI with segmented hand region. Elliptical lines are isolevels of estimated spatial distribution of the hand. Figures (b) and (c) are examples of hand and head segmentations, respectively. Left to right, top to bottom: ROIs, foreground spatial pdfs $Pr(\chi_{\cdot}^{(\cdot)} \mid x_{\cdot}^{(f)})$, foreground assignment estimates $\left\langle s_{\cdot}^{(\cdot)}(f) \right\rangle$, and foreground ROI pixel sets. . . . 115
7.21 Grammar of gestural command language. . . . 117
7.22 Model state diagram of "move up" gestural action. . . . 117
7.23 Block diagrams of three gestural action classifiers. Top to bottom: coupled mixture of LDS/HMM classifier uses approximate probabilistic inference to decode the states of gestural concepts $\hat{c}$ from images of moving hand y; decoupled LDS/HMM and its nonlinear counterpart estimate the hand motion driving force and hand shape without the knowledge of underlying gestural actions. . . . 119

8.1 Generative multimodal model. Gestures and speech originate as a communication concept C. It is arguable whether and to what extent there is interaction between the two modal concepts. Speech and gestures are separated in the processing phase and expressed using different modalities H_V and H_A. Finally, they are perceived using hearing A and vision V.125
8.2 Competing inference models for bimodal gesture/speech recognition. Top to bottom: decision-level inference, highly coupled feature-level inference, naively coupled inference, coupled inference with adaptive time-invariant weight HMM, and coupled inference with adaptive time-varying weight HMM. . . . 130
8.3 State transition probability matrices for three coupled HMM models of speech/gesture action "up." Superscripts (v, v), (v, a), (a, v), and (a, a) denote visual intramodal, audio-to-visual intermodal, video-to-audio intermodal, and visual intramodal transition pdfs. . 132
8.4 Segmentation of visual (top) and verbal (bottom) parts of action "up." Shown are gestime-level segmentations of visual unit force in x direction and word-level segmentations of the first MFCC coefficient of the verbal action. Also depicted (small graphs) are the concept states of the respective actions. . . . 134

CHAPTER 1

INTRODUCTION

1.1 Background

With the ever-increasing role of computers in society, *human–computer interaction* or HCI has become an increasingly important part of our daily lives. As computing, communication, and display technologies progress even further, the existing HCI techniques may become a bottleneck in the effective utilization of the available information flow. For example, the most popular mode of HCI still relies on keyboards and mice. These devices have become familiar but tend to restrict the information and command flow between the user and the computer system. This limitation has become even more apparent with the emergence of novel display technology such as virtual reality [1, 2, 3] and wearable computers [4]. Thus, in recent years there has been a tremendous interest in introducing new modalities into HCI that will potentially resolve this interaction bottleneck.

One long-term goal in HCI has been to migrate the "natural" means that humans employ to communicate with each other into HCI (Figure 1.1). With this motivation, automatic speech recognition has been a topic of research for decades [5]. Some other techniques like automatic gesture recognition, analysis of facial expressions, eye tracking, force sensing, or electroencephalograph (EEG) have only recently gained more interest as potential modalities for HCI. Though studies have been conducted to establish the feasibility of these novel modalities using appropriate sensing and interpretation techniques, their role in HCI is still being explored.

Humans perceive the environment they live in through their senses—vision, hearing, touch, smell, and taste. They act on and in the environment using their actuators such as whole body, hands, face, and voice. Human-to-human interaction is based on sensory perception of actuator actions of one human by another, often in the context of an environment (Figure 1.2). In the case of HCI, computers perceive actions of humans. To have the human–computer interaction be as natural as possible, it is desirable that computers be able to interpret all natural human actions. Hence, computers should interpret human hand, body, and facial gestures, human speech, eye movements, etc. Some computer sensory

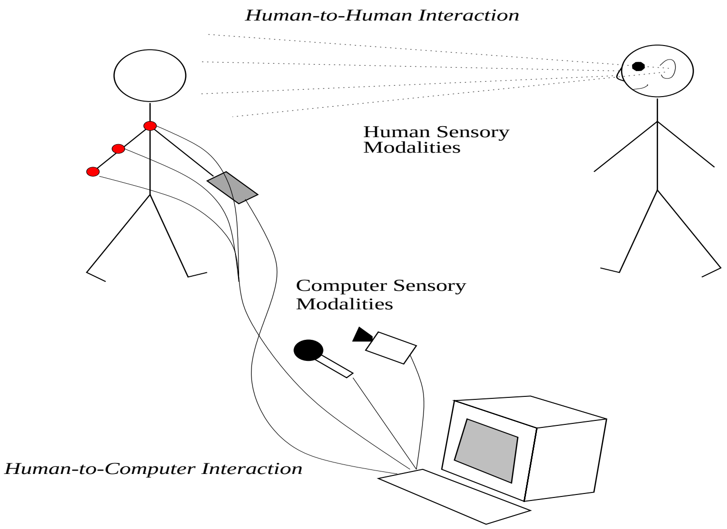

Human-to-Human Interaction
Human Sensory Modalities
Computer Sensory Modalities
Human-to-Computer Interaction

Figure 1.1 Human-to-human interaction and human-to-computer interaction. Humans perceive their environment through five basic senses. HCI, on the other hand, need not be bound to typical human senses.

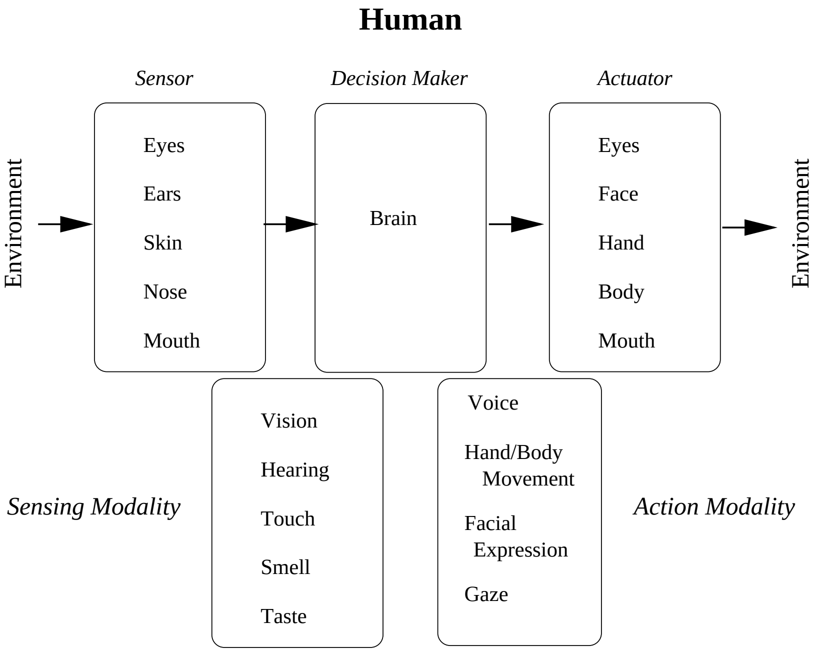


Figure 1.2 Modalities for human sensing and action. Human beings sense the environment they live in through their senses. They act on the environment using numerous actuators.

modalities are analogous to the human ones. Computer vision and automatic speech recognition mimic the equivalent human sensing modalities. However, computers also possess sensory modalities that humans lack. They can accurately estimate the position of the human hand through magnetic sensors and measure subtle changes of the electric activity in the human brain, for instance. Thus, there is a vast repertoire of human action modalities that can potentially be perceived by a computer.

1.1.1 Human action modalities for HCI

A large repertoire of human actions could possibly be incorporated into HCI by designing suitable sensing mechanisms. Historically, the action modalities most exploited for HCI are based on hand movements. This is largely due to the dexterity of the human hand, which allows accurate selection and positioning of mechanical devices with the help of visual feedback. Appropriate force and acceleration can also be applied easily using the human hand. The hand movement is exploited in the design of numerous interface devices—keyboard, mouse, stylus, pen, wand, joystick, trackball, etc. The keyboard provides a direct way of providing text input to the computer, but the speed is obviously limited and can be improved only at a limited rate. Similarly, hand movements cause a cursor to move on the computer screen (or a 3D display). The next level of action modalities involves the use of *hand gestures*, ranging from simple pointing, through manipulative gestures, to more complex symbolic gestures such as those based on the American Sign Language. With a glove-based device the ease of hand gestures may be limited, but with non-contact video cameras, free hand gestures would be easier to use for HCI. The role of free hand gestures in HCI could be further improved, to require less training for instance, by studying the role of gestures in human communication. A multimodal framework is particularly well suited for embodiment of hand gestures into HCI.

In addition to hand movements, a dominant action modality in human communication is the production of sound, particularly spoken words. The production of speech is usually accompanied by other visible actions, such as lip and eye movement, which can be exploited in HCI as well. The facial expression and body motion, if interpreted appropriately, can help in HCI. Even a subtle “action” such as a controlled thought has been investigated as a potential candidate for HCI.

1.1.2 Computer sensing modalities for HCI

What action modality to use for HCI is largely dependent on the available computer sensing technology. A broad number of categories of the computer sensing modalities has been previously considered for HCI. We next examine how the above human action modalities might be measured and interpreted by a computer.

1.1.2.1 Position and motion sensing

Many interface devices have been built to sense the position and motion of the human hand and other body parts for use in HCI. The keyboard is the simplest such interface, where the touch of a particular key indicates that one of a set of possible inputs was selected. More accurate position and motion sensing in a 2–D plane is used in interface devices such as a mouse, light pen, stylus, and tablet [6, 7]. Three-dimensional position/motion sensing is commonly done through a joystick or a trackball. For a brief history of HCI technology covering these familiar computer sensing modalities, see [8]. For tracking the head (to display the graphics with the correct perspective) various forms of sensors have been employed. Electromagnetic fields [9] are the most popular method, but are expensive and restricted to a small radius. Ultrasonic tracking requires line of sight and is inaccurate, especially at long ranges [10]. Other methods might include tracking of infrared LEDs, or inertial trackers, using accelerometers. Attempts to solve the hand tracking resulted in glove-based mechanical devices that directly measure hand and/or arm joint angles and spatial position [11, 12, 13, 14, 15]. Interpretation of motion perceived in this fashion has gone beyond simple position measurements. Numerous techniques based on statistical interpretation of dynamic patterns have been explored for classification of common types of stylus motion [6, 7] or finger movements [11].

1.1.2.2 Audio sensing

The direct motivation for sensing the sound waves using a (set of) microphone(s) and processing the information using techniques known as *automatic speech recognition* (ASR), is to be able to interpret speech, the most natural human action modality for HCI. Significant advances have been made toward the use of ASR for HCI [5]. However, the current ASR technology is still not robust, especially outside controlled environments, under noisy conditions and with multiple speakers [16]. Methods have been devised, for example, by using microphone arrays and noise cancelation techniques to improve the speech recognition. However, these tend to work only for the environments for which they are designed. An active research area is concerned with making the ASR sufficiently robust for use in HCI. For instance, it has been demonstrated conclusively that the recognition rate for speech can be improved by using visual sensing to simultaneously analyze the lip motion [17]. Other visual sensing modalities such as gesture analysis may also help in improving the speech interpretation [18].

1.1.2.3 Visual sensing

A video camera together with a set of techniques for processing and interpreting the image sequence can make it possible to incorporate a variety of human action modalities into HCI. These actions include hand gestures [19], lip movement [17], gaze [20, 21, 22], facial expressions [23], head and other body movements [24, 25]. Visually interpreted gestures, for instance, can allow a tetherless manipulation of virtual reality [26] or augmented reality displays [27]. However, the use of visual sensing for HCI suffers difficulties from both a theoretical and practical standpoint. The problem of visual interpretation of hand gestures is still not well understood, particularly when it is desirable not to put restrictions on the hand movements for more natural HCI [19]. From a practical standpoint, visual sensing involves the processing of huge amounts of information in real time, which could put undue demands on the processing power of the system being controlled. Furthermore, visual sensing requires an unoccluded view of the human, putting restrictions on the motion of the user and the physical setting for HCI. Nonetheless, the use of computer vision for improving HCI continues to be a topic of very active research [28]. Visual sensing can be especially useful in conjunction with other sensing modalities [29] such as lip reading with audio [17], lip reading with eye tracking [30], visual gesture recognition with speech [18], etc.

1.1.2.4 Tactile and force sensing

The dexterity of the human hand for accurately positioning a mechanical device can be coupled with application of “force” which can be sensed by using appropriate *haptic* devices. The computer sensing of touch and force is especially important for building a proper feel of “realism” in virtual reality. The key idea is that by exerting force or touch on virtual objects (with the corresponding haptic display for feedback), the user will be able to manipulate the virtual environment in a natural manner. Situations where such realism is especially important include, for example, simulation of surgery for training [31, 32]. Force sensing is a topic of very active research since is it difficult to design suitable devices with the desired accuracy without constraining the user [33, 34]. A better force sensing for HCI may also be obtained by also simultaneously considering the sensing of position and motion.

1.1.2.5 Neural sensing

One computer sensing modality that has been explored with increasing interest is based on the monitoring of brain electrical EEG activity [35, 36, 37, 38]. The brain activity can be monitored non-invasively from the surface of the scalp and used for directly controlling the computer display. This form of interaction is also termed *brain-actuated control*, or BAC. The “hands-free” nature of the resulting HCI makes it attractive for head-mounted displays and situations (such as aircraft pilot) where hands are being used in other tasks. Another very big impetus for pursuing this sensing modality is as a means of HCI for the physically disabled [39]. However, it requires training (using biofeedback and self-regulation) so that specific brain-responses may be modulated [40]. There are many theoretical and applied open problems that need to be addressed for BAC, for example, how user distractions and/or increased workload affect such an interface. An alternative approach includes sensing surface electromyographic

(EMG) signals [41]. Approaches have also been suggested for using multimodal sources that include eye tracking and monitoring of muscle tension in conjunction with EEG [42, 43].

1.2 Motivation

Recent advances in computing, communication, and display technologies have put significant stress on improving the information flow between the human user and a computer. A number of computer sensing modalities have become available to facilitate this task. However, existence of these devices alone is not sufficient for the improvement of HCI. Efficient interpretation of the sensed modalities is a major imperative towards more natural HCI.

1.2.1 Unimodal interpretation

The main task in interpreting a single sensed modality is to infer as much information content from the modality as possible. For instance, in tracking the mouse motion it may be beneficial to interpret it on a "scale" higher than just a sequence of unrelated screen coordinates. A sequence of points can describe a circle around a graphical user interface (GUI) window or a line across it to indicate the user's wish to move or delete the window. Similarly, a hand motion can simply indicate a deictic (pointing) action, or it may specify the size and shape of a complex virtual object. This level of interpretation has been long present in the machine interpretation of speech as the main human-to-human communication modality. However, it has only recently begun to find its way into interpretation of other modalities.

One common thread among all computer sensory interpretation tasks is the need to interpret *temporal* and sometimes *spatial* patterns of the data "transmitted" by the user and acquired by a computer. Thus, the framework of *pattern recognition* easily comes to mind. Common user's intentions and actions can be thought to produce "typical" patterns of sensory data that need to be recognized by the machine. However, such data patterns may vary more or less from one user to another and from time to time. It is then plausible to assume that even though some variability in data patterns exists it can be sufficiently well described in the *probabilistic* framework. Hence, *statistical pattern recognition* becomes an attractive direction to pursue. Different approaches to spatio-temporal series analysis and interpretation within the framework of statistical pattern recognition have been studied for decades. Hidden Markov models (HMMs) [5] have been particularly successful as tools for modeling and recognition of spoken language. On the other hand, the role of probabilistic or Bayesian networks (see Chapter 2) has been known in the field of artificial intelligence and exploited in different expert systems to model complex interaction among causes and consequences. Recently but not unexpectedly, Bayesian networks have found their way into time series modeling [44]. In fact, it was shown that the successful HMMs and grammars are nothing else but a special case of *dynamic Bayesian networks* (DBNs) [44]. With that in mind, it is natural and tempting to fully explore the role and benefits of Bayesian networks in interpreting different HCI modalities.

1.2.2 Multimodal interpretation

Until now we have focused my attention on solving (or posing) the problem of interpretation of single, independent interaction modalities. However, the interaction of humans with their environment (including other humans) involves *multiple, concurrent modes of communication.* We speak about, point at, and look at objects all at the same time. We also listen to the tone of a person's voice and look at a person's face and arm movements to find clues about his or her feelings. To get a better idea about what is going on around them people look, listen, touch, and smell. When it comes to HCI, on the other hand, people usually use only one interface device at a time—typing, clicking the mouse button, speaking, or pointing with a magnetic wand. The "ease" with which this unimodal interaction allows one to convey her intent to the computer is far from satisfactory.

Several studies have confirmed that people prefer to use multiple action modalities for virtual object manipulation tasks [2, 6]. In [2] Hauptmann and McAvinney concluded that 71 percent of their subjects preferred to use both speech and hands to manipulate virtual objects rather than just one of the modalities alone. Oviatt has shown in [6] that 95 percent of the subjects in a map manipulation task tend to use gestures together with speech. Multiple modalities also *complement* each other. Cohen has shown [45], for example, that gestures are ideal for direct object manipulation while natural language is more suited for descriptive tasks. Another drawback of current advanced single modality HCI is that it lacks robustness and accuracy. For example, modern automatic speech recognition systems are still error-prone in the presence of noise and require directed microphones or microphone arrays. Automatic gesture recognition systems are still constrained to the recognition of few predefined hand movements and are burdened by cables or strict requirements on background and camera placement [19]. However, concurrent use of two or more interaction modalities may loosen the strict restrictions needed for accurate and robust interaction with the individual modes. For instance, spoken words can *affirm* gestural commands, and gestures can *disambiguate* noisy speech. Redundant multimodal inputs can also enable physically or cognitively handicapped people access to computers (or computer-controlled devices).

A rationale for integration of multiple sensory modalities can be found in nature. Human beings as well as other animals integrate multiple senses. Studies of the superior colliculus have shown that different senses are initially segregated at the neural level. When they reach the brain, sensory signals converge to the same target area in the superior colliculus, which also receives signals from the cerebral cortex and which, in turn, modulates resultant behavior. A majority (about 75 percent) of neurons leaving superior colliculus are *multisensory.* This strongly suggests that the use of multimodality in HCI would be desirable, especially if the goal is to incorporate the naturalness of human communication into HCI.

Lastly, the rationale for combining different sensory data may come from statistical data analysis. The disadvantage of using a single sensor system is that it may not be able to adequately reduce the uncertainty for decision making. Uncertainty arises when features are missing, when the sensor cannot measure all relevant attributes, or when observations are ambiguous [46]. On the other hand, it is well known that it is statistically advantageous to combine multiple observations from the same source because improved estimates are obtained using redundant observations [47]. It is also known that multiple types of sensors may increase the accuracy with which a quantity can be observed. For example, if x_i and x_j are two (statistically independent) estimates of one quantity corrupted by Gaussian noise, the minimum mean square error combination of the two estimates results in

$$x_{ij} \;=\; \left(\Sigma_i^{-1} + \Sigma_j^{-1}\right)^{-1} \Sigma_i^{-1} x_i + \left(\Sigma_i^{-1} + \Sigma_j^{-1}\right)^{-1} \Sigma_j^{-1} x_j,$$

where Σ_i and Σ_j are the variances of x_i and x_j, respectively. Moreover, the variance of the fused estimate Σ_{ij} is given by

$$\Sigma_{ij}^{-1} \;=\; \Sigma_i^{-1} + \Sigma_j^{-1}.$$

Thus, the variance of the fused estimate Σ_{ij} is “smaller” than the variances of either of the two original estimates. This can be easily generalized to more than two redundant observations.

Bayesian networks again emerge as a powerful framework for the multimodal interpretation task. Modeling of different observations (modality sensory data) driven by more or less correlated multiple causes can be easily achieved in this network model domain, as will be shown in Chapters 2 and 8.

1.3 Organization

The dissertation is organized as follows. Chapter 2 introduces the theoretical concepts of probabilistic (Bayesian) networks. It defines the problems of inference and learning and addresses issues of approximate inference and general learning techniques.

Chapter 3 focuses on DBNs. This class of networks is particularly suitable for the modeling of time series. Within the domain of dynamic models, the problems of forward and backward propagation, smoothing, prediction, decoding, and learning are defined. Two essential classes of DBNs are studied: *discrete-state* HMMs and *continuous-state linear dynamic systems* (LDSs).

Chapters 4, 5, and 6 introduce three novel models of dynamic Bayesian networks. Each of the models fuses several basic DBN types to achieve modeling of complex processes. A *mixture of DBNs* in Chapter 4 considers the case of how one can associate multiple observations with a number of underlying dynamic processes. In Chapter 5 a model is formulated that describes the evolution of an LDS driven by a concept-generated input. Finally, Chapter 6 explores ways of modeling interactions between two or more HMM-based concept models that draw their measurements from different observation spaces. In each of the three chapters, the issues of efficient inference and model parameter learning are addressed.

Chapter 7 deals with modeling, analysis, and recognition of free hand gestures within the scope of HCI. A particular temporal and spatial model of hand gestures is first suggested. Analysis of the model's parameters is then tackled in this framework. Then, a DBN-based architecture for gesture recognition is proposed founded on the complex DBN models of Chapters 4 and 5. In particular, focus is on two examples of gestural interactions: mouse-acquired hand motion, and visually perceived gestural actions.

Chapter 8 further expands the notion of a single modality recognition to the realm of multiple modalities. Multiple modalities are an effective domain for efficient, natural HCI. To model the correlation of multiple modalities we propose a multimodal Bayesian network framework based on coupled HMMs introduced in Chapter 6. The framework is then applied to modeling of gestural and verbal actions for virtual display control.

The dissertation is concluded with the discussion of DBN models and results and the proposal for future work in Chapter 9.

1.4 Contribution

Original contributions of this work span the areas of DBNs and interpretation of unimodal and multimodal computer sensory inputs such as gestures and speech for advanced HCI. In particular, we have addressed the following important issues:

- formulation of *mixtures of dynamic Bayesian networks* model,
- formulation of *mixed-state dynamic Bayesian network* model,
- formulation of *coupled hidden Markov model*,
- introduction of dynamic Bayesian network framework for modeling and interpretation of hand gesture, and
- introduction of dynamic Bayesian networks for modeling and interpretation of multimodal gesture and speech interaction.

The issues addressed are not by any means confined to the area of human–computer interaction. The strong theoretical foundation of the proposed techniques allows for their applicability to general problems of time series modeling and classification under multiple observation data sets.

CHAPTER 2

BAYESIAN NETWORKS

2.1 Introduction

A Bayesian network is a graphical model used to describe dependencies in a probability distribution function (pdf) defined over a set of variables. Namely, dependencies among variables are represented in a graphical fashion and used to decompose (factor) the distribution in terms of conditional independence relations defined over subsets of variables.

Let $\mathcal{Z}_L = \{z_0, \cdots, z_{L-1}\}$ be a set of L random variables (for example, see Figure 2.1). Let $Pr(z_0, \cdots, z_{L-1})$ be a probability density function defined over $\mathcal{Z}_L$. Without knowing what the dependencies among variables $z_\cdot$ are, one can apply the chain rule of basic probability theory and decompose the pdf $Pr(\cdot)$ as, for example,

$$\begin{aligned} &Pr(z_0, \ldots, z_{L-1}) = \\ &\qquad Pr(z_0)Pr(z_1|z_0)Pr(z_2|z_1, z_0) \cdot \ldots \cdot Pr(z_{L-1}|z_{L-2}, \ldots, z_0). \end{aligned}$$

Again, this decomposition holds in general and does not presume any knowledge of particular random variable dependencies. However, it is often the case that a certain structure exists in those dependencies:

> A Bayesian network is a graphical way of representing a particular joint distribution factorization.

Each random variable of a pdf associated with the Bayesian network is represented as one *node* in such a network. For instance, in Figure 2.1 there are seven nodes, thus the graph defines a pdf of seven random variables. Directed arcs in the graph represent dependencies among variables. Thus, variable z_5 in Figure 2.1 is only influenced by z_3 and z_4. In other words, given the values of z_3 and z_4, z_5 is *conditionally independent* from the rest of the random variables. This observation can be generalized in the following fashion:

$$Pr(z_0, \cdots, z_{L-1}) = \prod_{i=0}^{L-1} Pr(z_i|a(z_i)), \tag{2.1}$$

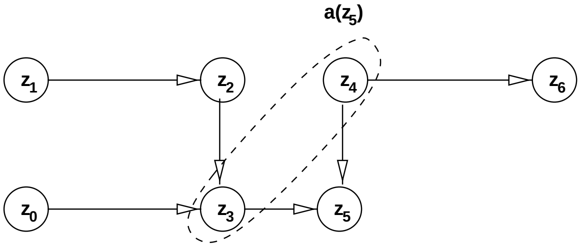


Figure 2.1 Bayesian network—graphical depiction of dependencies in $Pr(z_0, z_1, z_2, z_3, z_4, z_5, z_6)$. Each node in the network represents one random variable. Set $a(z_5)$ denotes ancestor variables of z_5. Pdf $Pr(z_1, z_2, z_3, z_4, z_5, z_6)$ can then be decomposed as $Pr(z_6|z_4)$ $Pr(z_5|z_3, z_4)$ $Pr(z_3|z_2, z_0)$ $Pr(z_2|z_1)$ $Pr(z_4)$ $Pr(z_1)$ $Pr(z_0)$.

where $a(z_i)$ denotes a subset of $\mathcal{Z}_L$ whose elements are directly influencing z_i. Subset $a(z_i)$ is usually denoted as *ancestors* or *parents* of z_i. For instance, in Figure 2.1, ancestors of z_5 are z_3 and z_4, denoted as the set $a(z_5) = \{z_3, z_4\}$. Hence, the joint pdf can be decomposed as

$$\begin{aligned} &Pr(z_0, z_1, z_2, z_3, z_4, z_5, z_6) = \\ &\quad Pr(z_6|z_4)\ Pr(z_5|z_3, z_4)\ Pr(z_3|z_2, z_0)\ Pr(z_2|z_1)\ Pr(z_4)\ Pr(z_1)\ Pr(z_0). \end{aligned}$$

At this point it is useful to define some basic notation often encountered in Bayesian network theory. The notation actually originates in graph theory.

- A *parent* of node z_i is every node z_j such that there is a directed arc from z_j to z_i.
- A directed arc from node z_j to node z_i implies that z_i is z_j's *child.*
- *Descendants* of node z_j are all its children, its children's children, etc.
- An *undirected path* from node z_j to node z_k is a sequence of nodes starting in z_j and ending in z_k such that each node in the sequence is either a parent or a child of its successor.
- A *directed path* from node z_j to node z_k is an undirected path from z_j to z_k, where each node in the path is strictly a parent of its successor.
- Node z_k in an undirected path has *converging arrows* if it is a child of both the previous and the following nodes in the path.

For example, in Figure 2.1 $\{z_0, z_3, z_5, z_4\}$ is an undirected path, while z_1, z_2, z_3 is a directed path.

I now outline some important rules of Bayesian networks [48]:

- Each node in the network is conditionally independent from its nondescendant given its parents.
- A set of nodes $\mathcal{Z}_i$ is conditionally independent of another (disjoint) set $\mathcal{Z}_j$ given set $\mathcal{Z}_s$ if $\mathcal{Z}_s$ *d-separates* $\mathcal{Z}_i$ and $\mathcal{Z}_j$. Namely, in every undirected path between a node in $\mathcal{Z}_i$ and a node in $\mathcal{Z}_j$ there is a node z_k such that:

- z_k has converging arrows and neither z_k nor its descendants are in $\mathcal{Z}_s$, or
- z_k does not have converging arrows and z_k is in $\mathcal{Z}_s$.

A power of Bayesian networks is that one can infer conditional variable dependencies by *visually* inspecting the network's graph. Returning to the network in Figure 2.1, one can quickly conclude that z_0 and z_1 are conditionally independent given z_2 and z_3 since they d-separate z_0 and z_1. Hence, it follows that $Pr(z_0, z_1|z_2, z_3) = Pr(z_0|z_2, z_3)Pr(z_1|z_2, z_3)$. However, z_1 and z_0 are not conditionally independent given only z_3 since z_3 has converging arrows.

Why is such decomposition important? The answer lies in *probability inference*. Inference is the task of efficiently deducing what a distribution over a particular subset of random variables is given that one knows the states of some other variables in the network. More precisely, one needs to efficiently calculate a particular conditional or marginal pdf from the one defined by the net. The following section is devoted to discussion of inference in Bayesian networks.

2.2 Inference

2.2.1 Bayesian rule inference

Consider the partition $\mathcal{Z}_L = \mathcal{X} \cup \mathcal{Y}$. Let $\mathcal{X} = \{x_0, \dots, x_{N-1}\}$ and $\mathcal{Y} = \{y_0, \dots, y_{M-1}\}$, $L = N + M$, and denote the two subsets as the sets of *hidden* and *visible* variables, respectively. Furthermore, let U_K be an arbitrary subset of $\mathcal{Z}_L$. The goal of inference is to find the conditional pdf over $\mathcal{U}_K$ given the observed variables $\mathcal{Y}$, namely $Pr(\mathcal{U}_K|\mathcal{Y})$.

If $\mathcal{U}_K \subseteq \mathcal{Y}$, the desired pdf is trivially equal to

$$Pr(\mathcal{U}_K|\mathcal{Y}) = \prod_{k=1}^{K} \delta(u_k - y_k),$$

where $\delta(x) = 1$ for $x = 0$ and $\delta(x) = 0$ otherwise.

A nontrivial case arises when $\mathcal{U}_K \subseteq X$. The desired pdf can now be obtained using the *Bayes rule*:

$$\begin{aligned} Pr(\mathcal{U}_K|\mathcal{Y}) &= \frac{Pr(\mathcal{U}_K, \mathcal{Y})}{Pr(\mathcal{Y})} &\quad (2.2) \\ &= \frac{Pr(\mathcal{U}_K, \mathcal{Y})}{\sum_{\mathcal{U}_K} Pr(\mathcal{U}_K, \mathcal{Y})}. &\quad (2.3) \end{aligned}$$

Clearly, it is sufficient to find the joint pdf $Pr(\mathcal{U}_K, \mathcal{Y})$ and then marginalize over $\mathcal{U}_K$. Moreover, the joint pdf over $\mathcal{U}_K$ and $\mathcal{Y}$ is obtained by marginalizing $Pr(\mathcal{Z}_L)$ over the set of hidden variables $\mathcal{X} - \mathcal{U}_K$:

$$Pr(\mathcal{U}_K, \mathcal{Y}) = \sum_{x \in \mathcal{X} - \mathcal{U}_K} Pr(x, \mathcal{U}_K, \mathcal{Y}) = \sum_{x \in \mathcal{X} - \mathcal{U}_K} Pr(\mathcal{Z}_L). \quad (2.4)$$

For instance, in Figure 2.1 one may want to know $Pr(z_3|z_1, z_5, z_6)$. One could directly apply the Bayesian rule, write

$$\begin{aligned} Pr(z_3|z_1, z_5, z_6) &= \frac{Pr(z_3, z_1, z_5, z_6)}{Pr(z_1, z_5, z_6)} \\ &= \frac{\sum_{z_0,z_2,z_4} Pr(z_0, \ldots, z_6)}{\sum_{z_0,z_2,z_3,z_4} Pr(z_0, \ldots, z_6)}, \end{aligned}$$

and then perform the marginalization given the decomposition defined by the inference graph. In small networks (or pdfs with few variables) like the one of Figure 2.1, that may not be a difficult problem. However, direct application of the above rule becomes less feasible as the number of variables increases. In fact, inference in arbitrary Bayesian networks is, in general, NP-hard [49]. Nevertheless, there are several special cases of Bayesian network topologies that allow for more efficient inference algorithms. I next consider one such network class.

2.2.2 Exact probability propagation in a singly connected network

An often encountered class of Bayesian networks is the class of *singly connected* Bayesian networks. Singly connected Bayesian networks contain no undirected cycles. In other words, their equivalent undirected graphs are in fact trees. The tree structure leads to an efficient inference algorithm for the case of a single desired variable ($K = 1$) known as the *sum-product* algorithm [50, 48, 51].

The idea behind the sum-product algorithm is to distribute the global sum over all hidden variables $\mathcal{U}_K$ of Equation 2.4 into a product of local sums over local subsets of hidden variables. Local subsets are determined by the topology of the net. Namely, consider the network in Figure 2.1. The network is singly connected. Its nodes can be rearranged as shown in Figure 2.2. Suppose one is interested in $Pr(z_3|z_1, z_5, z_6)$, as before. The basis of the sum-product algorithm is the fact that z_3 separates all the nodes in the network into two disjoint sets. These sets are labeled as $E^+(z_3)$ and $E^-(z_3)$. Set E^+ consists of parents of z_3 and all the nodes in undirected paths to z_3 that pass through one of its parents. Set E^- contains the children nodes and the nodes connected to z_3 through its children. To calculate the conditional pdf of interest, node z_3 collects "messages" from its parents and its children. Each parent of z_3 sends it a message containing the probability of every value of that parent given the observations in set E^+. Each child of z_3 sends back a message containing the probability of observations in E^- given every setting of z_3. The conditional pdf of z_3 is then the product of two terms. The first term is the sum of the messages to z_3 from its parents weighted by the conditional probability of z_3 given its parents. The second term is the product of the messages of the children of z_3. Hence,

$$Pr(z_3|z_1, z_4, z_5) \propto \left[\sum_{z_0,z_2} Pr(z_3|z_0, z_2) Pr(z_2|z_1) Pr(z_0|z_1) \right] Pr(z_5|z_3).$$

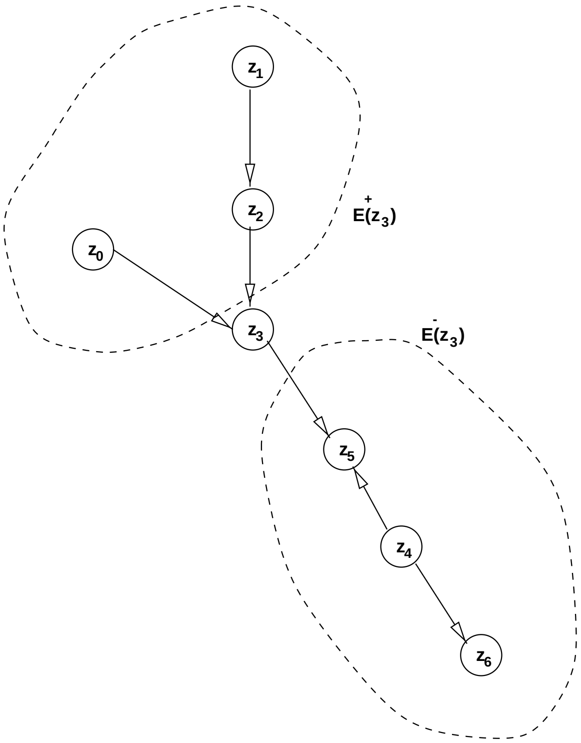


Figure 2.2 Exact probability propagation in singly connected Bayesian networks. Any variable (node) in the network separates the rest of the variables into two disjoint sets. Calculation of conditional probabilities then reduces to message passing between a node and its parents and children.

This procedure is repeated for every hidden variable (node) in the network. Hence, one can write similar expressions for $Pr(z_2|z_1)$ and $Pr(z_0|z_1)$. The well-known *forward-backward* algorithm often mentioned in the hidden Markov model literature [5] is clearly one particular view of the probability propagation algorithm in singly connected networks.

2.2.3 Approximate inference

In the case of arbitrary Bayesian networks, exact inference may not be feasible. Therefore, it is more plausible to look for an approximate, yet tractable, solution to the inference problem. I briefly mention a number of different approximate techniques.

Monte Carlo inference techniques [52] approximate a desired conditional distribution with the relative frequencies of occurrence obtained by simulation. In some cases, *inference by ancestral simulation* yields satisfactory results. Another alternative is to use *Gibbs sampling*, a Markov chain Monte Carlo technique [53], where a Markov chain is designed over the space of hidden variables so that its stationary distribution approximates the desired conditional distribution in the network.

Helmholtz machine inference [54] attempts to circumvent the main drawback of Monte Carlo and variational techniques, namely, the need for repeated expensive computations for every different constellation of hidden variables $\mathcal{X}$. This is achieved by constructing a recognition Bayesian network corresponding to a specific configuration of visible and hidden variables in the original generative network. The recognition network topology is designed so that the inference problem in this network becomes computationally tractable. A commonly found recognition network topology is the one of *factorial networks*, where $Q(\mathcal{X}|\mathcal{Y}) = \prod_{h_i \in \mathcal{X}} Q_i(h_i|\mathcal{Y})$.

Variational inference techniques [55] rely on calculation of a parameterized distribution which is in some sense close to the desired conditional network distribution, yet easier to compute. Briefly, a distribution $Q(\mathcal{X}|\eta)$ with variational parameters η is defined such that a convenient distance measure between $Q(\mathcal{X}|\theta, \eta)$ and $Pr(\mathcal{X}|\mathcal{Y})$ is minimized with respect to η. The most common choice of the distance measure is the Kullback–Leibler divergence:

$$\eta^* \quad = \quad \arg\min_{\eta} \sum_{\mathcal{X}} Q(\mathcal{X}|\eta) \log \frac{P(\mathcal{X}|\mathcal{Y})}{Q(\mathcal{X}|\eta)}. \tag{2.5}$$

Topology of Q is chosen such that it closely resembles the topology of P. However, as mentioned before, the topology of Q *must* allow a computationally efficient inference. In Appendix A we present an important theorem due to Ghahramani [56] that provides simplified conditions for optimal variational parameters of a specific exponential family of distributions Q.

2.3 Learning

The role of learning is to adjust the parameters of a Bayesian network so that the pdf defined by the network sufficiently describes statistical behavior of the observed data.

Let $\mathcal{M}$ be a parametric Bayesian network model with parameters θ of the probability distribution defined by the model. Let $Pr(\mathcal{M})$ and $Pr(\theta|\mathcal{M})$ be the prior distributions over the set of models and the space of parameters, respectively. Given some observed data assumed to have been generated by the model, the goal of learning in Bayesian framework is to estimate the model parameters θ such that the posterior probability of the model-given data (instances of random variables) $\mathcal{Z}_L$,

$$Pr(\mathcal{M}|\mathcal{Z}_L) = \frac{Pr(\mathcal{M})}{Pr(\mathcal{Z}_L)} \int_{\theta} Pr(\mathcal{Z}_L|\theta, \mathcal{M}) Pr(\theta|\mathcal{M}) d\theta, \tag{2.6}$$

gets maximized. To make this task tractable, however, it is usually assumed that the pdf of the parameter of the model, $Pr(\theta|\mathcal{M})$, is *highly peaked* around the *maximum likelihood estimates* of those parameters. In other words,

$$Pr(\mathcal{M}|\mathcal{Z}_L) \approx \frac{Pr(\mathcal{M})}{Pr(\mathcal{Z}_L)} Pr(\mathcal{Z}_L|\theta_{ML}, \mathcal{M}) Pr(\theta_{ML}|\mathcal{M}),$$

where the maximum likelihood estimate $\theta_M L$ is obtained from

$$\theta_{ML} = \arg\max_{\theta} \log Pr(\mathcal{Z}_L|\theta) \tag{2.7}$$

for a given model $\mathcal{M}$. Hence, in the rest of this section we simply focus on maximum likelihood learning of model parameters without explicitly discussing its appropriateness.

2.3.1 Learning with hidden variables

Consider now the case where not all variables $\mathcal{Z}$ in Bayesian network $\mathcal{M}$ are observed. Using notation of Section 2.2, maximum likelihood learning of network parameters can now be stated as the following optimization problem:

$$\hat{\theta} = \arg\max_{\theta} \log \sum_{\mathcal{X}} P(\mathcal{Y}, \mathcal{X}|\theta), \tag{2.8}$$

where P denotes a specific joint pdf defined by the network. Alternatively, one can minimize the cost function defined as

$$J(\theta) = -\log \sum_{\mathcal{X}} P(\mathcal{Y}, \mathcal{X}|\theta). \tag{2.9}$$

2.3.1.1 Gradient-based learning

To minimize the above cost one could, at least in principle, employ a number of different optimization techniques. A number of such techniques rely on the gradient of the cost function $J(\theta)$. It can easily be shown that the following expression for the gradient holds:

$$\nabla J(\theta) = \frac{\partial J}{\partial \theta} = -\sum_{\mathcal{X}} P(\mathcal{X}|\mathcal{Y}, \theta) \log P(\mathcal{X}, \mathcal{Y}|\theta). \tag{2.10}$$

This expression can be simplified in the case of discrete network variables [57]. In that case Bayesian network parameters θ are the conditional probabilities of hidden variables given their parents $Pr(x_i|a(x_i))$. Assuming that x_i can take on one of $j = 0, \ldots, J-1$ possible values whereas $a(x_i)$ can take on one of $k = 0, \ldots, K-1$ possible values, one can show that the gradient in the direction of parameter $\theta_{ijk} = Pr(x_i = j|a(x_i) = k)$ yields

$$\frac{\partial J}{\partial \theta_{ijk}} = -\frac{Pr(x_i = j, a(x_i) = k|\mathcal{Y})}{\theta_{ijk}}. \tag{2.11}$$

This shows that the gradient in the direction of *local* parameter θ_{ijk} involves computation of local statistics $Pr(x_i = j, a(x_i) = k|\mathcal{Y})$. Such statistics occur as by-products of inference calculations in all standard Bayesian net inference algorithms and are therefore readily available.

However, when minimizing cost J using a gradient approach, one has to keep in mind that such minimization is indeed constrained: $\sum_j \theta_{ijk} = 1$. Hence, every parameter update obtained through the use of Equation 2.11 must be projected onto an appropriate constrained surface. Together with Equation 2.11 this yields the following gradient-based network parameter update algorithm [57].

```
Initialize $\theta$;
while ( $\Delta\theta > maxError$ ) {
        for each ( i, j, k ) {
                $\Delta\theta_{ijk} = -\frac{Pr(x_i=j,a(x_i)=k|\mathcal{Y})}{\theta_{ijk}}$;
        }
        Project $\Delta\theta$ onto constraint surface;
        $w \leftarrow w + \alpha\Delta\theta$;
}
```

In the above algorithm α is used to denote the *learning rate.* To ensure convergence of the gradient minimization approach, α has to satisfy certain conditions, some of which are outlined in [58].

2.3.1.2 Expectation-maximization learning

"Generic" gradient optimization of cost function J is not always necessary. An iterative procedure known as the *expectation-maximization* (EM) algorithm [59] is usually employed. Here, however, I present a generalization of the original EM approach, known as the *generalized EM* (GEM) due to Hathaway [60] and Neal and Hinton [61]. This generalization elegantly encompasses a step that validates the use of *variational inference*, a powerful approximate inference technique discussed in Section 2.2.3.

To derive the GEM algorithm, consider any positive function $Q(\mathcal{X})$ such that

$$\sum_{x \in \mathcal{X}} Q(x) = 1.$$

It is convenient to write this function in the following form:

$$Q(\mathcal{X}) = \frac{e^{-H_Q(\mathcal{X},\mathcal{Y})}}{Z_Q}, \tag{2.12}$$

where $H_Q(\mathcal{X},\mathcal{Y})$ is some (positive) function, usually referred to as the Hamiltonian of Q, and $Z_Q = \sum_{\mathcal{X}} \exp(-H_Q(\mathcal{X},\mathcal{Y}))$ is a normalization factor. Similarly, I can define the Hamiltonian $H(\mathcal{X},\mathcal{Y})$ of the joint pdf described by the network as

$$P(\mathcal{X},\mathcal{Y}) = e^{-H(\mathcal{X},\mathcal{Y})}. \tag{2.13}$$

From Equation 2.9, using Jensen's inequality [62], one can obtain an *upper bound* on the cost function in the following manner:

$$\begin{aligned} J(\theta) &= -\log \sum_{\mathcal{X}} P(\mathcal{Y},\mathcal{X}|\theta) \\ &= -\log \sum_{\mathcal{X}} Q(\mathcal{X}) \frac{P(\mathcal{Y},\mathcal{X}|\theta)}{Q(\mathcal{X})} \end{aligned}$$

$$
\begin{aligned}
&\leq -\sum_{\mathcal{X}} Q(\mathcal{X}) \log \frac{P(\mathcal{Y},\mathcal{X}|\theta)}{Q(\mathcal{X})} \\
&= -\sum_{\mathcal{X}} Q(\mathcal{X}) \log P(\mathcal{Y},\mathcal{X}|\theta) + \sum_{\mathcal{X}} Q(\mathcal{X}) \log Q(\mathcal{X}) \\
&= \langle H(\mathcal{X},\mathcal{Y}) \rangle - \langle H_Q(\mathcal{X},\mathcal{Y}) \rangle - \log Z_Q \\
&= \mathcal{B}(P,Q,\theta). \qquad (2.14)
\end{aligned}
$$

Terms $\langle H(\mathcal{X},\mathcal{Y}) \rangle$ and $\langle H_Q(\mathcal{X},\mathcal{Y}) \rangle$ denote expectations with respect to distribution Q of Hamiltonians H and H_Q, respectively.

To minimize cost $J(\theta)$ (or maximize $\log Pr(\mathcal{Y}|\theta)$) with respect to θ one can iteratively alternate between the following two optimization steps of the upper bound $\mathcal{B}(P,Q,\theta)$:

```
Initialize $\theta$;
while ( $\Delta J(\theta) > maxError$ ) {
    Expectation (E)     $Q_{new} = \arg\min_Q \mathcal{B}(P,Q,\theta_{old})$;
    Maximization (M)    $\theta_{new} = \arg\min_\theta \mathcal{B}(P,Q_{new},\theta)$;
}
```

Consider first the expectation step E. It is easy to show that the step yields

$$Q_{new}(\mathcal{X}) = Pr(\mathcal{X}|\mathcal{Y},\theta_{old}), \qquad (2.15)$$

the expectation of the hidden variables $\mathcal{X}$ given the observations $\mathcal{Y}$. Also, the upper bound becomes tight, thus guaranteeing no increase in cost. The maximization step then becomes

$$\theta_{new} = \arg\min_\theta \langle H(\mathcal{X},\mathcal{Y}) \rangle_{Pr(\mathcal{X}|\mathcal{Y},\theta_{old})}, \qquad (2.16)$$

where the above expectation is defined over the posterior distribution $Pr(\mathcal{X}|\mathcal{Y},\theta_{old})$. Hence, we arrive at the "classical" version of the EM algorithm, as proposed in [59]. In the expectation step, one does what was earlier referred to as inference. The maximization step (or, actually, minimization) then reduces the cost by adjusting the model's parameters. It was shown in [59] that this procedure (maximization step in particular) is guaranteed not to increase the cost. Therefore, the EM procedure leads to a *stationary point* of the cost function [63]. In many cases this coincides with a local minimum.

Let us return for a moment to the expectation step. I mentioned that the result of optimization in the expectation step leads to the inference problem. In Section 2.2 I concluded that, unfortunately, only a small number of Bayesian networks allow efficient exact inference. The majority of models call for an approximate inference that can be computed more efficiently. How does this approximate inference affect the learning? The answer is simple: it does not! In the expectation, one need not fully optimize the bound, i.e., instead of optimizing over all possible functions Q one may optimize only over a subset of Q, Q_{fixed}, that allows computationally efficient inference:

$$\textbf{Expectation (E)}: \; Q^*_{new} = \arg \min_{Q \in Q_{fixed}} \mathcal{B}(P, Q, \theta_{old}) \tag{2.17}$$

That is exactly what happens in the variational inference approach described in Section 2.2.3. The maximization step, however, remains the same since the terms $\langle H_{Q_new}(\mathcal{X}, \mathcal{Y}) \rangle$ and Z_{Q_new} do not depend on θ:

$$\textbf{Maximization (M)}: \; \theta_{new} = \arg \min_{\theta} \langle H(\mathcal{X}, \mathcal{Y}) \rangle_{Q^*_{new}} . \tag{2.18}$$

Note however that in this case one only optimizes the bound on the cost, not the cost itself.

I have already mentioned that the EM iterative scheme leads to a stationary point of the cost function. Having said that, one realizes that the choice of initial parameter estimates becomes crucial. In most practical cases one often has some knowledge of (initial) model parameters. EM is then used to refine those initial choices. Nevertheless, there are occasions when a good choice of initial parameters is not available. In order not to settle for a "bad" local minimum one has to resort to techniques that will guarantee "acceptable" solutions. One such technique is called the *deterministic annealing variant of the EM algorithm.* Proposed by Ueda and Nakano [64], the approach combines deterministic annealing with the GEM. Briefly, instead of function Q defined in Equation 2.12, one considers a slightly modified Q_β:

$$Q_\beta(\mathcal{X}) = \frac{Q(\mathcal{X}, \mathcal{Y})^\beta}{\sum_{\mathcal{U}} Q(\mathcal{U}, \mathcal{Y})^\beta}, \quad 0 < \beta \leq 1. \tag{2.19}$$

The inverse of factor β is known as the annealing temperature $T_{\text{anneal}} = 1/\beta$. The expression for the bound on the cost function now becomes

$$B_\beta(P, Q, \theta) = \langle H(\mathcal{X}, \mathcal{Y}) \rangle - \frac{1}{\beta} \langle H_Q(\mathcal{X}, \mathcal{Y}) \rangle - \log Z_Q. \tag{2.20}$$

The algorithm basically follows the same steps as the classical GEM. However, at initialization one sets the annealing temperature to some initial high value. The effect of this is that, roughly, many small local minima of the cost function virtually disappear. As one progresses through the EM iterations, the temperature is "slowly" lowered until, for $T_{\text{anneal}} = 1$, the solutions of annealing EM satisfies the original optimization problem. Overall, this modified EM procedure seems to lead toward a global minimum of the cost function. Annealing EM seems particularly effective for GEM algorithms with approximate expectation steps (such as variational inference) [65].

CHAPTER 3

DYNAMIC BAYESIAN NETWORKS

3.1 Introduction

Dynamic Bayesian networks [44] are a special case of singly connected Bayesian networks specifically *aimed at time series modeling.* In this case, one assumes causal dependencies between events in time leading to a simplified network structure, such as the one depicted in Figure 3.1. Namely, in its simplest form, the states of some system described as a DBN satisfy the following Markovian condition:

> The state of a system at time t depends only on its immediate past: its state at time $t-1$.

This is of course the well-known model of Markov chains [66]. In fact, Markov chains are one specific example of DBNs. However, the states of a DBN need not be directly observable. They may influence some other variables that an observer can directly measure. This is indicated in Figure 3.1 by the sets of squares (observables) influenced by the sets of circles (state variables). The model that immediately comes to mind at this point is the *hidden Markov model* (HMM) [5]. Again, as will become more clear

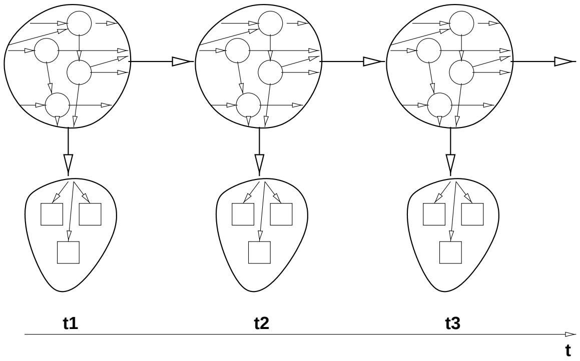


Figure 3.1 General notion of dynamic Bayesian networks. DBNs describe evolution of the states of some system in time in terms of their joint probability distribution. At each instance in time t the states of the system depend only on the states at the previous $(t-1)$ and possibly current time instance.

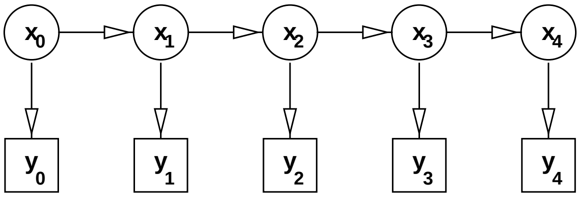


Figure 3.2 Dependency graph of a dynamic Bayesian network (DBN) with one hidden and one observable state at each time instance.

from the sections to follow, an HMM is nothing but a special case of a DBN. Of course, nothing prevents one from assuming that the states of a DBN can take on values from an uncountable, unbounded set such as $\Re$. In fact, they may belong to a vector space $\Re^N$. In that case, an HMM-like DBN may become a classical linear dynamic system (LDS) affected by random noise. Finally, as indicated in Figure 3.1 the state of some system need not be a single, simple state. It may be viewed as a complex structure of interacting states. Each state at one time instance may depend on one or more states at the previous time but also on some current states. Complex structures like this can also be represented as DBNs.

Thus, the previous DBN "definition" is reformulated:

> States of a system at time t depend on their immediate past (states at time $t-1$) and possibly on the current states of their neighbors.

It the rest of this chapter we first present a formal definition of the dynamic Bayesian network model. General inference, decoding, and learning tasks are formalized in this new framework. Finally, a substantial part of the chapter is devoted to two specific examples of DBNs: hidden Markov models and linear dynamic systems.

3.2 Model

The formal model of a DBN is now defined more strictly. To accomplish this we focus our discussion, without loss of generality, on the case of a single hidden state and a single observation at each time instance. This is depicted in Figure 3.2. Given the dependency topology of Figure 3.2, the DBN is a probability distribution function on the sequence of T hidden-state variables $\mathcal{X} = \{x_0, \ldots, x_{T-1}\}$ and the sequence of T observables $\mathcal{Y} = \{y_0, \ldots, y_{T-1}\}$ that has the following factorization:

$$Pr(\mathcal{X}, \mathcal{Y}) = \prod_{t=1}^{T-1} Pr(x_t|x_{t-1}) \cdot \prod_{t=0}^{T-1} Pr(y_t|x_t) \cdot Pr(x_0). \tag{3.1}$$

Clearly, the factorization satisfies the requirements for DBNs that state x_t depend only on state x_{t-1}.

In order to completely specify a DBN one needs to define

- state transition pdf $Pr(x_{t+1}|x_t)$,

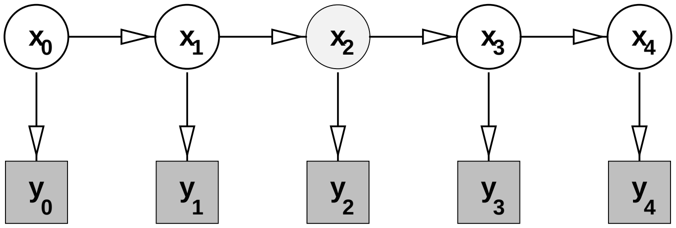


Figure 3.3 Inference in DBNs. Given a sequence of observations one needs to estimate the distribution of hidden variables (shaded circles).

- observation pdf $Pr(y_t|x_t)$, and
- initial state distribution $Pr(x_0)$

for all $t = 0, \ldots, T-1$. All of the conditional pdfs can be time-varying ($Pr(x_{t+1}|x_t) = (x_{t+1}|x_t, t)$) or time invariant, parametric ($Pr(x_{t+1}|x_t) = (x_{t+1}|x_t, \theta)$) or nonparametric (probability tables). Depending on the type of the state space of hidden and observable variables, a DBN can be discrete, continuous, or a combination of the two. For instance, HMMs are usually defined over a set of N discrete hidden states and a set of continuous observations. LDSs are, on the other hand, specified over sets of continuous variables.

As in the case of general Bayesian networks, one may be interested in the following tasks:

- Inference: estimate the pdf of hidden states given some known observations $Pr(\mathcal{X}|\mathcal{Y})$.
- Decoding: find the best sequence of hidden states that may have generated the known sequence of observations.
- Learning: given a number of sequences of observations, estimate parameters of a DBN such that it "best" models the data.

In the sections to follow, we address solutions to the above tasks within the Bayesian network framework.

3.3 Inference

The problem of inference in dynamic Bayesian networks can be posed as the problem of finding $Pr(\mathcal{X}_0^{T-1}|\mathcal{Y}_0^{T-1})$, where $\mathcal{Y}_0^{T-1}$ denotes a finite set of T consecutive observations, $\mathcal{Y}_0^{T-1} = \{y_0, y_2, \ldots, y_{T-1}\}$ and $\mathcal{X}_0^{T-1}$ is the set of the corresponding hidden variables. This is depicted in Figure 3.3. The shaded circle indicates that the distribution of x_t is to be estimated based on observations $\mathcal{Y}_0^{T-1}$.

Depending on the type of DBN, it may be more efficient (or, in fact, only possible) not to estimate the conditional pdf $Pr(\mathcal{X}_0^{T-1}|\mathcal{Y}_0^{T-1})$ for all constellations of $\mathcal{X}_0^{T-1}$ but instead to estimate the pdf's sufficient statistics [62]. Thus, for instance, in the case where the conditional pdf is Gaussian it is sufficient to estimate the mean and variance of x_t, $\langle x_t | \mathcal{Y}_0^{T-1} \rangle$ and $\langle x_t x_t' | \mathcal{Y}_0^{T-1} \rangle$ for every t as well as the covariance $\langle x_t x_{t-1}' | \mathcal{Y}_0^{T-1} \rangle$. Similarly, when x_t is discrete and the conditional pdf is given as a table of probabilities, one needs to find $Pr(x_t | \mathcal{Y}_0^{T-1})$ (which is the same as $\langle x_t | \mathcal{Y}_0^{T-1} \rangle$) and $Pr(x_t x_{t-1}' | \mathcal{Y}_0^{T-1})$. We will return to this discussion in more detail when we revisit the DBN flavor of HMMs and LDSs. Since all of the above statistics can in principle be derived from $Pr(x_t | \mathcal{Y}_0^{T-1})$ and $Pr(x_t x_{t-1} | \mathcal{Y}_0^{T-1})$, we focus the rest of my general DBN inference discussion on those two quantities.

As in the case of general singly connected Bayesian networks (see Section 2.2.2), an efficient forward-backward algorithm can be employed for this purpose. Namely, a two-step process is needed to accomplish the inference task: propagation of probabilities in the forward direction (direction of time) and then the backward propagation. The two steps are formulated in the following section.

3.3.1 Forward propagation

Let $\alpha_t(x_t)$ be the forward probability distribution defined as the joint probability of observations up until time t and the state at time t:

$$\alpha_t(x_t) = Pr(\mathcal{Y}_0^t, x_t). \tag{3.2}$$

Given the network topology of Figure 3.3 it is easy to see that

$$\alpha_{t+1}(x_{t+1}) = Pr(y_{t+1} | x_{t+1}) \sum_{x_t} Pr(x_{t+1} | x_t) \alpha_t, \tag{3.3}$$

with the initial condition $\alpha_0(x_0) = Pr(x_0)$.

One "by-product" of forward propagation is the likelihood of the observation data sequence $\mathcal{Y}_0^{T-1}$. From the definition of the forward factor *alpha* in Equation 3.2 it follows that

$$Pr(\mathcal{Y}_0^{T-1}) = \frac{\alpha_{T-1}(x_{T-1})}{\sum_{\xi_{T-1}} \alpha_{T-1}(\xi_{T-1})}. \tag{3.4}$$

Hence, the probability of the observation sequence is proportional to the forward factor of the last hidden state. This probability is useful when one needs to determine how well different DBN models "fit" a data sequence in the framework of maximum likelihood estimation.

3.3.2 Backward propagation

Let $\beta_t(x_t)$ be the backward probability distribution, i.e., the conditional probability of observations from time $t+1$ until the last observation at time $T-1$ conditioned on the values of the state at time t:

$$\beta_t(x_t) = Pr(\mathcal{Y}_{t+1}^{T-1} | x_t). \tag{3.5}$$

Then, the following recursive relationship holds:

$$\beta_{t-1}(x_{t-1}) = \sum_{x_t} \beta_t(x_t) Pr(x_t | x_{t-1}) Pr(y_t | x_t), \tag{3.6}$$

with $\beta_T(x_{T-1}) = 1$ final condition.

3.3.3 Smoothing

Given the expressions for forward and backward probability propagation, it easily follows that

$$\gamma_t(x_t) = Pr(x_t|\mathcal{Y}_0^{T-1}) = \frac{\alpha_t(x_t)\beta_t(x_t)}{\sum_{x_t} \alpha_t(x_t)\beta_t(x_t)}, \tag{3.7}$$

where $\gamma_t(x_t)$ is the smoothing operator. One can also derive higher-order smoothing equations. In particular, a first-order smoothing is defined as

$$\xi_{k,k-1}(x_t, x_{t-1}) = Pr(x_t, x_{t-1}|\mathcal{Y}_0^{T-1}) = \frac{\alpha_{t-1}(x_{t-1})Pr(x_t|x_{t-1})Pr(y_t|x_t)\beta_t(x_t)}{\sum_{x_t} \alpha_t(x_t)\beta_t(x_t)}. \tag{3.8}$$

3.3.4 Prediction

Another interesting inference problem deals with predicting future observation or hidden states based on the past observation data. Namely, a one-step prediction can be stated as the following inference problem:

$$Pr(x_{t+1}|\mathcal{Y}_0^t)$$

or

$$Pr(y_{t+1}|\mathcal{Y}_0^t).$$

It is easy to show that

$$Pr(x_{t+1}|\mathcal{Y}_0^t) = \frac{\sum_{x_t} Pr(x_{t+1}|x_t)\alpha_t(x_t)}{\sum_{x_t} \alpha_t(x_t)}. \tag{3.9}$$

Similarly,

$$Pr(y_{t+1}|\mathcal{Y}_0^t) = \frac{\sum_{x_{t+1}} \alpha_{t+1}(x_{t+1})}{\sum_{x_t} \alpha_t(x_t)}. \tag{3.10}$$

Instead of considering the pdfs themselves, it is sometimes more convenient to express the prediction problem in terms of the expected or maximum likelihood estimates:

$$\begin{aligned}
\langle x_{t+1,t}\rangle &= E[x_{t+1}|\mathcal{Y}_0^t] \\
x_{t+1,t_{ML}} &= \arg\max_{x_{t+1}} Pr(x_{t+1}|\mathcal{Y}_0^t) \\
\langle y_{t+1,t}\rangle &= E[y_{t+1}|Y_t] \\
y_{t+1,t_{ML}} &= \arg\max_{y_{t+1}} Pr(y_{t+1}|\mathcal{Y}_0^t).
\end{aligned}$$

3.3.5 Decoding

The goal of sequence decoding in dynamic Bayesian networks is to find the most likely sequence of hidden variables given the observations

$$\mathcal{X}^{*T-1}_0 = \arg\max_{\mathcal{X}_0^{T-1}} Pr(\mathcal{X}_0^{T-1}|\mathcal{Y}_0^{T-1}). \tag{3.11}$$

This task can be achieved using the dynamic programming Viterbi algorithm [67]. Let

$$\delta_{t+1}(x_{t+1}) = \max_{\mathcal{X}_0^t} Pr(\mathcal{X}_0^{t+1}, \mathcal{Y}_0^{t+1}). \tag{3.12}$$

Given the topology of the DB network,

$$\begin{aligned}\delta_{t+1}(x_{t+1}) &= Pr(y_{t+1}|x_{t+1}) \cdot \max_{x_t} \left[Pr(x_{t+1}|x_t) \cdot \max_{\mathcal{X}_0^{t-1}} Pr(\mathcal{X}_0^t, \mathcal{Y}_0^t) \right] & (3.13)\\ &= Pr(y_{t+1}|x_{t+1}) \cdot \max_{x_t} \left[Pr(x_{t+1}|x_t) \cdot \delta_t(x_t) \right]. & (3.14)\end{aligned}$$

It now readily follows that

$$\max_{\mathcal{X}_0^{T-1}} Pr(\mathcal{X}_0^{T-1}|\mathcal{Y}_0^{T-1}) = \max_{x_{T-1}} \delta_{T-1}(x_{T-1}). \tag{3.15}$$

To find ${\mathcal{X}^*}_0^{T-1}$ one also needs to keep track of the argument x_t that maximizes $\delta_{t+1}(x_{t+1})$,

$$\psi_{t+1}(x_{t+1}) = \arg\max_{x_t} \left[Pr(x_{t+1}|x_t) + \delta_t(x_t) \right], \tag{3.16}$$

and then trace back:

$$x_t^* = \psi_{t+1}(x_{t+1}^*). \tag{3.17}$$

For a sequence to be decoded using the Viterbi algorithm it is necessary to first receive a complete set of observations $y_1, \ldots, y_T$. However, for on-line decoding it is inconvenient to wait for a complete sequence of observations to be received before proceeding with the backtracking. A suboptimal solution known as the *truncated Viterbi algorithm* engages backtracking every time after a fixed number of $T_{max} << T$ observations is received. Assuming that T_{max} is sufficiently large, the suboptimally decoded sequence becomes close enough to the optimal ${\mathcal{X}^*}_0^{T-1}$.

3.4 Learning

The learning algorithm for dynamic Bayesian networks follows directly from the EM or GEM algorithm described in Section 2.3. However, the expression for the joint probability distribution now assumes a specific form

$$\log Pr(\mathcal{X}_0^{T-1}, \mathcal{Y}_0^{T-1}|\theta) = \sum_{t=1}^{T-1} \log Pr(x_t|x_{t-1}) + \sum_{t=0}^{T-1} \log Pr(y_t|x_t) + \log Pr(x_0),$$

where θ is the model parameter vector. The maximization step now finds parameters θ that satisfy

$$\begin{aligned}&\frac{\partial \mathcal{B}(P, Q^*)}{\partial \theta} = \\ &\sum_{t=1}^{T-1} \left\langle \frac{\partial \log Pr(x_t|x_{t-1})}{\partial \theta} \right\rangle + \sum_{t=0}^{T-1} \left\langle \frac{\partial \log Pr(y_t|x_0)}{\partial \theta} \right\rangle \\ &+ \left\langle \frac{\partial \log Pr(x_0)}{\partial \theta} \right\rangle = 0, \end{aligned} \tag{3.18}$$

where the expectation is as defined in Section 2.3. Equivalently, a gradient-based learning procedure can be implemented that utilizes $\frac{\partial \mathcal{B}(P,Q^*)}{\partial \theta}$.

Two types of dynamic Bayesian networks are often encountered in the literature: linear dynamic systems and hidden Markov models. Both types posses the same DBN topology (as shown in Figure 3.3), but they differ with respect to the underlying spaces over which the hidden variables are defined.

3.5 Linear Dynamic Systems and Continuous-State Dynamic Bayesian Networks

Continuous-state DBNs are well known in the theory of linear systems. However, only recently have the linear systems been studied as a special case of dynamic Bayesian networks [68, 44]. Consider the following state space equation:

$$\begin{aligned} x_{t+1} &= A_{t+1}x_t + v_{t+1} && (3.19) \\ y_t &= C_t x_t + w_t && (3.20) \\ x_0 &= v_0, && (3.21) \end{aligned}$$

where $x_t \in \Re^N$, $y_t \in \Re^M$, v_t has a zero-mean Gaussian distribution with variance Q_t, and w_t has a zero-mean Gaussian distribution with variance R_t, for all $t = 0, 1, \ldots$. Assume furthermore that x_0 has a Gaussian distribution with mean μ and variance Q_0. It is then easy to show that any x_t and y_t are also distributed normally according to the following distributions:

$$\begin{aligned} Pr(x_{t+1}|x_t) &= \mathcal{N}(x_{t+1}, A_{t+1}x_t, Q_t) \\ &= (2\pi)^{-\frac{N}{2}}|Q_t|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(x_{t+1} - A_{t+1}x_t)^t Q_{t+1}^{-1}(x_{t+1} - A_{t+1}x_t)\right\}, \qquad (3.22) \end{aligned}$$

and

$$\begin{aligned} Pr(y_t|x_t) &= \mathcal{N}(y_t, Cx_t, R_t) \\ &= (2\pi)^{-\frac{M}{2}}|R_t|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(y_t - Cx_t)^t R_t^{-1}(y_t - Cx_t)\right\}. \qquad (3.23) \end{aligned}$$

Obviously, this leads one to the typical dynamical Bayesian network topology from Figure 3.3. The formulation of an LDS presented above is for a time-varying system. However, the rest of this section focuses on time-invariant LDSs.

Given the above definition of the transition and observation pdfs one can now write an expression for a joint pdf that is defined by a Gaussian LDS. We choose, equivalently, to define its Hamiltonian:

$$H = \frac{1}{2}\sum_{t=1}^{T-1} (x_t - Ax_{t-1})' \, Q^{-1} \, (x_t - Ax_{t-1}) + \frac{T-1}{2} \log|Q|$$

$$
\begin{aligned}
&+\frac{1}{2}\left(x_0-\mu\right)' {Q_0}^{-1}\left(x_0-\mu\right)+\frac{1}{2}\log|Q_0|+\frac{NT}{2}\log 2\pi \\
&+\frac{1}{2}\sum_{t=0}^{T-1}\left(y_t-Cx_t\right)' R^{-1}\left(y_t-Cx_t\right)+\frac{T}{2}\log|R|+\frac{MT}{2}\log 2\pi. \qquad (3.24)
\end{aligned}
$$

The expression of this form is particularly useful when dealing with the model parameter updates. We will use the Hamiltonian form to derive the EM-based parameter update equations of Section 3.5.2.

In the case of LDSs with independent and identically distributed (iid) Gaussian noise processes, as mentioned in Section 3.3, it is not necessary to consider the actual expressions for inference pdfs. Namely, sufficient statistics are sufficient to completely determine the stochastic behavior of the network. One can show [69] that the following first- and second-order statistics are in fact sufficient for an LDS with iid Gaussian noise:

- $\left\langle x_t|\mathcal{Y}_0^{T-1}\right\rangle$,
- $\left\langle x_t x_t'|\mathcal{Y}_0^{T-1}\right\rangle$, and
- $\left\langle x_t x_{t-1}'|\mathcal{Y}_0^{T-1}\right\rangle$,

for $t = 0, \ldots, T-1$. The rest of the discussion of LD systems as DBNs thus focuses on the calculation and use of these sufficient statistics.

3.5.1 Inference

Inference in continuous-state DBNs directly follows from the general inference framework of DBNs (see Section 3.3). We next consider the problems of forward propagation, smoothing, prediction, and decoding. Backward propagation is excluded from the discussion because it represents an integral part of smoothing. Even though it is possible to formulate backward propagation on its own [69], its fusion with smoothing yields a more clear set of DBN equations.

3.5.1.1 Forward propagation

Forward probability propagation in LD systems requires calculation of the following statistics:

- $\langle x_t|\mathcal{Y}_0^t\rangle$,
- $\langle x_t x_t'|\mathcal{Y}_0^t\rangle$, and
- $\langle x_t x_{t-1}'|\mathcal{Y}_0^t\rangle$,

Obviously, these statistics are elements of the well-known Kalman filtering equations [70]

$$
\langle x_t|\mathcal{Y}_0^t\rangle \;=\; \left\langle x_t|\mathcal{Y}_0^{t-1}\right\rangle + K_t^f\left(y_t - C\left\langle x_t|\mathcal{Y}_0^{t-1}\right\rangle\right) \qquad (3.25)
$$

$$\Sigma_{t,t} = \Sigma_{t,t-1} - K_t^f C \Sigma_{t,t-1} \tag{3.26}$$

$$\left\langle x_t | \mathcal{Y}_0^{t-1} \right\rangle = A \left\langle x_{t-1} | \mathcal{Y}_0^{t-1} \right\rangle \tag{3.27}$$

$$\Sigma_{t,t-1} = A \Sigma_{t,t} A' + Q \tag{3.28}$$

$$K_t^f = \Sigma_{t,t-1} C' \left(C \Sigma_{t,t-1} C' + R \right)^{-1} , \tag{3.29}$$

where

$$\begin{aligned} \Sigma_{t,t} &= \left\langle (x_t - \left\langle x_t | \mathcal{Y}_0^t \right\rangle)(x_t - \left\langle x_t | \mathcal{Y}_0^t \right\rangle)' | \mathcal{Y}_0^t \right\rangle \\ \Sigma_{t,t-1} &= \left\langle (x_t - \left\langle x_t | \mathcal{Y}_0^{t-1} \right\rangle)(x_t - \left\langle x_t | \mathcal{Y}_0^{t-1} \right\rangle)' | \mathcal{Y}_0^{t-1} \right\rangle . \end{aligned}$$

At this point it is useful to define the expression for likelihood of the observed data. It follows from the theory of Kalman filters that

$$\begin{aligned} - \log Pr(\mathcal{Y}_0^{T-1}) = \frac{1}{2} T \log(2\pi) \\ + \frac{1}{2} \sum_{t=0}^{T-1} \left((y_t - \langle x_t \rangle)' \, {\Sigma_{t,t-1}}^{-1} \left(y_t - \langle x_t \rangle \right) + \log \left| \Sigma_{t,t-1} \right| \right) . \end{aligned} \tag{3.30}$$

3.5.1.2 Smoothing

Smoothing or inference equations for LDSs can be obtained from the above forward equations and a similar set of backward equations [71, 72]. However, it is also possible to solve the smoothing problem directly from the forward equations [73]. The set of equations defines the Rauch-Tung-Streibel (RTS) smoother:

$$K_{t-1}^s = \Sigma_{t-1,t-1} A' \Sigma_{t,t-1}^{-1} \tag{3.31}$$

$$\left\langle x_{t-1} | \mathcal{Y}_0^{T-1} \right\rangle = \left\langle x_{t-1} | \mathcal{Y}_0^{t-1} \right\rangle + K_{t-1}^s \left(\left\langle x_t | \mathcal{Y}_0^{T-1} \right\rangle - A \left\langle x_t | \mathcal{Y}_0^{t-1} \right\rangle \right) \tag{3.32}$$

$$\Sigma_{t-1,T-1} = \Sigma_{t-1,t-1} + K_{t-1}^s \left(\Sigma_{t,T-1} - \Sigma_{t,t-1} \right) {K_{t-1}^s}' . \tag{3.33}$$

Again, the smoothed variance $\Sigma_{t,T-1}$ is defined as

$$\Sigma_{t,T-1} = \left\langle (x_t - \left\langle x_t | \mathcal{Y}_0^{T-1} \right\rangle)(x_t - \left\langle x_t | \mathcal{Y}_0^{T-1} \right\rangle)' | \mathcal{Y}_0^{T-1} \right\rangle .$$

To complete the sufficient statistics necessary for inference, We need to define the expression for the variance of hidden states across two consecutive times steps. It can be shown that [74]

$$\begin{aligned} &\langle (x_t - \langle x_t \rangle)(x_{t-1} - \langle x_{t-1} \rangle)' \rangle = \\ &\quad \Sigma_{t,t} {K_{t-1}^s}' + K_t^s \left(\langle (x_{t+1} - \langle x_{t+1} \rangle)(x_t - \langle x_t \rangle)' \rangle - A \Sigma_{t,t} \right) {K_{t-1}^s}' . \end{aligned}$$

In the rest of this work, unless explicitly mentioned otherwise, We use a shortened notation of $\langle x_t \rangle$ to denote the smoothed estimate $\left\langle x_t | \mathcal{Y}_0^{T-1} \right\rangle$. Similar notation rules are followed for other estimates of sufficient statistics.

3.5.1.3 Decoding

Decoding of the "best" sequence as defined in Section 3.3.5 trivially reduces to the sequence of expectations:

$$\mathcal{X}^*{}_0^{T-1} = \{\langle x_t|\mathcal{Y}_0^t\rangle, t = 0, \ldots, T-1\}. \tag{3.34}$$

As will be seen in the section on discrete DBNs (hidden Markov models), the Viterbi estimates in general do not coincide with the filtered ones.

3.5.2 Learning

Learning of Kalman filter/smoother parameters $\theta = (A, Q, C, R, \mu, Q_0)$ is usually less emphasized in time series prediction literature. The reason for this is that often parameters of the system are known from physical models and measurements. Nevertheless, estimating the system parameters from data can in fact be useful.

Consider the case of a time-invariant dynamic system, where $\theta = (A, Q, C, R, \mu, Q_0)$ is independent of time. Given the form of the probability distribution defined by an LDS or, equivalently, its Hamiltonian (see Equation 3.24) the update equations are found by setting the partial derivatives of $\langle H\rangle$ with respect to elements of θ to zero. For the case of a single data sequence this yields the following set of equations:

$$A_{new} = \left(\sum_{t=1}^{T-1} \langle x_t x_{t-1}{}'\rangle\right)\left(\sum_{k=1}^{T-1} \langle x_t x_t{}'\rangle\right)^{-1} \tag{3.35}$$

$$Q_{new} = \frac{1}{T-1}\left(\sum_{t=1}^{T-1} \langle x_t x_{t-1}{}'\rangle - A_{new}\sum_{t=1}^{T-1} \langle x_{t-1} x_t{}'\rangle\right) \tag{3.36}$$

$$C_{new} = \left(\sum_{t=0}^{T-1} y_t \langle x_t{}'\rangle\right)\left(\sum_{t=0}^{T-1} \langle x_t x_t{}'\rangle\right)^{-1} \tag{3.37}$$

$$R_{new} = \frac{1}{T}\sum_{t=0}^{T-1} \left(y_t y_t{}' - C_{new}\langle x_t\rangle y_t{}'\right) \tag{3.38}$$

$$\mu_{new} = \langle x_0\rangle \tag{3.39}$$

$$Q_{0,new} = \langle x_0 x_0{}'\rangle - \langle x_0\rangle\langle x_0\rangle{}'. \tag{3.40}$$

Note that the above expressions are defined with respect to the sufficient statistics evaluated on the model with parameter values before the update (as imposed by the EM algorithm).

The question naturally arises how to select some good initial parameter values to start the recursion. There are several possible answers to that question. It is usually the case that one has some initial guess of what the parameters should be. It is also possible that one only needs to "refine" the model he has and fit it to a new set of data. Finally, one may have no guess of what the parameters are. In that case one usually resorts to initialization of more complex models using models that are one level simpler. In the case of LDSs one may assume that the simpler model contains no temporal dependencies between hidden states x. One can then use *factor analysis* [44] to initialize such a model and infer its states. From there one can employ the LDS update equations to initialize the estimates of LDS parameters.

3.6 Hidden Markov Models and Discrete-State Dynamic Bayesian Networks

Discrete-state DBNs are also known as hidden Markov models or HMMs. Hidden Markov models have been used successfully for more than a decade in the field of automatic speech recognition (ASR) [5]. In this section we present the prospect of HMMs from the point of view of their DBN nature.

As mentioned before, an HMM is a DBN with a discrete-valued state space. Nevertheless, the dependency graph of an HMM is still the same as the general dependency graph of a dynamic Bayesian net (see Figure 3.2). Hence, the joint pdf defined by this model is clearly the one of Equation 3.1. The HMM becomes a distinct model by defining a unique family of state transition pdfs.

3.6.1 Notation

Unlike LDS models of the previous section, HMMs are usually specified directly by defining the necessary conditional probabilities. Consider an HMM with N discrete hidden states. We denote the state variables of this model by s in order to differentiate them from the continuous-valued states of an LDS. The state space of an HMM can then be defined in two equivalent ways. One is to assign each state a different integer from a set of N different integers:

$$\mathcal{S} = \{0, 1, \ldots, N-1\}.$$

Each s_t can now take on values from this set. A slightly different state space (yet mathematically equivalent) can be constructed by assigning one unit vector of dimension N to each different state. In other words,

$$\mathcal{S} = \{e_0, \ldots, e_{N-1}\},$$

where e_i denotes the unit vector with a nonzero element in the ith position. Even though the indexed notation tends to prevail in the classical HMM literature, We often use the “vector”-type notation. The reason for that will become more clear as the model is developed further.

Consider now the state transition pdf $Pr(s_t|s_{t-1})$. Assume that the HMM in question is time invariant. Hence, the transition pdf now becomes an $N \times N$ probability table (matrix) P whose entries are

$$P(i,j) = Pr(s_t = i|s_{t-1} = j),$$

or equivalently

$$P(i,j) = Pr(s_t = e_i | s_{t-1} = e_j),$$

where $i, j = 0, \ldots, N-1$. If we use the vector notation, the following expression holds:

$$Pr(s_t = e_i | s_{t-1} = e_j) = {s_t}' P s_{t-1}$$

where P is the transition pdf matrix. The same notation holds when one leaps into the log space: $\log Pr(s_t = e_i | s_{t-1} = e_j) = {s_t}' \log P s_{t-1}$, where $\log P$ denotes an element-wise operation.

Note another important implication of this notation, namely, that the following vector and matrix identities are easily shown to be true:

$$\begin{aligned} \langle s_t \rangle &= [Pr(s_t = e_0) \cdots Pr(s_t = e_{N-1})]' = Pr(s_t), \\ \langle s_t s_t' \rangle &= \mathrm{diag}(\langle s_t \rangle). \end{aligned}$$

The observation pdf for an HMM, $Pr(y_t | s_t)$, can be defined in a number of ways. Next, three cases are considered that are most commonly found in the literature.

3.6.2 Discrete observation HMM

In the discrete observation HMM, observation state variables can take on values from one of M different states $\mathcal{Y} = \{e_0, \ldots, e_{M-1}\}$. Consequently, one can define an observation probability table (matrix) as

$$P_o(i,j) = Pr(y_t = e_i | s_t = e_j),$$

with $i = 0, \ldots, M-1,\ j = 0, \ldots, N-1$. Equivalently, $Pr(y_t = e_i | s_t = e_j) = {y_t}' P_o s_t$.

The joint pdf defined by this model is defined by the following Hamiltonian:

$$H = \sum_{t=1}^{T-1} {s_t}' \log P s_{t-1} + \sum_{t=0}^{T-1} {y_t}' P_o s_t + s_0 \log \pi_0. \tag{3.41}$$

3.6.3 Gaussian observation HMM

Observation states are distributed according to Gaussian distribution with means and variances determined by the conditioning hidden states. In other words,

$$Pr(y_t | s_t = e_i) = \mathcal{N}(C_i, R_i),$$

where C_i and R_i belong to the sets of predefined means and variances. Let $s_t(j)$ denote the jth component of state s_t. Then one can also write

$$Pr(y_t | s_t = e_i) = \sum_{j=0}^{N-1} \mathcal{N}(c_j, R_j) \cdot s_t(j),$$

since all $s_t(j)$ are zero except for $s_t(i)$. Finally, if all states have identical variances $R_0 = \ldots = R_{N-1} = R$ a compact notation of the following form holds:

$$Pr(y_t|s_t = e_i) = \mathcal{N}(C \cdot s_t, R),$$

where C is a matrix whose columns are the means associated with the hidden states $C = [C_0 \cdots C_{N-1}]$.

The model's Hamiltonian for the case of all different observation means and variances now takes the form

$$\begin{aligned} H = & \sum_{t=1}^{T-1} s_t{}' \log P s_{t-1} + s_0 \log \pi_0 \\ & + \frac{1}{2} \sum_{t=0}^{T-1} \sum_{i=0}^{N-1} \left(\log |R_i| + (y_t - C_i)' \, R_i{}^{-1} \, (y_t - C_i) \right) s_t(i) + \frac{1}{2} NT \log 2\pi. \end{aligned}$$

3.6.4 Mixture-of-Gaussians observation HMM

Modeling the observation pdf as a mixture of Gaussians allows one, at least in principle, to model any desired distribution on the observation space. Consider the case of a mixture with S components. Denote with $P_m(j,i), j = 0, \ldots, S-1$ the mixing weights (or mixture probabilities) of the ith hidden state. Furthermore, assume that each state's mixture has a different set of mixture component means $C_{i,j}$ and $R_{i,j}$, where index j represents the mixture component and index i denotes the hidden state. Without loss of generality, let all the hidden states have the same number of mixture components. One can then say that

$$Pr(y_t|s_t = e_i, c_t = e_j) = \mathcal{N}(C_{ij}, R_{ij}) = \sum_{k=0}^{N-1} \sum_{l=0}^{S-1} \mathcal{N}(C_{k,l}, R_{k,l}) s_t(k) c_t(l),$$

where c_t is the state variable indicating the "state of the mixture" at time t. If one thinks of a generative model defined by this HMM, then s_t denotes the active state at time t and c_t denotes the active observation mixture once the system is in state s_t. Clearly, given that s_t, c_t has the distribution defined by the s_tth column of P_m,

$$Pr(c_t|s_t = e_j) = P_m s_t.$$

This fact is depicted by the modified DBN of Figure 3.4.

The Hamiltonian of an HMM with the observation pdf defined as a mixture of S Gaussians then finally becomes

$$\begin{aligned} H = & \sum_{t=1}^{T-1} s_t{}' \log P s_{t-1} + s_0 \log \pi_0 \\ & + \frac{1}{2} \sum_{t=0}^{T-1} \sum_{i=0}^{N-1} \sum_{j=0}^{S-1} \left\{ \log |R_{i,j}| + (y_t - C_{i,j})' \, R_{i,j}{}^{-1} \, (y_t - C_{i,j}) \right\} c_t(j) s_t(i) \\ & + \frac{1}{2} NST \log 2\pi + \sum_{t=0}^{T-1} c_t{}' \log P_m s_t. \end{aligned} \tag{3.42}$$

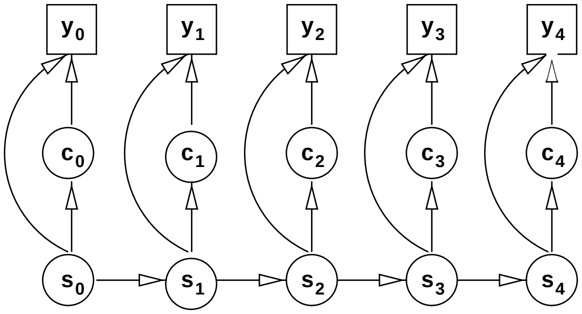


Figure 3.4 Dependency graph of a hidden Markov model with a mixture of observation pdfs. Variable c_t specifies which observation distribution mixture component is active at time t. The observation y_t depends both on the hidden state x_t as well as the mixture variable c_t. The model is often used to define an HMM with a mixture of Gaussian observation pdfs.

3.6.5 A different state-space model

It is also feasible to define an alternative state-space model associated with an HMM [44]. This state-space model is similar to the one for the DBN. However, the HMM defined in such space is nonlinear:

$$\begin{aligned} s_{t+1} &= \mathrm{WTA}(A_{t+1}s_t + w_{t+1}) && (3.43) \\ y_t &= C_t s_t + v_t, && (3.44) \end{aligned}$$

where WTA is the *winner-takes-all* nonlinear operator: its output is a unit vector with nonzero in the position of a largest component of the operator's argument. As before, the space of s_t is the set of all unit vectors of dimensions N. In fact, one can show that it is possible to go between the two alternative representations of HMMs, i.e., to find $Pr(x_t|x_{t-1})$ from A_t and Q_t and vice versa [44].

3.6.6 Inference

Inference in HMMs follows readily from the general inference in DBNs. As HMMs are directly specified by probability tables and/or Gaussian (or mixture-of-Gaussians) pdfs, forward, backward, and smoothing equations are simply those found in Section 3.3. Here, the expressions are presented again in accordance with the adopted HMM notation.

3.6.6.1 Forward propagation

Let α_t be an N-dimensional vector whose components are $\alpha_t(i) = Pr(s_t = e_i, \mathcal{Y}_0^t)$, $i = 1, \ldots, N$. Then,

$$\alpha_t = \mathrm{diag}(Pr(y_t)) \cdot P \cdot \alpha_{t-1}, \qquad (3.45)$$

where $Pr(y_t)$ is an N dimensional vector with the element i equal to $Pr(y_t|s_t = e_i)$.

3.6.6.2 Backward propagation

Let β_t be an N-dimensional vector whose components are $\beta_t(i) = Pr(\mathcal{Y}_{t+1}^{T-1}|s_t = e_i)$, $i = 1, \ldots, N$. Then,

$$\beta_t = P' \cdot \text{diag}(Pr(y_{t+1})) \cdot \beta_{t+1}. \tag{3.46}$$

3.6.6.3 Smoothing

Let γ_t be an N-dimensional vector whose components are $\gamma_t(i) = Pr(s_t = e_i|\mathcal{Y}_0^{T-1})$, $i = 1, \ldots, N$. Then,

$$\gamma_t = \frac{\text{diag}(\alpha_t)\beta_t}{\alpha_t{}'\beta_t}. \tag{3.47}$$

Keeping in mind the notation introduced in Section 3.6.1, one can also write that

$$\gamma_t = \langle s_t \rangle .$$

We will use this notation throughout the rest of this work.

The joint probability of two consecutive hidden states can also be easily found. Let ξ_t be an $N \times N$ matrix whose components are $\xi_t(i,j) = Pr(s_t = e_i, s_{t+1} = e_j|\mathcal{Y}_0^{T-1})$. Then,

$$\xi_t = \text{diag}(\alpha_t) \cdot P_A{}' \cdot \text{diag}\left(\text{diag}(Pr(y_{t+1}))\beta_{t+1}\right). \tag{3.48}$$

Again, one can equivalently write

$$\xi_t = \left\langle s_t s_{t+1}' \right\rangle .$$

In the case of mixture-of-Gaussians observation HMMs, another quantity is of interest. That is the joint distribution of the hidden system states and the states of the mixture components, $Pr(s_t = e_i, c_t = e_j|\mathcal{Y}_0^{T-1})$. It is trivial to show that these statistics can be obtained as

$$Pr(s_t = e_i, c_t = e_j|\mathcal{Y}_0^{T-1}) = Pr(s_t = e_i|\mathcal{Y}_0^{T-1})Pr(c_t = e_j|s_t = e_i, y_t). \tag{3.49}$$

Of course, $Pr(s_t = e_i|\mathcal{Y}_0^{T-1}) = \gamma_t(i)$ and

$$Pr(c_t = e_j|s_t = e_i, y_t) = \left. \frac{P_m(j,i)\mathcal{N}(C_{i,j}, R_{i,j})}{\sum_{k=0}^{S-1} P_m(j,i)\mathcal{N}(C_{i,j}, R_{i,j})} \right|_{y_t}$$

is the proportion that the $j$the mixture component of the ith state "contributes" to the observation y_t, or in other words, the probability that y_t came from mixture component j in state i.

3.6.6.4 Prediction

Prediction equations are not commonly seen in the classical HMM literature. However, they can be readily obtained from equations in Section 3.3.4 and Section 3.6.6.1:

$$P_{k+1,k}^s = \frac{P\alpha_k}{\mathbf{1}'\alpha_t} \tag{3.50}$$

$$P^y_{k+1,k} = \frac{\mathbf{1}'\alpha_{t+1}}{\mathbf{1}'\alpha_t}, \quad (3.51)$$

where $P^x_{k+1,k}(i) = Pr(s_{t+1} = e_i|\mathcal{Y}_0^t)$, $P^y_{k+1,k} = Pr(y_{t+1}|\mathcal{Y}_0^t)$, and $\mathbf{1}$ is an N-dimensional vector of all ones. For continuous observation densities, the expected value estimate can be shown to be

$$y^e_{k+1,k} = CP^x_{k+1,k}. \quad (3.52)$$

3.6.6.5 Decoding

Application of the Viterbi algorithm from Section 3.3.5 leads to

$$\delta_t = \text{diag}(Pr(y_t)) \max_{columns} \left[P \text{ diag}(\delta_{t-1})\right], \quad (3.53)$$

where δ_t is an N-dimensional vector with elements $\delta_t(i) = \max_{\mathcal{S}_0^{t-1}} Pr(s_t = i, \mathcal{S}_0^{t-1}, \mathcal{Y}_0^t)$. Similarly,

$$\psi_{t+1} = \arg \max_{columns} \left[Pr(s_{t+1}|s_t)\text{diag}(\delta_t(s_t))\right], \quad (3.54)$$

and

$$s^*_t = \psi_{t+1}(\check{x}_{t+1}) \quad (3.55)$$

with $s^*_{T-1} = \arg\max \delta_{T-1}$.

3.6.7 Learning

Batch learning in HMMs has been studied extensively in the classical HMM framework [5]. This section considers the problem of parameter learning for the case of a mixture-of-Gaussians observation HMM. The mixture-of-Gaussians observation HMM is a generalization of both the single Gaussian observation and the discrete observation models. Hence, the parameter update equations of the mixture-of-Gaussians model parameters can be easily specialized for the simpler models.

As before, we consider the parameter update equations that are the consequence of the EM learning on a single data sequence. Hence, one needs to find partial derivatives of the mean Hamiltonian in Equation 3.42 with respect to the model parameters: transition pdf matrix P, mixture distribution matrix P_m, and $N \times S$ mixture parameters $C_{i,j}$ and $R_{i,j}$. Given the simple form of Equation 3.42, it readily follows that

$$P(i,j) = \frac{\sum_{t=1}^{T-1} \langle s_t(i)s_{t-1}(j)\rangle}{\sum_{t=1}^{T-1} \langle s_{t-1}(j)\rangle} \quad (3.56)$$

$$P_m(k,i) = \frac{\sum_{t=0}^{T-1} \langle c_t(k)s_t(i)\rangle}{\sum_{t=0}^{T-1} \langle s_t(i)\rangle} \quad (3.57)$$

$$C_{k,i} = \frac{\sum_{t=0}^{T-1} y_t \langle c_t(k)s_t(i)\rangle}{\sum_{t=0}^{T-1} \langle c_t(k)s_t(i)\rangle} \quad (3.58)$$

$$R_{k,i} = \frac{\sum_{t=0}^{T-1}(y_t - C_{k,i})(y_t - C_{k,i})' \langle c_t(k)s_t(i)\rangle}{\sum_{t=0}^{T-1} \langle c_t(k)s_t(i)\rangle}, \quad (3.59)$$

where $i = 0, \ldots, N-1$ and $k = 0, \ldots, S-1$. In the case of a single Gaussian observation $S = 1$, all statistics of the form $\langle c_t(k) s_t(i) \rangle$ become identical to $\langle s_t(i) \rangle$. Obviously, $P_m(k,i) = 1$ for all model states i. The discrete observation case parameter updates are also easy to deduce. One simply needs to assume that all variances $R_{k,i}$ are infinitesimally small, whereas means $C_{k,i}$ take on values of integers k or unit vectors e_k. The observation probability table is then simply identical to the mixture probability matrix P_m, and c_t can be thought of as the observation state variables:

$$P_o(k,i) = \frac{\sum_{t=0}^{T-1} \langle y_t(k) s_t(i) \rangle}{\sum_{t=0}^{T-1} \langle s_t(i) \rangle}.$$

Besides the outlined batch learning approach, on-line gradient learning schemes (see Section 2.3), can also be employed for HMM parameter updates. Nevertheless, such schemes are not as common as the batch approach. Examples of on-line parameter updates for HMMs can found in [68].

3.6.8 Bayesian network representation and state transition diagrams

At this point it may be useful to draw a parallel between two different graphical representations found in HMM literature. One is, of course, the DBN representation. The other, more common, representation is the *state transition diagram.* To point out the difference and similarities between the two, consider the example of a discrete observation HMM. Assume that the model has $N = 6$ hidden states and $M = 3$ observation states. Specifically, the model is completely specified with

- (hidden) state space $\mathcal{S} = \{e_0, e_1, \ldots, e_5\}$ or equivalently $\mathcal{S} = \{0, 1, 2, 3, 4, 5\}$,
- observation state space $\mathcal{Y} = \{e_0, e_1, e_2\}$ or equivalently $\mathcal{Y} = \{0, 1, 2\}$,
- state transition probability matrix

$$P = \begin{bmatrix} .7 & .2 & & & & .7 \\ & & & & & \\ & .8 & .3 & & & \\ & & .5 & .8 & & \\ & & .1 & & .7 & \\ .3 & & .1 & .2 & .3 & .3 \end{bmatrix},$$

- observation probability matrix

$$P_o = \begin{bmatrix} & .1 & .3 & .1 & .2 & .9 \\ .2 & .2 & .5 & .4 & .2 & \\ .8 & .7 & .2 & .5 & .6 & .1 \end{bmatrix},$$

- and initial state distribution $Pr(s_0 = e_0) = 1$.

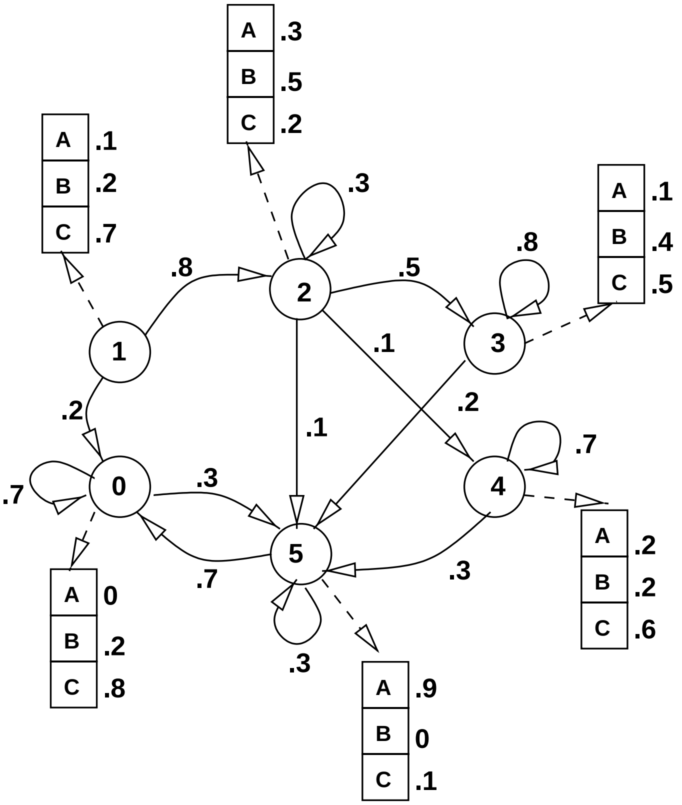


Figure 3.5 State diagram representation of a discrete observation hidden Markov model in text. The model has six hidden states and three observation states with transition and observation probabilities denoted on the diagram. Equivalent DBN representation of the model is still the same as that depicted in Figure 3.2, but must also include the probability tables specified in text.

The classical state-space diagram for the above model is shown in Figure 3.5. It provides information such as what transitions between states are allowed and how likely they are, as well as the probabilities of observation symbols. On the other hand, its DBN dependency graph is that of a "generic" DBN. It is shown again in Figure 3.6 for the case of a length-5 observation sequence. Clearly, the DBN representation is not complete. It only specifies that there are Markovian dependencies between consecutive hidden variables and that observations depend only on current hidden states. However, without specifying the state and observation pdfs, this could just as well be the model of an LDS. On the other hand, the state transition diagram in Figure 3.5 completely specifies the HMM. Nevertheless, it does not clearly show the temporal structure that the model supports.

3.7 Complex Dynamic Bayesian Networks

A complex DBN is an extension of the concept of dynamic Bayesian networks. The network is formed by combining two or more dynamic Bayesian nets into a complex dependency structure.

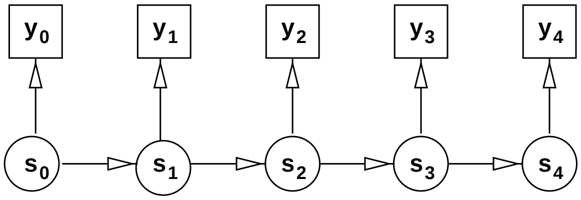


Figure 3.6 Dependency graph for an HMM with an observation sequence of length 5.

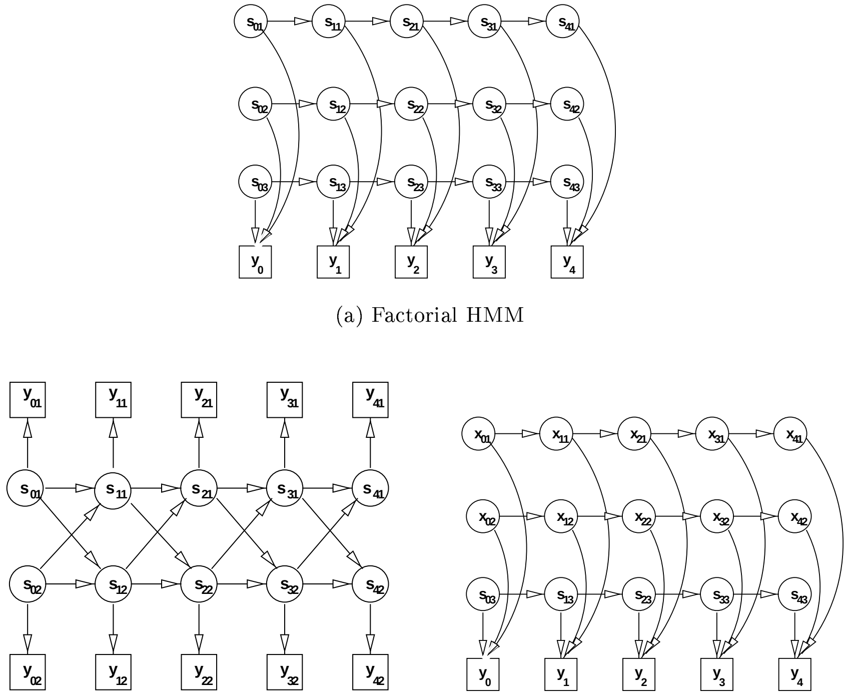
We will consider three types of complex DBNs: factorial HMMs [55], switching state space models [65], and multistate HMMs [75]. Network topologies for the above networks are depicted in Figure 3.7. Factorial HMMs are based on the idea that a sequence of observations can be modeled as driven by a "combination" of underlying discrete Markov processes (see Figure 3.7(a)). Switching space models, on the other hand, assume that a sequence of observations can be obtained by time-multiplexing the outputs of several different linear systems. The multiplexer in this case, as depicted in Figure 3.7(c), is modeled as a discrete Markov process. If a number of different processes concurrently evolves in time and a correlation between processes is assumed to exist, a multistate HMM (see Figure 3.7(b)) can prove to be a reasonable model of such behavior.

Despite their different topologies, all of the above network types can be reduced to basic DBNs using the combination-of-variables method [51]. Consider the case of a complex network obtained by "merging" in some way L different basic network structures. The basis of the method is then to consider all concurrent variables x_{km}, $m = 1, \dots, L$, as a new complex variable $\mathbf{x}_t = (x_{k1}, \dots, x_{kL})$. The new complex variable is defined over its own complex space $\mathcal{X} = \mathcal{X}_1 \times \dots \times \mathcal{X}_L$ of dimension $N_c = N_1 \cdots N_L$. Moreover, the inter- and intra-network conditional pdfs can be easily reformulated to fit this new space. Once this reduction is performed, the inference methods of basic DBNs from Section 3.3 can be directly applied.

However, brute force application of the above method exponentially increases the complexity of inference. To make the inference task tractable, approximate inference techniques, such as the ones described in Section 2.2.3, can be applied. Several particularly appealing complex DBN structures are discussed in great detail in Chapters 4, 5 and 6.

3.8 Hierarchy of Models

In discussing DBNs so far, focus has been on the case of a single network used for modeling of the complete set of observations $\mathcal{Y}_T$. However, in many situations it is more plausible to consider such a sequence being modeled by a set of different DBNs. In other words, different subsequences of $\mathcal{Y}_T$ are modeled using different models from the set. Such situations are often encountered in automatic speech recognition. For example, words are modeled as sequences of smaller units known as phonemes. Each



(a) Factorial HMM

(b) Multistate HMM

(c) Switching state space model

Figure 3.7 Complex dynamic Bayesian networks. The terms s and x denote discrete and continuous space variables, respectively.

phoneme is in turn modeled as an HMM. Besides making modeling more "natural," the use of multiple simpler models usually reduces the complexity of modeling of the whole process. Thus, in modeling spoken English one usually uses combinations of 47 basic phoneme models to represent any particular word in the language (as opposed to having an individual model for every word.)

Let $\mathcal{N} = \{\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_{N_M}\}$ be a set of N_M DBN models. Each model $\mathcal{M}_i,\ i = 1, \ldots, N_M$ with different parameters θ_i is defined over its own state space $\mathcal{X}_i$ and a common observation space $\mathcal{Y}$. To completely specify the set of models one needs to define a relationship between the models in the set (i.e., a relationship between their state spaces). Namely, one needs to define a cross-probability density function

$$P_{i,j} = Pr(x_t^{(i)} | x_{t-1}^{(j)}),\ x_t^{(i)} \in \mathcal{X}_i,\ x_{t-1}^{(j)} \in \mathcal{X}_j. \tag{3.60}$$

between different models. Effectively, this defines a first-order Markov chain over the space of models $\mathcal{N}$, as depicted in the state diagram of Figure 3.8.

Given the above structure of the model set, one can now view it in the following manner. Let $\mathcal{X}$ be the state space formed as a Descartes product of individual model state spaces $\mathcal{X}_i$:

$$\mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_2 \times \cdots \times \mathcal{X}_{N_M}.$$

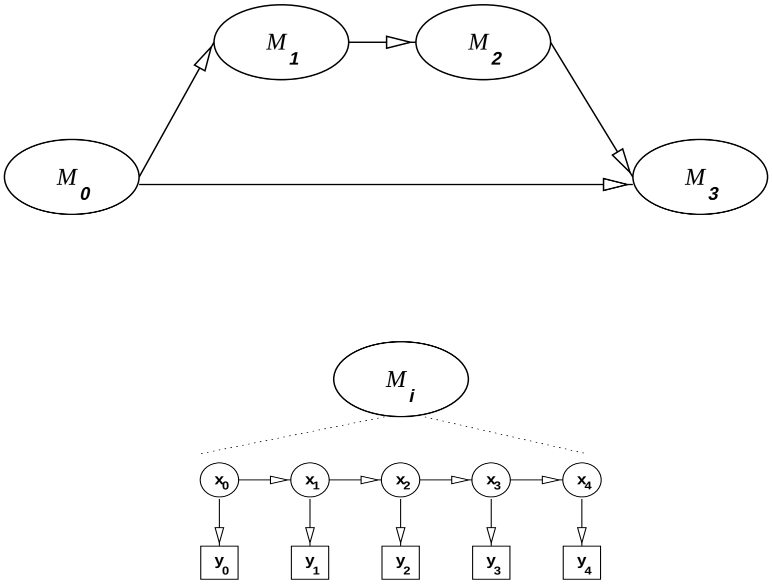


Figure 3.8 State diagram of a Markov chain defined over the space of four models. Markov chain allows transitions from model m_0 to model m_1 or model m_3, model m_1 to model m_2, and model m_2 to model m_3. Given this model-to-model transition topology, only $P_{1,0}$, $P_{3,0}$, $P_{2,1}$, and $P_{3,2}$ are nonzero. Each model is assumed to be a specific DBN, as indicated in the lower figure.

Hence, if the dimension of each model state space $\mathcal{X}_i$ is N_i, the dimension of $\mathcal{X}$ is $\prod_{i=1}^{N_M} N_i$. One can now easily define

$$Pr(x_t|x_{t-1}),\ x_t, x_{t-1} \in \mathcal{X}$$

$$Pr(y_t|x_t),\ y_t \in \mathcal{Y}$$

from Equation 3.60 and the individual model parameters θ_i. Therefore, a global DBN is now defined which models the sequence $\mathcal{Y}_T$ by encompassing a set of models $\mathcal{N}$.

The fact that one can view a structured set of DBNs as a single complex dynamic Bayesian model enables one to readily apply the inference and learning techniques from Sections 3.3 and 3.4 to this case. However, increased state-space and model parameter dimensionality may in practice influence the performance of inference and learning algorithms presented in the above sections. Simplifications can be obtained by noting that very often the structures of global network parameters are very sparse. For instance, it is common that $P_{i,j}(x_t^{(i)}|x_{t-1}^{(j)})$ is nonzero only for a few (i,j). It is sometimes sufficient to assume that the cross-probability density can be factored as $P_{i,j}(x_t^{(i)}|x_{t-1}^{(j)}) = P_i^{in}(x_t^{(i)}) \cdot P_j^{out}(x_{t-1}^{(j)})$. More often, one simply assumes that transitions among models are only possible when the system finds itself in one particular model state known as the *exit state*. Once the system is in an exit state of one model, it can transition to a number of *entry* states of the other models in the set. This corresponds to a very sparse structure of $P_{i,j}$. For discrete state-space models, model-to-model state transition matrices

$P_{i,j}$ become

$$\begin{bmatrix} 0 & \cdots & 0 & x \\ 0 & \cdots & 0 & 0 \\ & \ddots & & \\ & & & 0 \end{bmatrix},$$

where x denotes the only nonzero element of this matrix.

CHAPTER 4

MIXTURES OF DYNAMIC BAYESIAN NETWORKS

4.1 Introduction

Chapter 3 discussed a general as well as two simpler models of DBNs: linear dynamic systems and hidden Markov models. Both of these models inherently carry the following assumption: *all observations y_t at some fixed time instance t are produced by one and only one model.* Thus, even though a pool of N models may be available, it is assumed that only one model is active at a time. Nevertheless, it may often be the case that, at one instance, a number of measurements are available that come from several objects in the observation space. For instance, a camera charge-coupled device (CCD) array provides a set of measurements of a visual scene that may consist of one or more objects (e.g., foreground and background). A question quickly comes to mind: given that one knows how many objects are in the scene at one time, how does one *associate* the available measurements with the objects? In other words, how does one know which measurement came from which object? Moreover, how does one track an object (or multiple objects) in the case of multiple, unassociated measurements? This problem is commonly referred to as the *data association* (DA) and tracking problem. As alluded to before, the DA/tracking problem occurs in many realistic situations. In the case of a visually tracked object, for example, a CCD image captures both the object of interest and its background. It is also feasible that two or more objects of interest are present within the same image in addition to some "noisy" background. Similar situations occur often in ballistic missile tracking, air traffic control, and underwater sonar tracking [76].

The DA/tracking problem has been studied extensively for the past four decades within the framework of single or multiple target tracking in cluttered environments [76]. It was observed early on that the exact Bayesian solution to the DA/tracking problem is in general not tractable. Numerous approximate approaches stemming from generalizations of single object–single observation Kalman filter tracking have been proposed, such as probabilistic data association (PDA) for a single target in a cluttered environment (multiple nontarget related observations) and joint PDA (JPDA) or maximum likelihood PDA (ML/PDA) for multiple targets [77].

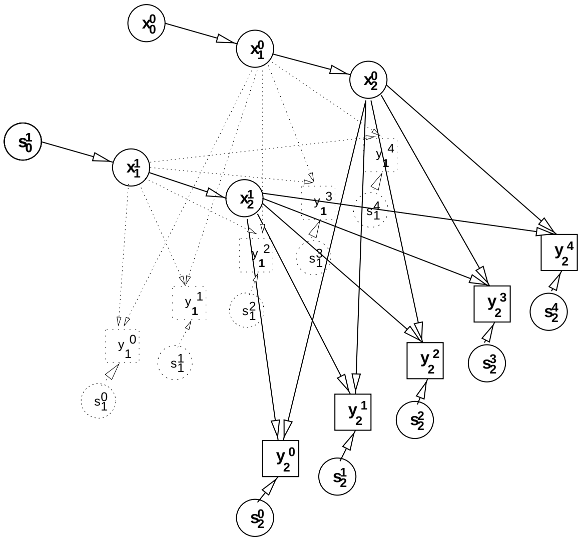


Figure 4.1 A mixture of $N = 2$ dynamic Bayesian networks with observations in the joint observation set of cardinality $M = 5$. At each time instance t an observation $y_t^{(m)}, m = 0, \ldots, 4$ is generated by one of N states $x_t^{(n)}, n = 0, \ldots, 1$. Switching variable $s_t^{(m)}$ determines which of the two dynamic systems generates the observation $y_t^{(m)}$. For convenience, only the observations at $t = 2$ are shown.

On the other hand, there has long been interest in the so-called switching linear dynamic systems [78]. Switching LDSs are those LD models whose parameters instantaneously change (switch) in time according to some (probabilistic) law. For instance, the noise process variance can switch among several different values according to some Markovian dynamics. Similarly, a number of LDSs can be concurrently active while only one of the systems produces all the observations at any given time, again according to some switching dynamics. Recently, Ghahramani and Hinton [65] have proposed a solution to this switching LDS problem from the perspective of approximate inference in DBNs. The resulting switching network topology was mentioned briefly in Section 3.7.

In this section, we extend the idea of a single observation switching DBN to the multiple observation DBNs with unknown observation associations. We denote these network topologies as *mixtures of dynamic Bayesian networks.*

4.2 Model

Consider the case of N concurrent dynamic models, each modeling the temporal evolution of one of N dynamic systems. Let the *joint* observation set for all N models consist of M different observations. Assume furthermore that at each time instance t every observation $y_t^{(m)}$, $m = 0, \ldots, M-1$ is generated by one and only one dynamic system $n, n = 0, \ldots, N-1$. This situation is depicted in Figure 4.1, for $N = 2$ and $M = 5$. To model the dependence of observations on only one dynamic system at every time

t we introduce the switching variable $s_t^{(m)}$ for every observation m. Namely,

$$Pr(y_t^{(m)}|x_t^{(0)},\ldots,x_t^{(N-1)},s_t^{(m)}=n)=Pr(y_t^{(m)}|x_t^{(n)}). \tag{4.1}$$

Here, $s_t(m)$ can take on values from the set of N discrete variables $\{0,\ldots,N\}$ indicating which subnet n the mth observation has come from. Moreover, assume that $s_t^{(m)}$ is a random variable with a known pdf $P_s(n)=Pr(s_t^{(m)}=n)$. Let the dynamics of the systems be modeled using the general DBN framework of Chapter 3, i.e., the dynamic systems can be modeled as either discrete (HMMs), continuous (LDS), or some combination thereof.

The joint pdf defined by this model can now be written as

$$\begin{aligned}P(\mathcal{X},\mathcal{Y},\mathcal{S}) = & \prod_{n=0}^{N-1}\prod_{t=1}^{T-1} Pr(x_t^{(n)}|x_{t-1}^{(n)})Pr(x_0^{(n)}) \\ & \prod_{m=0}^{M-1}\prod_{t=0}^{T-1} Pr(y_t^{(m)}|x_t^{(0)},x_t^{(1)},\ldots,x_t^{(N-1)},s_t^{(m)}) \\ & \prod_{m=0}^{M-1}\prod_{t=0}^{T-1} Pr(s_t^{(m)}),\end{aligned} \tag{4.2}$$

where the sets $\mathcal{X}$, $\mathcal{Y}$, and $\mathcal{S}$ denote the sets of DBN (hidden) state variables, observations, and switching state variables, respectively.

Equivalent to the definition of the joint pdf, one can write the Hamiltonian of this model, using the factorization of Equation 4.1:

$$\begin{aligned}H = & -\sum_{n=0}^{N-1}\left\{\sum_{t=1}^{T-1}\log Pr(x_t^{(n)}|x_{t-1}^{(n)}) - \log Pr(x_0^{(n)})\right\} \\ & -\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}\sum_{t=0}^{T-1} s_t^{(m)}(n)\log Pr(y_t^{(m)}|x_t^{(n)}) \\ & -\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}\sum_{t=0}^{T-1} s_t^{(m)}(n)\log P_s(n).\end{aligned} \tag{4.3}$$

Here, $s_t^{(m)}(n)$ denotes the nth component of the vector index variable $s_t^{(m)}$, similar to the HMM notation of Section 3.6.

Parameters of the mixture of DBNs are clearly the individual parameters of each underlying DBN and the set of observations, as well as the parameters of the switching distribution P_s.

4.3 Inference

Given the complex structure of a mixture of DBNs one is usually interested in recovering the hidden dynamics of the system, given some set of observations: $Pr(\mathcal{X}|\mathcal{Y})$. Moreover, one would like to know the likelihood $Pr(\mathcal{S}|\mathcal{Y})$ of model-to-observation assignments. In the case of a single DBN, as seen

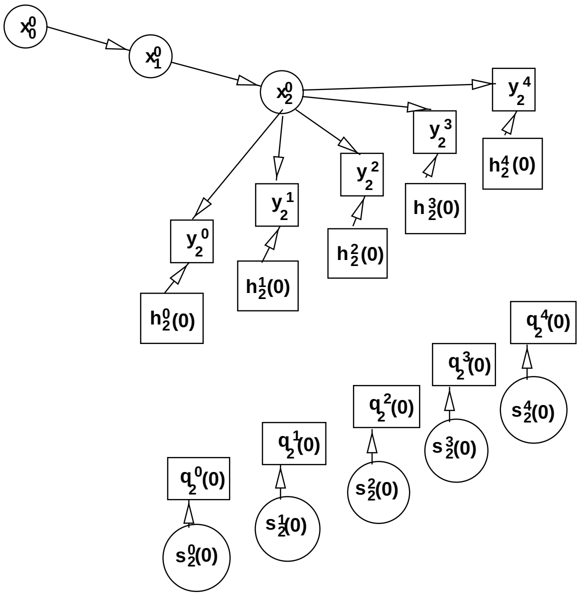


Figure 4.2 Factorization of the mixture of dynamic Bayesian networks from Figure 4.1. For clarity, only the factorization of the mixture subnet $n = 0$ with observations at time $t = 2$ is depicted. Complete network factorization contains N such subnets.

in Chapter 3, efficient inference algorithms exist that provide the desired answer. However, such inference in the case of a mixture of DBNs is clearly intractable. It is therefore necessary to consider some approximate techniques that result in tractable inference.

An appealing approximate inference technique can be readily applied to this problem, namely, the structured variational inference approach described in Section 2.2.3. In this approach one defines an overparameterized and factorized model structure that "resembles" the original intractable topology. In the case of a mixture of DBNs the factorized topology chosen is that of Figure 4.2. Given the proposed factorization, the approximating Hamiltonian becomes

$$\begin{aligned} H_Q = & -\sum_{n=0}^{N-1} \left\{ \sum_{t=1}^{T-1} \log Pr(x_t^{(n)} | x_{t-1}^{(n)}) - \log Pr(x_0^{(n)}) \right\} \\ & - \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \sum_{t=0}^{T-1} h_t^{(m)}(n) \log Pr(y_t^{(m)} | x_t^{(n)}) \\ & - \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \sum_{t=0}^{T-1} \left(s_t^{(m)}(n) + q_t^{(m)}(n) \right) \log P_s^{(n)}. \end{aligned} \tag{4.4}$$

Parameters $h_t^{(m)}(n)$ and $q_t^{(m)}(n)$ are the variational parameters of the approximating distribution Q that are to be optimized in the sense of Section 2.2.3. These parameters are denoted as the observation and switching state weights, respectively.

Using Theorem 1 it is easy to show (see Appendix B) that the following set of fixed-point equations defines the optimal approximating distribution

$$\begin{aligned} h_t^{(m)}(n) &= \left\langle s_t^{(m)}(n) \right\rangle & (4.5) \\ q_t^{(m)}(n) &= \exp \left\langle \log \left(Pr(y_t^{(m)} | x_t^{(n)}) \right) \right\rangle . & (4.6) \end{aligned}$$

From the above equations it is clear that the weight $h_t^{(m)}(n)$ of observation m in system n is proportional to the probability that the observation came from this system, given all measurements. The switching state weight $q_t^{(m)}(n)$ measures the "likelihood" of the mth observation given the estimates of the nth DBN's states. It is now easy to show that the switching state weights $q_t^{(m)}(n)$ in Equation 4.6 together with the factorization Q yield

$$\left\langle s_t^{(m)}(n) \right\rangle = \frac{q_t^{(m)}(n) P_s(n)}{\sum_{l=0}^{N-1} q_t^{(m)}(l) P_s(l)}. \tag{4.7}$$

Hence, $\left\langle s_t^{(m)}(n) \right\rangle$ is a MAP-like Bayesian estimate of the probability that observation m came from system n. The final cost function of this approximation (see Section 2.2.3) can then be written as

$$\text{Cost} = B(P, Q) = \langle H - H_Q \rangle - \log Z_Q, \tag{4.8}$$

where Z_Q denotes the "observation" probability in the factorized network and $\langle H - H_Q \rangle$ depends on the choice of DBN dynamics.

In summary, the inference algorithm can be now written as follows:

```
error = $\infty$;
Initialize sufficient statistics of $Pr(y_t^{(m)}|x_t^{(n)})$ ($\langle x \rangle$, etc.) ;
while (error > maxError) {
    Find $\langle s_\cdot \rangle$ using Equation 4.7;
    Estimate sufficient statistics of $Pr(y_t^{(m)}|x_t^{(n)})$ from $\langle s_\cdot \rangle$ and $y_c dot$ in the factorized network $Q$;
    Update Cost using Equation 4.8;
    error $\leftarrow$ ( oldCost - Cost ) / Cost;
}
```

We now consider the variational inference approximation for two special cases of mixed DBNs where the underlying mixture networks are LDSs and HMMs.

4.3.1 Mixture of linear dynamic systems

In the case of a set of N linear dynamic systems, the observation pdf of the mth measurement given the nth system is assumed to be Gaussian with mean $C^{(n)}$ and variance $R^{(n)}$:

$$Pr(y_t^{(m)} | x_t^{(n)}) = \mathcal{N}(C^{(n)}, R^{(n)}),$$

where $y_t^{(m)} \in \Re^{N_y}$ is assumed for all m.

Therefore, the switching weight fixed-point equation, Equation 4.6, becomes

$$\begin{aligned} q_t^{(m)}(n) &= (2\pi)^{-\frac{1}{2}N_y}|R^{(n)}|^{-\frac{1}{2}} \exp\left\{ -\frac{1}{2} \left\langle \left(y_t^{(m)} - C^{(n)} x_t^{(n)} \right)' {R^{(n)}}^{-1} \left(y_t^{(m)} - C^{(n)} x_t^{(n)} \right) \right\rangle \right\} \\ &= (2\pi)^{-\frac{1}{2}N_y}|R^{(n)}|^{-\frac{1}{2}} \exp\left\{ -\frac{1}{2} \left(y_t^{(m)} - C^{(n)} \left\langle x_t^{(n)} \right\rangle \right)' {R^{(n)}}^{-1} \left(y_t^{(m)} - C^{(n)} \left\langle x_t^{(n)} \right\rangle \right) \right\} \\ &\quad \cdot \exp\left\{ -\frac{1}{2} \text{trace} \left({R^{(n)}}^{-1} \hat{R}_t^{(n)} \right) \right\}, \end{aligned}$$

where $\hat{R}_t^{(n)}$ denotes the variance of state $x_t^{(n)}$:

$$\hat{R}_t^{(n)} = \left\langle \left(x_t^{(n)} - \left\langle x_t^{(n)} \right\rangle \right) \left(x_t^{(n)} - \left\langle x_t^{(n)} \right\rangle \right)' \right\rangle .$$

Clearly, the closer the observation $y_t^{(m)}$ is to the one infered by the state of the nth LDS, the more it weighs the switching state. The estimate of the switching state $\left\langle s_t^{(m)}(n) \right\rangle$ in Equation 4.7 therefore resembles an estimate of a mixture of Gaussian distribution mixing weights.

To obtain the estimates of the hidden system states $x_t^{(n)}$ one needs to use a time-varying version of the LDS inference described in Section 3.5 on each LDS subnet. However, there are two subtle differences when applying the LDS inference to each mixed DBN subnet. First of all, one needs to weigh the observation variance $R^{(n)}$ at each time instance t by the factor $1/h_t^{(m)}(n)$, depending on which one of M observations is being fused. This stems from the fact that according to the chosen factorization in Equation 4.4 the inverse of each observation variance $R^{(n)}$ gets multiplied by $h_t^{(m)}(n)$.[1] Second, in the observation fusing phase of LDS inference, there are M different observations instead of a usual single observation. However, this does not pose any difficulty because the M-observation inference task can be easily reduced to a one-observation inference by concatenating the M observations into an $M \times N_y$ vector with a block diagonal observation variance consisting of weighed matrices $R^{(n)}/h_t^{(m)}(n)$.[2]

4.3.2 Mixture of hidden Markov models

In the mixture-of-HMMs network one assumes that each of N dynamic subnets can be modeled as an HMM of Section 3.6. Consider the case of discrete observation HMMs.[3] Here, the observation pdf of the mth measurement coming from the nth HMM subnet is given by the probability table $P_o^{(m)}(\cdot,\cdot)$:

$$Pr(y_t^{(m)} = j | x_t^{(n)} = i) = P_o^{(m)}(j,i),$$

[1] Recall from Equation 4.3 that the observation log likelihood $\log Pr(y_t^{(m)}|x_t^{(n)})$ is scaled by variational parameter $h_t^{(m)}(n)$. Since the pdf in question is Gaussian, scaling the log likelihood by $h_t^{(m)}(n)$ yields the same inference results as scaling of the distribution variance by the inverse of $h_t^{(m)}(n)$.

[2] Obviously, the observations can also be recursively fused using the recursive least-squares procedure, or, equivalently, a Kalman filter in the observation space with a unit transition matrix and zero state noise.

[3] The case of continuous, Gaussian observation HMM is analogous to the case of Gaussian observation noise LDS of the previous section.

where i and j take on values from the sets of indices of hidden states $\{0, \ldots, N_x - 1\}$ and observations $\{0, \ldots, N_y - 1\}$, respectively.

The switching weight Equation 4.6 now assumes the following form:

$$q_t^{(m)}(n) = \prod_{i=0}^{N_x-1} P_o(m,i)^{\left\langle x_t^{(n)}(i) \right\rangle}. \tag{4.9}$$

As before, the estimates of the switching states are defined by Equation 4.7. To obtain the estimates of the hidden dynamical system states $x_t^{(n)}$ one uses a modified time-varying version of the standard HMM inference of Section 3.6. Namely, as in the case of the LDS inference above, two modifications are necessary. To account for M different measurements, one simply considers a single measurement with the probability equal to the product of M individual observation probabilities. Furthermore, each one of M observation probabilities is weighed by the power $h_t^{(m)}$, according to the factorization Q in Equation 4.4.

4.4 Learning

Given the pdf factorization of Equation 4.4, learning the parameters of a mixture of DBNs becomes trivial and reduces to the learning of parameters of individual LDS or HMM subnets. This follows directly from the discussion of the generalized EM algorithm in Section 2.3. For instance, maximization of a mixture of LDS parameters yields the set of update equations, similar to the ones of Section 3.5.2:

$$\begin{aligned}
A_{new}^{(n)} &= \left(\sum_{t=1}^{T-1} \left\langle x_t^{(n)} x_{t-1}^{(n)}{}' \right\rangle \right) \left(\sum_{k=1}^{T-1} \left\langle x_t^{(n)} x_t^{(n)\prime} \right\rangle \right)^{-1} && (4.10) \\
Q_{new}^{(n)} &= \frac{1}{T-1} \left(\sum_{t=1}^{T-1} \left\langle x_t^{(n)} x_{t-1}^{(n)}{}' \right\rangle - A_{new} \sum_{t=1}^{T-1} \left\langle x_{t-1}^{(n)} x_t^{(n)\prime} \right\rangle \right) && (4.11) \\
C_{new}^{(n)} &= \left(\sum_{m=0}^{M-1} \sum_{t=0}^{T-1} y_t^{(m)} \left\langle x_t^{(n)\prime} \right\rangle \left\langle s_t^{(m)}(n) \right\rangle \right) \left(\sum_{t=0}^{T-1} \left\langle x_t^{(n)} x_t^{(n)\prime} \right\rangle \right)^{-1} && (4.12) \\
R_{new}^{(n)} &= \frac{1}{T} \sum_{m=0}^{M-1} \sum_{t=0}^{T-1} \left(y_t^{(m)} y_t^{(m)\prime} \left\langle s_t^{(m)}(n) \right\rangle - C_{new} \left\langle x_t^{(n)} \right\rangle y_t^{(m)\prime} \left\langle s_t^{(m)}(n) \right\rangle \right) && (4.13) \\
\mu_{new}^{(n)} &= \sum_{m=0}^{M-1} \left\langle x_0^{(n)} \right\rangle \left\langle s_t^{(m)}(n) \right\rangle && (4.14) \\
Q_{0,new}^{(n)} &= \sum_{m=0}^{M-1} \left(\langle x_0 x_0{}' \rangle - \langle x_0 \rangle \langle x_0 \rangle{}' \right) \left\langle s_t^{(m)}(n) \right\rangle . && (4.15)
\end{aligned}$$

The equations are obviously a generalization of the single LDS parameter update equations from Section 3.5.2. Similar parameter update equations can be obtained for a mixture of HMMs.

In addition to the subnet parameters, the probability of associations P_s can be easily reestimated in the same GEM framework. Namely, the update equation for the data association can be easily shown to be

$$P_{s,new}(n) = c \sum_{t=0}^{T-1} \sum_{m=0}^{M-1} \left\langle s_t^{(m)}(n) \right\rangle, \tag{4.16}$$

where c is the normalization constant.

CHAPTER 5

MIXED-STATE DYNAMIC BAYESIAN NETWORKS

5.1 Introduction

In this chapter we consider a specific instance of dynamic Bayesian networks (DBNs) that arise from those of the previous chapter: hidden Markov models and Kalman filters (KFs). We call such DBNs *mixed-state DBNs*. A mixed-state DBN is a coupled combination of an HMM and a KF. In it, the output of an HMM is the *driving* input to a linear system. A block diagram of this system is depicted in Figure 5.1.

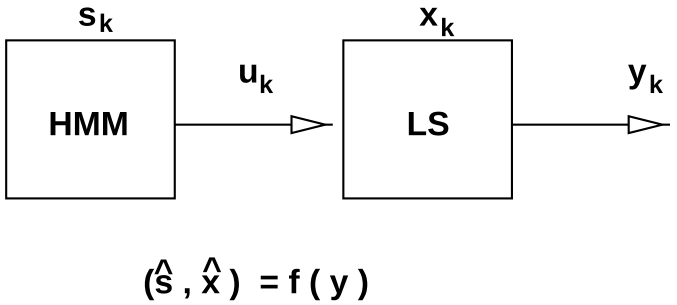


Figure 5.1 Block diagram of a physical system (LDS) driven by an input generated from a hidden Markov model (HMM). The system has an equivalent representation as a mixed-state dynamic Bayesian network.

What motivates one to consider such a combination of systems? Several examples may easily come to mind. Suppose that one observes an autonomous moving target. The target motion is governed by Newtonian physics as well as the input force (thrust) imposed upon it by its human operator. Assume that one has some knowledge of what sequence of actions the operator may take in time. In other words, one knows that there are dependencies between levels of control thrust at successive time instances. Thus, it is plausible to model the thrust controlled by an operator as an HMM where the hidden states correspond to a number of possible actions the operator may take, and the observables model the thrust value at those action instances. Knowledge of object motion under Newtonian laws of physics is embedded in the linear system model.

Another example of a physical system that can be modeled as a mixed DBN is the human hand/arm motion during gestural communication. Like spoken language, hand gestures are claimed to posses an inherent linguistic structure of the concepts that drive them (e.g., see [79]). Consequently (and again similar to spoken language modeling), one may want to model such concepts using HMMs. However, instead of driving the human vocal tract, the concepts now control the motion of the human arm through, for example, arm joint torques. This physical arm motion can, in turn, be described using different kinematic and dynamic motion models of simple or articulated structures. Again, these physical models are built into the block denoted as the linear/nonlinear system.

In both of the above examples the aim of modeling the systems in the mixed DBN framework is to be able to infer what the underlying concept that drives the physical system is. This can help to distinguish among different motion patterns of a target observed by a radar or among different hand gestures observed by a computerized interactive kiosk. Moreover, the concepts need not only be inferred—they can also be predicted. Consequently, the motion of the target or the human hand can be predicted too (based on the predicted concept) and used for tracking of the physical systems.

In the rest of this chapter techniques are formulated to achieve the goal of estimating and predicting the system concept and the states of the physical system. Furthermore, we show how this can be used to learn the model parameters of both the concept generator and the physical system.

5.2 Model

Consider a coupled system whose block diagram is depicted in Figure 5.1. The system can in general be described using the following set of state-space equations:

$$\begin{aligned} x_{t+1} &= A_{t+1}x_t + B_{t+1}u_{t+1} + v_{t+1}, \\ y_t &= C_t x_t + w_t, \text{ and} \\ x_0 &= v_0 \end{aligned}$$

for the physical system, and

$$\begin{aligned} Pr(s_{t+1}, s_t) &= Pr(s_{t+1}|s_t)Pr(s_t), \\ Pr(u_t, s_t) &= Pr(u_t|s_t)Pr(s_t), \text{ and} \\ Pr(s_0) &= \pi_0 \end{aligned}$$

for the generating (driving) concept. The meaning of the variables is as usual (see Section 3.5): x is used to denote the (hidden) state of the LDS system, u is an input to this system, while v represents the state

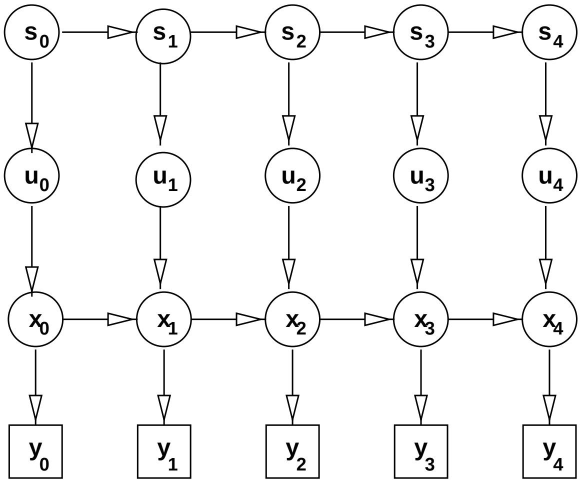


Figure 5.2 Bayesian network representation (dependency graph) of the mixed-state DBN. Term s denotes instances of the discrete valued concept states driving the continuous valued physical system states x and observations y.

noise process. Similarly, y is the measurement (observation) and w is the measurement noise. Parameters A, B, and C are the typical LDS parameters: the state transition matrix, the input "gain" matrix, and the observation matrix, respectively. We represent the concept generator as a hidden Markov model. States of this model are written as s. The model itself is defined using an appropriate state transition pdf $Pr(s_{t+1}|s_t)$, observation pdf $Pr(u_t|s_t)$, and initial state distribution π_0. Again, the notation is analogous to that used in Section 3.6. Note that the input u to the LDS is the output of the concept HMM.

Finally, assume that the dimensions of the variable spaces are

- $s. \in \{e_0, \ldots, e_{S-1}\}$,[1]
- $x. \in \Re^N$, and
- $y. \in \Re^M$.

The state-space representation is equivalently represented in the dependency graph in Figure 5.2 and can be written as

$$Pr(\mathcal{Y}, \mathcal{X}, \mathcal{U}, \mathcal{S}) = \prod_{t=1}^{T-1} Pr(s_t|s_{t-1}) Pr(s_0)$$
$$\prod_{t=0}^{T-1} Pr(u_t|s_t)$$
$$\prod_{t=1}^{T-1} Pr(x_t|x_{t-1}, u_t) Pr(x_0|u_0)$$

[1] e_i is the unit vector of dimension S with a non-zero element in the i-th position. For more detail on this notation see Section 3.6

$$\prod_{t=0}^{T-1} Pr(y_t|x_t), \tag{5.1}$$

where $\mathcal{Y}, \mathcal{X}, \mathcal{U}$, and $\mathcal{S}$ denote the sequences of length T of observations and hidden state variables. Of course, as we will soon show, variable $u_\cdot$ can be absorbed into $x_\cdot$, thus yielding a simplified pdf form:

$$Pr(\mathcal{Y}, \mathcal{X}, \mathcal{S}) = \prod_{t=1}^{T-1} Pr(s_t|s_{t-1}) Pr(s_0)$$
$$\prod_{t=1}^{T-1} Pr(x_t|x_{t-1}, s_t) Pr(x_0|s_0)$$
$$\prod_{t=0}^{T-1} Pr(y_t|x_t).$$

Terms v_t and w_t in the physical system formulation are used to denote random noise terms. One can write an equivalent representation of the physical system in the probability space assuming that the following conditional pdfs are defined:

$$\begin{aligned} Pr(x_{t+1}|x_t, u_{t+1}) &= P_x(x_{t+1} - A_{t+1}x_t - B_{t+1}u_{t+1}), &(5.2)\\ Pr(y_t|x_t) &= P_y(y_t - C_t x_t), &(5.3)\end{aligned}$$

where P_x and P_y are some known, usually parameterized, pdfs.

Throughout the rest of this chapter we assume without loss of generality that the state noise v of the physical system is zero with probability one (w.p.1) or nonexistent. This is allowed given the fact that noise in the states of the physical system can be accounted for by the noise in the output of the driving HMM. In addition, assume that the observation noise of the physical system is modeled as an i.i.d. zero-mean Gaussian process:

$$w_t \sim \mathcal{N}(0, R).$$

Furthermore, we restrict the output distribution of the HMM subsystem to be Gaussian with constant variance Q and the mean determined by the process's hidden states:

$$\begin{aligned} u_t|s_t &\sim \mathcal{N}(Ds_t, Q), \quad t > 0\\ u_0|s_0 &\sim \mathcal{N}(D_0 s_0, Q_0).\end{aligned}$$

Finally, assume that the physical system parameters A, B, C and the concept system state transition pdf P are time-invariant, and without loss of generality reduce B to identity $B = I$. The model parameters can then be summarized as follows:

- continuous subsystem parameters A, C, and R,
- discrete subsystem parameters P and π_0, and

- coupling parameters D and Q.

The model state equations now yield

$$\begin{aligned} Pr(s_{t+1}|s_t) &= s_{t+1}' P s_t, &(5.4)\\ u_t &= D s_t + v_t, &(5.5)\\ x_{t+1} &= A x_t + u_{t+1}, &(5.6)\\ y_t &= C x_t + w_t, &(5.7)\end{aligned}$$

with initial conditions

$$\begin{aligned} Pr(s_0) &= \pi_0,\\ x_0 &= D_0 s_0 + v_0.\end{aligned}$$

Given the above assumptions, the joint pdf of the mixed-state DBN of duration T (or, equivalently, it's Hamiltonian) can be written as

$$\begin{aligned} H = &\frac{1}{2}\sum_{t=1}^{T-1}(x_t - Ax_{t-1} - Ds_t)' Q^{-1} (x_t - Ax_{t-1} - Ds_t) + \frac{T-1}{2}\log|Q|\\ &+\frac{1}{2}(x_0 - D_0 s_0)' {Q_0}^{-1} (x_0 - D_0 s_0) + \frac{1}{2}\log|Q_0| + \frac{NT}{2}\log 2\pi\\ &+\frac{1}{2}\sum_{t=0}^{T-1}(y_t - Cx_t)' R^{-1} (y_t - Cx_t) + \frac{T}{2}\log|R| + \frac{MT}{2}\log 2\pi\\ &+\sum_{t=1}^{T-1} s_t'(-\log P)s_{t-1} + s_0'(-\log \pi_0). \qquad (5.8)\end{aligned}$$

Note that in this formulation we distinguish the initial parameter values D_0, Q_0 from the same parameters in the rest of the temporal sequence D, Q.

5.3 Inference

The goal of inference in mixed-state DBNs is to estimate the likelihood of hidden states of the system (s and x) for some known sequence of observations $y_0, \ldots, y_{T-1}$ and the known model parameters. Namely, one needs to find

$$Pr(s_0, \ldots, s_{T-1}, x_0, \ldots, x_{T-1} | y_0, \ldots, y_{T-1}).$$

Let $\mathcal{S}, \mathcal{X}$, and $\mathcal{Y}$ denote the sequences (sets) of variables of interest, where each set contains a sequence of T consecutive temporal variables. The inference problem requires one to determine

$$Pr(\mathcal{S}, \mathcal{X} | \mathcal{Y}).$$

Alternatively, one can find the sufficient statistics of the above distribution.

In this formulation we have omitted a set of intermediate hidden variables $\mathcal{U} = \{u_0, \dots, u_{T-1}\}$, which is the set of inputs to the physical system. However, these variables are crucial for the inference. They can be reintroduced using the following marginalization:

$$\begin{aligned} Pr(\mathcal{S}, \mathcal{X}|\mathcal{Y}) &= \int_{\mathcal{U}} Pr(\mathcal{S}, \mathcal{U}, \mathcal{X}|\mathcal{Y}) \\ &= \int_{\mathcal{U}} Pr(\mathcal{X}|\mathcal{U}, \mathcal{Y}) Pr(\mathcal{S}|\mathcal{U}) Pr(\mathcal{U}|\mathcal{Y}). \end{aligned} \tag{5.9}$$

Now, the theory of HMMs and Kalman filters from Chapter 3 directly provides the intermediate answers to the constituting conditionals. Using the KF framework one can easily find $Pr(\mathcal{X}|\mathcal{U}, \mathcal{Y})$. Similarly, it is easy to determine $Pr(\mathcal{S}|\mathcal{U})$. The problem lies in the marginalization of Equation 5.9. To see that, assume that there are S possible discrete hidden states of the HMM subsystem. Let the initial distribution of x_0 be Gaussian with some mean and variance. At $t = 1$ the pdf of the physical system state x_1 becomes a mixture of S Gaussian pdfs since one needs to marginalize over S possible but unknown input levels. It is clear that this procedure yields a pdf exponential in the number of mixture components. Clearly, the inference task is of exponential complexity.

Next we consider two types of approaches that deal with the complexity of inference. One is a completely decoupled inference while the other maintains coupling between the two subsystems using an iterative procedure.

5.3.1 Decoupled inference

Decoupled inference is the more common of the two approaches. It is often (wrongfully) considered the only approach to inference in the mixed-state systems.

Consider the inference equation as formulated in 5.9. Assume, furthermore, that the conditional pdf $Pr(\mathcal{U}|\mathcal{Y})$ in Equation 5.9 has a dominant peak at the maximum likelihood (ML) estimate of u, $u = u_{ML}$. One can then approximate the inference equation with

$$Pr(\mathcal{S}, \mathcal{X}|\mathcal{Y}) \propto Pr(\mathcal{X}|\mathcal{U}_{ML}, \mathcal{Y}) Pr(\mathcal{S}|\mathcal{U}_{ML}). \tag{5.10}$$

Clearly, once the ML estimate of u is known, $\mathcal{S}$ and $\mathcal{X}$ become independent. Inference of $\mathcal{S}$ and $\mathcal{X}$ can then be independently formulated in the decoupled subnets. This is figuratively depicted in Figure 5.3. The question now becomes how to obtain the ML estimate of the input u. The theory of LDS presented in Section 3.5 answers the question of how to find the ML estimate of the system states x when the noise processes are i.i.d. Gaussian. A common trick can be used to find the desired ML estimates of u in the same manner. Consider the LDS state space equations defined in 5.1. These equations can be rewritten

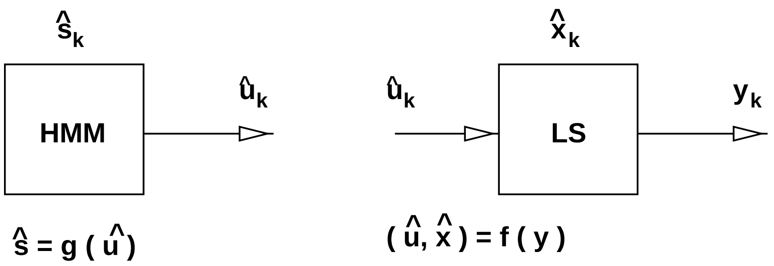


Figure 5.3 Inference in decoupled mixed-state DBN can be easily accomplished once the coupling $u.$ is known.

in the following form:

$$\begin{aligned} \begin{bmatrix} x_{t+1} \\ u_{t+2} \end{bmatrix} &= \begin{bmatrix} A_{t+1} & B_{t+1} \\ 0 & I \end{bmatrix} \begin{bmatrix} x_t \\ u_{t+1} \end{bmatrix} + \begin{bmatrix} v_{t+1} \\ v^u_{t+1} \end{bmatrix}, \\ y_t &= C_t x_t + w_t, \\ x_0 &= v_0. \end{aligned}$$

Here, we introduce an additional constraint on input u: it must be almost constant between two consecutive time steps. Of course, the level of constancy can be determined by the variance of the noise process v^u. Techniques have been developed that adaptively adjust the level of noise so that the state estimates satisfy a fixed significance level [76]. These techniques are a modification of the basic Kalman filter principle formulated in Section 3.5. Hence, an ML estimate of u (together with an ML estimate of x) can be easily obtained. Given this estimate, the inference of s is trivially achieved in the HMM framework.

The problem with the presented approach, however, is in its basic assumption that there is a dominant peak around the ML estimate of u. Even though this assumption may hold in majority of cases, instances can be constructed where the assumption fails. To deal with that problem we next formulate an alternative *coupled* approach to inference in mixed-state DBNs based on variational inference approximation.

5.3.2 Variational inference

Application of variational inference techniques from Section 2.2.3 to mixed-state DBN is particularly appealing. To see this, consider the factorization of the original network into an approximate but decoupled one, depicted in Figure 5.4. The two subnetworks of the original network are an over-parameterized HMM with variational parameters q and an LDS with variational parameters u.

More precisely, we define the Hamiltonian of the approximating network as

$$H_Q = \frac{1}{2} \sum_{t=1}^{T-1} (x_t - Ax_{t-1} - u_t)' \, Q^{-1} \, (x_t - Ax_{t-1} - u_t) + \frac{T-1}{2} \log |Q|$$

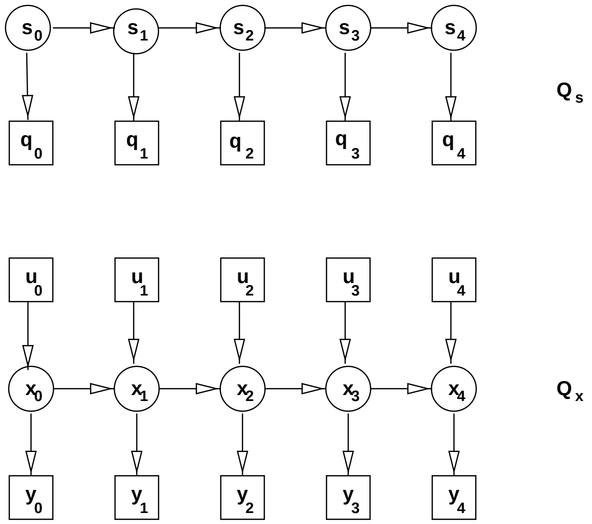


Figure 5.4 Factorization of the original mixed-state DBN. Factorization reduces the coupled network into a decoupled pair of an HMM (Q_s) and an LDS (Q_x).

$$
\begin{aligned}
&+\frac{1}{2}\left(x_0-u_0\right)' {Q_0}^{-1}\left(x_0-u_0\right)+\frac{1}{2}\log\left|Q_0\right|+\frac{NT}{2}\log 2\pi \\
&+\frac{1}{2}\sum_{t=0}^{T-1}\left(y_t-Cx_t\right)' R^{-1}\left(y_t-Cx_t\right)+\frac{T}{2}\log\left|R\right|+\frac{MT}{2}\log 2\pi \\
&+\sum_{t=1}^{T-1} s_t'(-\log P)s_{t-1}+s_0'(-\log \pi_0)+\sum_{t=0}^{T-1} s_t'(-\log q_t).
\end{aligned}
\tag{5.11}
$$

The meaning of the variational parameters will become clear once we identify their update (fixed-point) equations. The fixed-point equations are easily obtained by applying Theorem 1 with the factorization of Equation 5.11. Skipping over the intermediate derivation steps (which can be found in Appendix C) the theorem yields

- Variational parameters of the LDS subnet Q_x:

$$
u_\tau^* = D\left\langle s_\tau\right\rangle, \quad \forall\tau \tag{5.12}
$$

- Variational parameters of the HMM subnet Q_s:

$$
q_\tau^*(i) = \begin{cases} \exp\left\{d_i' Q^{-1}\left(\left\langle x_\tau\right\rangle - A\left\langle x_{\tau-1}\right\rangle - \frac{1}{2}d_i\right)\right\} & \tau > 0 \\ \exp\left\{{d_0}_i' Q_0^{-1}\left(\left\langle x_0\right\rangle - \frac{1}{2}d_0\right)\right\} & \tau = 0, \end{cases} \tag{5.13}
$$

where d_i denotes the ith column of D. To obtain the expectation terms $\langle s_\tau\rangle$ and $\langle x_\tau\rangle$ in the above equations one needs to use the classical inference in the HMMs with output probabilities q_τ and LDSs with inputs u_τ, respectively. This follows directly from the chosen network factorization and is figuratively depicted in Figure 5.4.

The error bound of the above approximation can be shown to be (see Appendix C)

$$
\begin{aligned}
&\mathrm{Cost} = B(P, Q) = \\
&\quad \sum_{t=1}^{T-1} \left(\langle x_t \rangle - A \langle x_{t-1} \rangle \right)' Q^{-1} \left(u_t^* - D \langle s_t \rangle \right) \\
&\quad + \frac{1}{2} \sum_{t=1}^{T-1} \mathrm{tr} \left\{ D' Q^{-1} D \langle s_t s_t' \rangle \right\} - \frac{1}{2} \sum_{t=1}^{T-1} \left(u_t^* \right)' Q^{-1} \left(u_t^* \right) \\
&\quad + \langle x_0 \rangle Q_0^{-1} \left(u_0^* - D \langle s_0 \rangle \right) + \frac{1}{2} \mathrm{tr} \left\{ D_0' Q_0^{-1} D_0 \langle s_0 s_0' \rangle \right\} - \frac{1}{2} \left(u_0^* \right)' Q_0{}^{-1} \left(u_0^* \right) \\
&\quad + \sum_{t=0}^{T-1} \langle s_t \rangle' \log q_t^* \\
&\quad - \log P_{Q_s} - \log P_{Q_x},
\end{aligned}
\tag{5.14}
$$

where $\log P_{Q_s}$ and $\log P_{Q_x}$ roughly correspond to the log likelihoods of quasi-observations and observations in the HMM and LDS subnets of Q, respectively. Namely, following the definitions in Section 2.2.3,

$$P_{Q_s} = \sum_{\mathcal{S}} Q_s(\mathcal{S}),$$

and

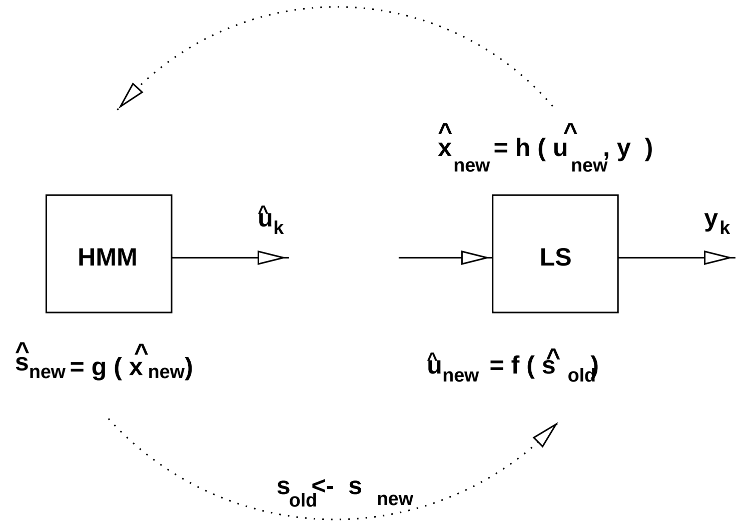
$$P_{Q_x} = \int_{\mathcal{X}} Q_x(\mathcal{X}, \mathcal{Y}).$$

The variational inference algorithm for mixed-state DBNs can now be summarized as

```
error = $\infty$;
Initialize $\langle x \rangle$;
while (error > maxError) {
    Find $q.$ from $\langle x. \rangle$ using Equation 5.13;
    Estimate $\langle s. \rangle$ from $q.$ using HMM inference;
    Find $u.$ from $\langle s. \rangle$ using Equation 5.12;
    Estimate $\langle x. \rangle$ from $y.$ and $u.$ using Kalman inference;
    Update Cost using Equation 5.14;
    error $\leftarrow$ ( oldCost - Cost ) / Cost;
}
```

The algorithm is symbolically depicted in Figure 5.5. All joint statistics of the original network of the form $\langle x_t s_t' \rangle$ can now be found decoupled as $\langle x_t \rangle \langle s_t \rangle'$.

The meaning of the variational parameters can now be examined. It is obvious from Equation 5.12 and the factorization of the network defined in Equation 5.11 that $u.$ parameters can be viewed as the estimated inputs of the LDS, based on the estimates of the hidden states of the HMM subnet. The input at time τ is estimated to be a linear combination of all possible inputs d_i weighted by their



$\hat{x}_{new} = h(\hat{u}_{new}, y)$
HMM
$\hat{u}_k$
LS
y_k
$\hat{s}_{new} = g(\hat{x}_{new})$
$\hat{u}_{new} = f(\hat{s}_{old})$
$s_{old} \leftarrow s_{new}$

Figure 5.5 Symbolical depiction of a variational inference loop in a mixed-state DBN. Assume that one is given an estimate of s. Estimate of LDS input u is obtained from s, and estimate of LDS state x is found from u and y. Finally, an updated estimate of s can be found from x.

corresponding likelihoods $\langle s_\tau(i) \rangle$, $D \langle s_\tau \rangle = \sum_{i=0}^{N-1} d_i \langle s_\tau(i) \rangle$. The meaning of $q_\cdot$ is not immediately obvious. Based on Equation 5.11, $q_\cdot$ can be viewed as the probabilities of some fictional discrete-valued inputs presented to the HMM subnet. These probabilities are related to the estimates of the states x_τ of the LDS through Equation 5.13. To better understand the meaning of this dependency consider the plot in Figure 5.6 of q_τ versus $d = d_i$ for a fixed value of the difference $\langle x_\tau \rangle - A \langle x_{\tau-1} \rangle$ and unit variance Q. We restrict ourselves to the scalar case and assume that d can take on any value in $\Re$. Clearly, the function assumes a maximum value for $d = \langle x_\tau \rangle - A \langle x_{\tau-1} \rangle$. If we had a set of discrete values of d corresponding to N possible LDS input levels d_i, $q_\tau(i)$ would be maximized for d_i closest to the estimated difference $\langle x_\tau \rangle - A \langle x_{\tau-1} \rangle = \langle u_\tau \rangle$. Thus, those states of the HMM are favored which produce inputs "closer" to the ones estimated from the LDS dynamics.

5.4 Decoding

The inference task presented in the previous section demands that estimates of hidden states of the HMM and LDS subnets be found. These estimates are conditioned on all available observations, both in past and in future. However, we can restrict ourselves to the estimates of the hidden variables based only on the observations up to the time instance of the variable of interest. Then, we deal with the forward propagation in DBNs, as defined in Chapter 3. One alternative of the forward propagation is Viterbi decoding. Here, a best sequence of hidden variables is found that maximizes the probability of

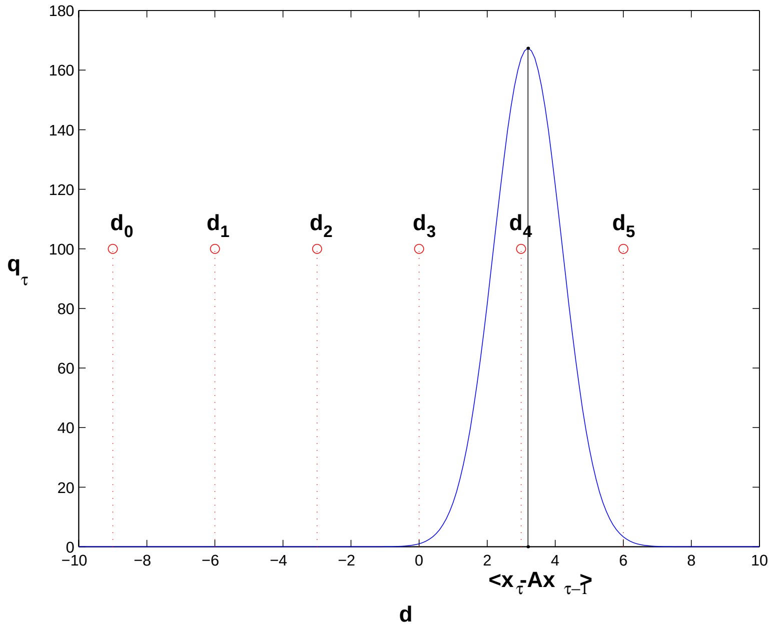


Figure 5.6 Variational parameter q_τ as a function of input level d. Shown is a set of six different input levels d_0 through d_5. The function attains maximum for input level d_4, which is closest to the global maximum at $d^* = \langle x_\tau \rangle - A \langle x_{\tau-1} \rangle$.

observations. Viterbi decoding in both HMMs and LDSs was tackled in Chapter 3. We now formulate a Viterbi decoding procedure for mixed-state DBNs.

Consider a pair (x_t, s_t). Given a sequence of observations it is plausible that only one sequence of pairs (x_t, s_t), $0 \leq t < T$ may have generated the observations. Assume that there are S such pairs at time t that maximize the probability of observations up to time t. At the next time instance $t+1$, one can arrive from any of the original S pairs by applying a set of S possible new inputs $d_0, \ldots, d_{S-1}$. The goal of a Viterbi update is to find the best transition from any of S previous states (x_t, s_t) to a particular new state (x_{T+1}, s_{t+1}) that maximizes the probability of all observations. Namely, we choose the transition as the best one if

$$\begin{aligned} & cost(x_{t+1}, s_{t+1}) = \\ & \quad \min_{(x_t, s_t)} \{ cost(x_t, s_t) \\ & \quad + \frac{1}{2} \log |O| + \frac{1}{2} \left(y_{t+1} - C x_{t+1|t} \right)' O^{-1} \left(y_{t+1} - C x_{t+1|t} \right) \\ & \quad - \log Pr(s_{t+1} | s_t) \} , \end{aligned} \tag{5.15}$$

where O denotes the variance of the innovation $y_{t+1} - C x_{t+1|t}$ as defined in Section 3.5. Of course, LDS state prediction $x_{t+1|t}$ is a function of (x_t, s_t) and s_{t+1}.

In addition to keeping track of the cost, one also needs to keep a record of best transitions as well as pairs (x_t, s_t) and variances associated with x_t which are necessary for Kalman update equations of $x_{t+1|t}$.

5.5 Learning

The task of learning the parameters of the mixed-state DBNs can be formulated as the problem of maximum likelihood learning in general Bayesian networks. This enables us to use the optimization approach of generalized EM presented in Section 2.3.

Assuming a knowledge of the sufficient statistics obtained in the inference phase, it is easy to show that the following parameter update equations result from the maximization step of the GEM (see Appendix C for detailed derivations):

$$\begin{aligned}
A_{new} &= \left(\sum_{t=1}^{T-1} \left\langle x_t x'_{t-1} \right\rangle - D_{new} \left\langle s_t x'_{t-1} \right\rangle \right) \left(\sum_{t=1}^{T-1} \left\langle x_{t-1} x'_{t-1} \right\rangle \right)^{-1} && (5.16) \\
Q_{new} &= \frac{1}{T-1} \sum_{t=1}^{T-1} \left(\langle x_t x'_t \rangle - A_{new} \langle x_{t-1} x'_t \rangle - D_{new} \langle s_t x'_t \rangle \right) && (5.17) \\
D_{new} &= \left(\sum_{t=1}^{T-1} \langle x_t s'_t \rangle - A_{new} \langle x_{t-1} s'_t \rangle \right) \left(\sum_{t=1}^{T-1} \langle s_t s'_t \rangle \right)^{-1} && (5.18) \\
C_{new} &= \left(\sum_{t=0}^{T-1} y_t \langle x'_t \rangle \right) \left(\sum_{t=0}^{T-1} \langle x_t x'_t \rangle \right)^{-1} && (5.19) \\
R_{new} &= \frac{1}{T} \sum_{t=0}^{T-1} \left(y_t y_t^t - C_{new} \langle x_t \rangle y'_t \right) && (5.20) \\
P_{new} &= \left(\sum_{t=1}^{T-1} \left\langle s_t s'_{t-1} \right\rangle \right) \mathrm{diag} \left(\sum_{t=1}^{T-1} \langle s_t \rangle \right)^{-1} && (5.21) \\
\pi_{0,new} &= \langle s_0 \rangle . && (5.22)
\end{aligned}$$

All the variable statistics are, as required by GEM, evaluated in the network with the parameter values before update. Some of the equations, notably the ones for A and D parameters (Equations 5.16 and 5.18) are coupled and have to be solved simultaneously. This is, however, not a problem since the solution can be formulated easily as a solution to the system of two linear equations with two matrix unknowns. Finally, it is obvious that the above equations represent a generalization of the parameter update equations of zero-input LDS models derived in Section 3.5.

In the case of variational inference, the parameter learning equations retain the same form. However, computation becomes significantly simpler given the fact that coupled statistics such as $\langle x_t s'_t \rangle$ can be factored in the form of $\langle x_t \rangle \langle s_t' \rangle$. Therefore, a formula for the update of D, for instance, becomes

$$D_{new} = \left(\sum_{t=1}^{T-1} \langle x_t \rangle \langle s_t \rangle' - A_{new} \langle x_{t-1} \rangle \langle s_t \rangle' \right) \left(\sum_{t=1}^{T-1} \langle s_t s_t' \rangle \right)^{-1} .$$

CHAPTER 6

COUPLED HIDDEN MARKOV MODELS

6.1 Introduction

Many processes in nature are products of complex interactions among a multitude of simple processes. Each of the simple processes may have its realization in a different observation domain or *modality.* The task is often to infer some information about the underlying global process.

Consider, for instance, the production of the human speech. The air streaming through the vocal tract gets modulated by different motions and changes of vocal cords, nasal cavity, and lips. We perceive the sound signal using our auditory apparatus. However, it is well known that a part of the spoken information is transmitted visually, through images of lip movements. In the absence of reliable audio information, such as in a noisy vehicle, humans often supplement the process of sound disambiguation using visual appearance of the accompanying lip movements. Thus, we unconsciously use the fact that the sound production is a result of lip motion correlated with many other processes and that by collecting information from different observation domains (audio signal and visual images) we can acquire a better estimate of the underlying word concept associated with that sound. Hence, for instance, if one could measure the physical motion of vocal cords one could possibly attain an even better estimate of the uttered word.

As a second example, consider the process of natural human communication. Elements of such communication are not only the speech but also hand, arm, and body movements, also known as gestures. When a person tries to describe a position of some object in space he or she often points at the object and simultaneously utters: “Look at that thing!” Clearly, the information about the object’s position can be transmitted much more effectively if one uses both the spoken language and the gestures. However, the production dynamics of the hand motion and the vocal tract changes are obviously *different* even though they are somewhat *correlated.*

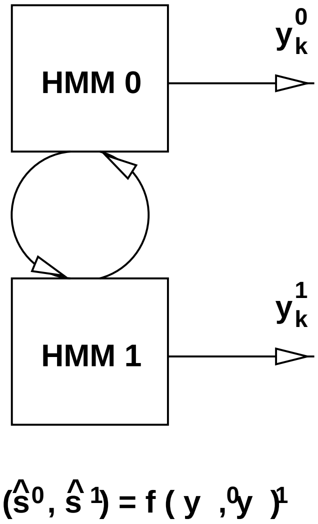


Figure 6.1 Block diagram of two coupled systems with different internal dynamics. In this example, each system is modeled as a hidden Markov model.

The above two examples motivate consideration of models of such physical systems that consist of a number of subprocesses with different but coupled internal dynamics and possibly different observation domains. A block diagram depiction of such a system is presented in Figure 6.1.

The rest of this chapter considers one such model of coupled systems. We assume that the coupling is on the level of discrete-valued concepts while the observations may belong to either discrete or continuous valued spaces. Such models are denoted *coupled hidden Markov models.* They are also often referred to as *multimodal hidden Markov models*, indicating that their observations as well as dynamics come from different observation/production modalities. Coupled DBN-like models were first considered for multimodal information fusion by Hennecke and Stork [80] in their *Boltzmann zipper* architecture. Later on, Brand [81, 75] has proposed a coupled HMM topology that coincides with the very basic coupled DBN topology utilized in this chapter.

6.2 Model

Coupled hidden Markov models are a generalization of the concept of ordinary HMMs introduced in Section 3.6. Namely, each state of one subsystem (or modality) at time t depends on the states of all other subsystems (or modalities) at time $t-1$. This dependency structure is depicted as a particular dynamic Bayesian network topology in Figure 6.2.

More precisely, we define an M-modal coupled HMM of length T as the following joint pdf:

$$P(\mathcal{S}^{(0)}, \ldots, \mathcal{S}^{(M-1)}, \mathcal{Y}^{(0)}, \ldots, \mathcal{Y}^{(M-1)}) =$$

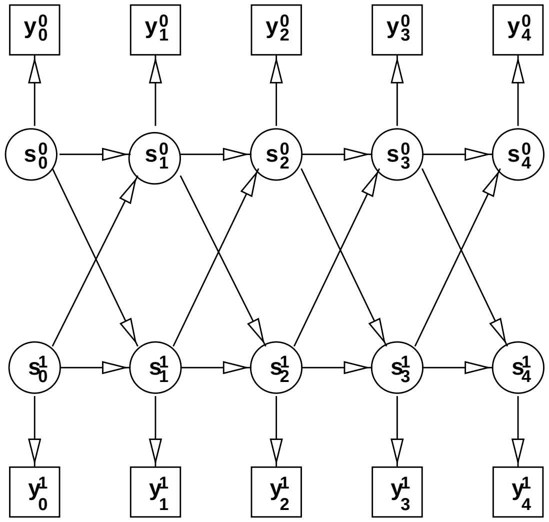


Figure 6.2 Dependency graph of a two-subsystem (bimodal) coupled hidden Markov model.

$$\begin{aligned}
&\prod_{t=1}^{T-1}\prod_{n=0}^{M-1} Pr(s_t^{(n)}|s_{t-1}^{(0)},\ldots,s_{t-1}^{(M-1)})Pr(s_t^{(n)}) \\
&\prod_{t=0}^{T-1}\prod_{n=0}^{M-1} Pr(y_t^{(n)}|s_t^{(n)})
\end{aligned} \tag{6.1}$$

or, equivalently, its Hamiltonian:

$$\begin{aligned}
&H(\mathcal{S}^{(0)},\ldots,\mathcal{S}^{(M-1)},\mathcal{Y}^{(0)},\ldots,\mathcal{Y}^{(M-1)}) = \\
&\quad -\sum_{t=1}^{T-1}\sum_{n=0}^{M-1}\log Pr(s_t^{(n)}|s_{t-1}^{(0)},\ldots,s_{t-1}^{(M-1)}) - \sum_{n=0}^{M-1}\log Pr(s_0^{(n)}) \\
&\quad -\sum_{t=0}^{T-1}\sum_{n=0}^{M-1}\log Pr(y_t^{(n)}|s_t^{(n)}),
\end{aligned} \tag{6.2}$$

where $\mathcal{S}^{(n)} = \{s_0^{(n)},\ldots,s_{T-1}^{(n)}\}$ denotes a sequence of hidden state variables of modality n, and $\mathcal{Y}^{(n)} = \{y_0^{(n)},\ldots,y_{T-1}^{(n)}\}$ denotes a sequence of observations of the same modality. The state space of modality n is assumed to be of dimension $N(n)$, i.e., $s_t^{(n)} \in \{e_0,\ldots,e_{N(n)-1}\}$.

Note that the following assumption holds in this model:

> The state of any modality n at time t is (conditionally) *independent* of the states of all other modalities at the same time t given that the states of all modalities are known at time $t-1$.

This is reflected in the form of the transition pdf in Equation 6.2:

$$Pr(s_t^{(0)},\ldots,s_t^{(M-1)}|s_{t-1}^{(0)},\ldots,s_{t-1}^{(M-1)}) = \prod_{n=0}^{M-1} Pr(s_t^{(n)}|s_{t-1}^{(0)},\ldots,s_{t-1}^{(M-1)}).$$

At first glance it may seem that this assumption constricts the generality of my model. However, it will become clear as the model is developed further in subsequent sections that, in fact, the assumption yields a very general coupled DBN framework. To construct computationally efficient models we will need to further simplify the structure of this pdf.

6.3 Inference

Even though the topology of a coupled HMM resembles that of an ordinary HMM (in fact, it contains ordinary HMMs as its subgraphs), the inference schemes of ordinary HMMs are not directly applicable to the coupled ones. Of course, one could attempt to apply the classical forward-backward algorithm of Chapter 3. However, the complexity of formulas will increase significantly. We therefore attempt to find other (preferably simpler) ways of solving the inference problem.

For sake of completeness, we now state the inference problem of coupled HMMs. The task is to find the conditional pdf over the space of hidden states given some fixed sequence of observations of length T of all M modalities:

$$Pr(\mathcal{S}^{(0)}, \ldots, \mathcal{S}^{(M-1)} | \mathcal{Y}^{(0)}, \ldots, \mathcal{Y}^{(M-1)}).$$

In the following sections we will exploit specific structures of the transition pdfs of coupled HMM to come up with computationally appealing solutions to the inference problem.

6.3.1 Naive inference

Naive inference is, in general, not a computationally efficient approach to inference in coupled HMMs. However, it is important to study this approach given that it is often the first to come to mind.

Consider the dependency graph of a general coupled HMM in Figure 6.2. Instead of looking at M hidden states at time t as the states of M individual modalities, one can group them into an M-tuple $\sigma_t = (s_t^{(0)}, \ldots, s_t^{(M-1)})$. The model can now be formulated in terms of the new, *complex* variable σ_t. This is depicted in Figure 6.3. Namely, we perform the Cartesian product of all sub-HMMs' (HMMs corresponding to individual modalities) state spaces to construct the new, complex state space. This new state space has dimension of $N = N(0) \cdot \ldots \cdot N(M-1)$, i.e., $\sigma_{\cdot} \in \{e_0, \ldots, e_{N-1}\}$. The hidden state transition pdf of the new model becomes

$$Pr(\sigma_t | \sigma_{t-1}) = \prod_{n=0}^{M-1} Pr(s_t^{(n)} | s_{t-1}^{(0)}, \ldots, s_{t-1}^{(M-1)}). \tag{6.3}$$

Clearly, P of Equation 6.3 can be described as a table (matrix) of $N \times N$ entries. A similar Cartesian product argument can be used to construct the space of observations $\psi_t = (y_t^{(0)}, \ldots, y_t^{(M-1)})$.

As is clear now, we have reduced the M-modal coupled HMM to an ordinary but complex HMM. The drawback is that the state and observation space dimensions have increased exponentially! Assume

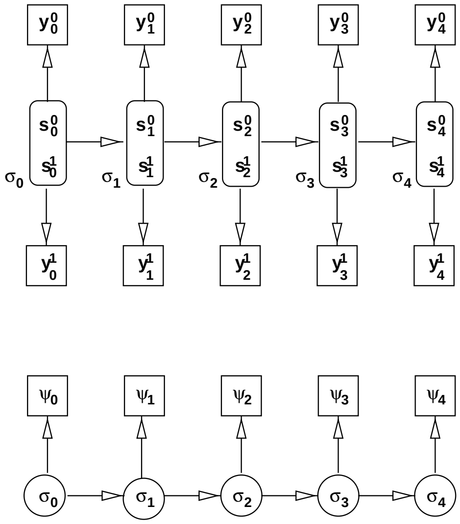


Figure 6.3 Naive approach to inference in coupled HMMs. The M-modal coupled HMM is reduced to an ordinary HMM whose hidden states σ_t live in the Cartesian product of M state spaces of the original modal HMMs. The dimension of this new state space is $N = N(0) \cdot \ldots \cdot N(M-1)$.

for a moment that each sub-HMM had a state space of the same dimension N_s. Complex HMM will have the state space of dimension N_s^M. This in turn means an increase in computational complexity of inference (see Section 3.6). Furthermore, the number of parameters (or, more precisely, their dimension) has increased, too (state transition table P has N_s^{2M} entries). This can seriously affect the model's parameter learning phase. It will require a tremendous increase in the number of training data samples to achieve the same parameter estimate accuracies.

In the above discussion, however, we have disregarded the fact that often sub-HMM transition matrices $Pr(s_t^{(n)}|s_{t-1}^{(0)}, \ldots, s_{t-1}^{(M-1)})$ have a *sparse* structure. Namely, only a few of all possible transitions are actually allowed. Bringing that into the consideration of complexity (dimensionality) of the HMM enables pruning-out a substantial number of states that can never be visited. Hence, the effective dimension N_{eff} of the state space of our complex HMM will become $N_{eff} \ll N$.

6.3.2 Variational inference

The previous section concluded that naive reduction of the coupled HMM inference problem to that of a complex, ordinary HMM carries with an increase in computational complexity and model parameter dimensions.

To deal with these problems we now consider an alternative, approximate inference approach. As the approximation tool, we use the variational inference technique of Section 2.2.3. However, direct application of variational inference to the coupled model defined in Equation 6.2 will not yield any significant reduction in the model complexity. We therefore consider the following factorized form of state transitional pdf:

$$Pr(s_t^{(n)}|s_{t-1}^{(0)},\ldots,s_{t-1}^{(M-1)}) = \sum_{m=0}^{M-1} w^{(n)}(m) Pr(s_t^{(n)}|s_{t-1}^{(m)}), \tag{6.4}$$

where $w^{(n)}(m)$ defines an *intermodal weight* with which modality m influences modality n. The weights sum up to one according to

$$\sum_{m=0}^{M-1} w^{(n)}(m) = 1,\ n = 0,\ldots,M-1. \tag{6.5}$$

Hence, the transition pdf for each mode M is a combination of M "bimodal" transition pdfs $Pr(s_t^{(n)}|s_{t-1}^{(m)})$ ("bimodal" here denotes that the pdfs link only the state variables of two modalities n and m out of M possible variables). This factorization yields a total of $M \times M$ state transition pdfs and $M \times M$ weights $w^{\cdot}(\cdot)$ that completely describe the coupled model's state transition pdf. Suppose again that each modality's state space has the same dimension N_s. The coupled model's transition pdf then has a total of $M^2 N_s^2 + M^2$ parameters compared to N_s^M of the naive approach in Section 6.3.1.

Given the above factorization we next consider three special cases of this model. In the first case we assume that the weights w are fixed a priori. The second case relaxes this assumption and adaptively determines the weights based on some prior model. Finally, the third model allows the adaptation of time-varying weights w_t.

6.3.2.1 Fixed weights model

Consider the case of the state transition pdf factorization of Equation 6.4 where all intermodal weights $w^{(n)}(m), n, m = 0, \ldots, M-1$ are fixed. The model's joint pdf (or, equivalently, its Hamiltonian) can then be written as

$$\begin{aligned} H = & \\ & -\sum_{t=1}^{T-1}\sum_{n=0}^{M-1}\sum_{m=0}^{M-1} w^{(n)}(m) s_t^{(n)\prime} \log P^{(n,m)} s_{t-1}^{(m)} - \sum_{n=0}^{M-1} s_0^{(n)\prime} \log \pi_0^{(n)} \\ & -\sum_{t=0}^{T-1}\sum_{n=0}^{M-1} \log Pr(y_t^{(n)}|s_t^{(n)}). \end{aligned} \tag{6.6}$$

The network associated with this factorization is shown in Figure 6.4. In accordance with the variational inference technique of Section 2.2.3, we choose the approximating distribution Q to be

$$\begin{aligned} H_Q = & \\ & -\sum_{t=1}^{T-1}\sum_{n=0}^{M-1} w^{(n)}(n) s_t^{(n)\prime} \log P^{(n,n)} s_{t-1}^{(n)} - \sum_{n=0}^{M-1} s_0^{(n)\prime} \log \pi_0^{(n)} \\ & -\sum_{t=0}^{T-1}\sum_{n=0}^{M-1} \log Pr(y_t^{(n)}|s_t^{(n)}) \end{aligned}$$

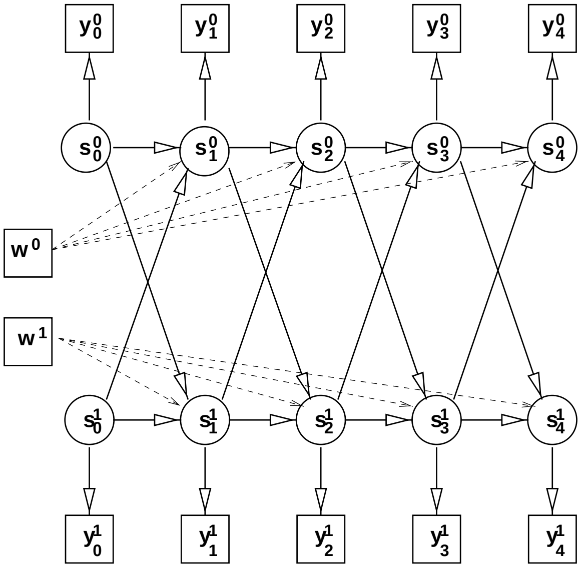


Figure 6.4 Dependency graph of a coupled HMM with fixed intermodal weights.

$$-\sum_{t=0}^{T-1}\sum_{n=0}^{M-1} s_t^{(n)\prime}\log q_t^{(n)}. \tag{6.7}$$

This approximating network Q is depicted in Figure 6.5. Variational parameters of the approximating network Q are denoted by $q_t^{(n)}(m)$ and are represented as parameter vectors $q_t^{(n)}$. The choice of the approximating topology seems straightforward: by decoupling the sub-HMMs we allow for easy inference in each of the submodels.

The task of variational inference is to find the "best" values of q, the ones that minimize the KL-distance between P and Q (see Section 2.2.3). Using Theorem 1, after a couple of intermediate steps (detailed in Appendix D) one arrives at the following optimal estimates of the variational parameters:

$$\log q_\tau^{(l)} = \begin{cases} \sum_{m=0,m\neq l}^{M-1} w^{(m)}(l)\log P^{(m,l)\prime}\left\langle s_1^{(m)}\right\rangle & \tau = 0 \\ \sum_{m=0,m\neq l}^{M-1} w^{(l)}(m)\log P^{(l,m)}\left\langle s_{\tau-1}^{(m)}\right\rangle & \\ \quad + \sum_{m=0,m\neq l}^{M-1} w^{(m)}(l)\log P^{(m,l)\prime}\left\langle s_{\tau+1}^{(m)}\right\rangle & 0 < \tau < T-1 \\ \sum_{m=0,m\neq l}^{M-1} w^{(l)}(m)\log P^{(l,m)}\left\langle s_{T-2}^{(m)}\right\rangle & \tau = T-1. \end{cases} \tag{6.8}$$

The log of a variational parameter of modality l at time τ is a linear combination of the state estimates of all other modalities $0,\ldots,l-1,l+1,\ldots,M-1$ at the neighboring time instances $\tau-1$ and $\tau+1$. One can also think of $q_\tau^{(l)}$ as the "probability"[1] of state $s_\tau^{(l)}$ given the states of all other modalities. This "probability" is proportional to a weighted geometric mean (with weights $w^{(l)}(m)$) of the probabilities of states of other modalities ($s_{\tau-1}^{(m)}$ and $s_{\tau+1}^{(m)}$) mapped into the modality l (with mappings $P^{(m,l)\prime}$ and

[1] Strictly speaking $q_\tau^{(l)}$ is not a (discrete) pdf since $\sum_{m=0}^{N(l)-1} \neq 1$.

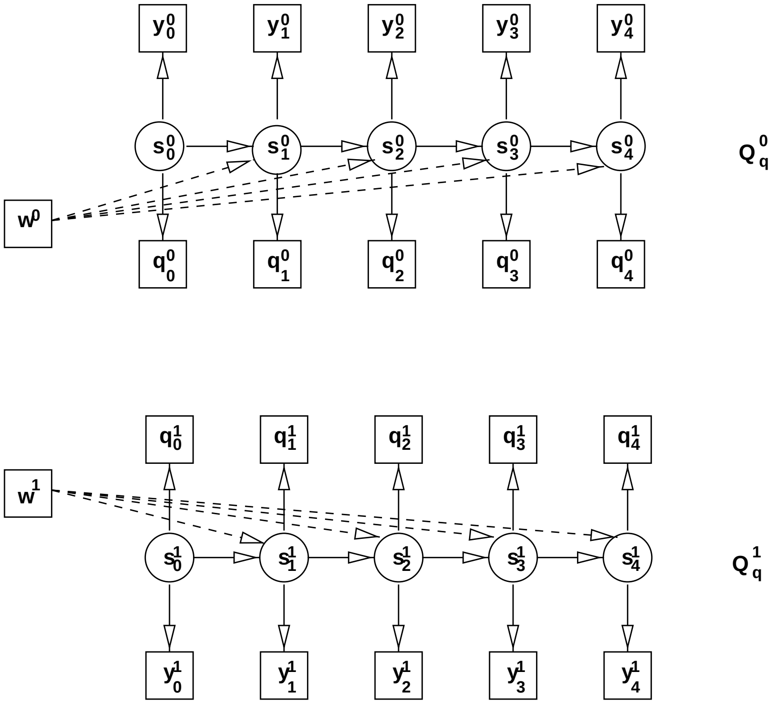


Figure 6.5 Factorization of fixed-weight coupled HMM.

$P^{(l,m)}$):

$$q_\tau^{(l)} = \prod_{m=0,m\neq l}^{M-1} \left(P^{(l,m)} \left\langle s_{\tau-1}^{(m)} \right\rangle \right)^{w^{(l)}(m)} \cdot \prod_{m=0,m\neq l}^{M-1} \left(P^{(m,l)\prime} \left\langle s_{\tau+1}^{(m)} \right\rangle \right)^{w^{(m)}(l)} .$$

This fact is also figuratively depicted in Figure 6.6. For instance, consider the following brief example of a bimodal coupled HMM. Assuming that in the absence of noise the model performs well, its transition pdf parameters are given as

$$\begin{array}{cc} P^{(1,1)} = \begin{bmatrix} 0.8 & 0.1 \\ 0.2 & 0.9 \end{bmatrix} & P^{(1,2)} = \begin{bmatrix} 0.9 & 0.4 & 0.1 \\ 0.1 & 0.6 & 0.9 \end{bmatrix} \\ P^{(2,1)} = \begin{bmatrix} 0.5 & 0.3 \\ 0.3 & 0.1 \\ 0.2 & 0.6 \end{bmatrix} & P^{(2,2)} = \begin{bmatrix} 0.9 & 0.2 & 0 \\ 0.1 & 0.7 & 0.1 \\ 0 & 0.1 & 0.9 \end{bmatrix} \end{array} .$$

Let the state distribution of modality 1 at time τ be biased towards state one:

$$\left\langle s_\tau^{(1)} \right\rangle = \begin{bmatrix} 0.9 \\ 0.1 \end{bmatrix} .$$

One then may expect that with high likelihood the observation in modality 1 at the next instance $\tau + 1$ will again come from state 1. However, the observation in modality 1 at that time instance is ambiguous

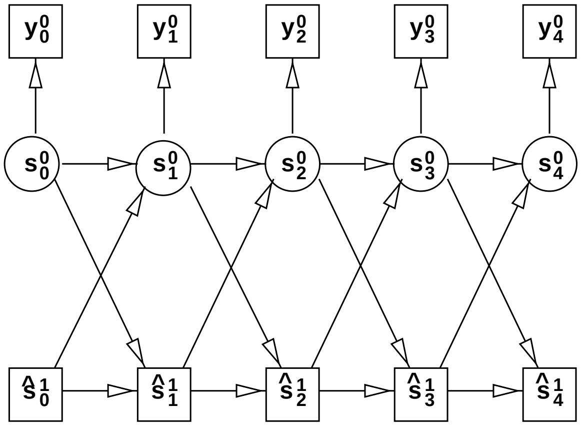


Figure 6.6 Result of variational inference on one sub-HMM of the approximating distribution Q. Inference in the submodel is performed by assuming that all other submodels have fixed and known hidden states.

because of noise, say

$$Pr(y^{(1)}_{\tau+1}|s^{(1)}_{\tau+1}) = \begin{bmatrix} 0.35 \\ 0.65 \end{bmatrix}.$$

Based only on modality 1, the likelihood of its states at $\tau + 1$ then becomes

$$\left\langle s^{(1)}_{\tau+1} \right\rangle = c\text{diag}(Pr(y^{(1)}_{\tau+1}|s^{(1)}_{\tau+1}))P^{(1,1)} \left\langle s_{\tau} \right\rangle = \begin{bmatrix} 0.59 \\ 0.41 \end{bmatrix},$$

very close to a uniform distribution. On the other hand, assume that the state estimates in the second modality are less ambiguous:

$$\left\langle s^{(2)}_{\tau} \right\rangle = \begin{bmatrix} 0.9 \\ 0.1 \\ 0 \end{bmatrix},$$

and

$$\left\langle s^{(2)}_{\tau+2} \right\rangle = \begin{bmatrix} 0.7 \\ 0.2 \\ 0.1 \end{bmatrix}.$$

If the influence of the two modalities is measured by the weight factor $w^{(1)}(2) = 0.3$, the state distribution estimate of modality 1 with the influence of modality 2 now becomes

$$\left\langle s^{(1)}_{\tau+1} \right\rangle = \begin{bmatrix} 0.73 \\ 0.27 \end{bmatrix},$$

clearly better then the unimodal estimate itself.

In summary, to perform inference in the approximating network Q one applies the following algorithm:

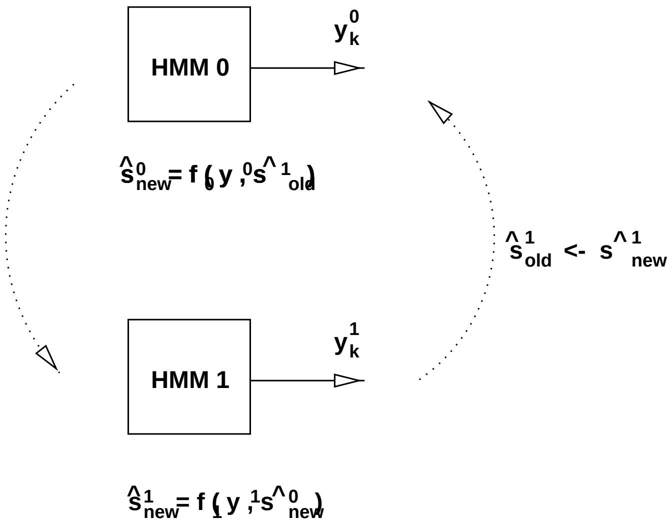

HMM 0
y 0 k
^s 0 new = f (y 0 , ^s 1 old)
^s 1 old <- ^s 1 new
HMM 1
y 1 k
^s 1 new = f (y 1 , ^s 0 new)

Figure 6.7 Figurative representation of the variational inference algorithm for coupled HMMs. At each iteration, state variables of all submodels except one are assumed constant (subscript "old"). The remaining submodels' states are then re-estimated (s_{new}) as functions of the observations y and the fixed states of other models s_{old}.

```
Cost = $\infty$;
Initialize $\langle s^{(m)} \rangle$ for all $m = 0, \ldots, M-1$;
while (error > maxError) {
    for ( $l = 0 : M-1$ ) {
        Update $q_{\cdot}^{(l)}$ from $\left\langle s_{\cdot}^{(m)} \right\rangle, m = 0, \ldots, l-1, l+1, \ldots, M-1$ using Equation 6.8;
        Estimate $\left\langle s_{\cdot}^{(l)} \right\rangle$ from $y_{\cdot}^{(l)}$ and $q_{\cdot}^{(l)}$ using ordinary HMM inference;
    }
    Update Cost using Equation 6.9;
    error $\leftarrow$ ( oldCost - Cost ) / Cost;
}
```

As a consequence of the employed factorization, we have

$$\left\langle s_{\tau}^{(n)} s_{\tau}^{(m)\prime} \right\rangle = \left\langle s_{\tau}^{(n)} \right\rangle \left\langle s_{\tau}^{(m)} \right\rangle', \ \forall m \neq n.$$

The idea behind the inference algorithm is depicted in Figure 6.7.

The cost of inference at each step is given as

$$\langle H - H_Q \rangle - \log Z_Q =$$
$$\sum_{t=1}^{T-1} \sum_{n=0}^{M-1} \left[- \sum_{m=0, m \neq n}^{M-1} w^{(n)}(m) \mathrm{tr} \left\{ \log P^{(n,m)} \left\langle s_{t-1}^{(m)} s_t^{(n)\prime} \right\rangle \right\} + \left\langle s_t^{(n)} \right\rangle' \log q_t^{(n)} \right]$$

$$+\sum_{n=0}^{M-1}\left\langle s_0^{(n)}\right\rangle' \log q_0^{(n)} - \sum_{n=0}^{M-1}\log Z_Q^{(n)}, \tag{6.9}$$

where $\log Z_Q^{(n)}$ represents the log likelihood of "observations" $y_{\cdot}^{(n)}$ and $q_{\cdot}^{(n)}$ in the nth sub-HMM.

6.3.2.2 Adaptive intermodal weights model

The previous section considered a factorized pdf model with the known intermodal weight factors w. However, we did not answer the question of how to determine the weights. In the following model we assume that the weights are random variables distributed according to some known probability distribution.

Consider the intermodal weight $w^{(n)}(m)$. This weight describes the influence of modality m on modality n. Define the following discrete pdf on weights $w^{(n)}(\cdot)$:

$$Pr(w^{(n)} = m) = W^{(n)}(m),$$

where $\sum_{m=0}^{M-1} W^{(n)}(m) = 1,\ \forall n = 0, \ldots, M-1$. The joint pdf of the coupled network then becomes:

$$\begin{aligned}
H = &-\sum_{t=1}^{T-1}\sum_{n=0}^{M-1}\sum_{m=0}^{M-1} w^{(n)}(m) s_t^{(n)\prime} \log P^{(n,m)} s_{t-1}^{(m)} - \sum_{n=0}^{M-1} s_0^{(n)\prime}\log \pi_0^{(n)} \\
&-\sum_{t=0}^{T-1}\sum_{n=0}^{M-1}\log Pr(y_t^{(n)}|s_t^{(n)}) \\
&-(T-1)\sum_{n=0}^{M-1}\sum_{m=0}^{M-1} w^{(n)}(m)\log W^{(n)}(m).
\end{aligned} \tag{6.10}$$

The difference between Equation 6.6 and Equation 6.10 is in the additional factor $w^{(\cdot)}(\cdot)\log W^{(\cdot)}(\cdot)$ that accounts for uncertainty in w. The dependency graph of this pdf is shown in Figure 6.8.

Similar to the factorization of Section 6.3.2.1, we introduce a new factorized approximating distribution model:

$$\begin{aligned}
H_Q = &-\sum_{t=1}^{T-1}\sum_{n=0}^{M-1} c^{(n)} s_t^{(n)\prime}\log P^{(n,n)} s_{t-1}^{(n)} - \sum_{n=0}^{M-1} s_0^{(n)\prime}\log \pi_0^{(n)} \\
&-\sum_{t=0}^{T-1}\sum_{n=0}^{M-1}\log Pr(y_t^{(n)}|s_t^{(n)}) \\
&-\sum_{t=0}^{T-1}\sum_{n=0}^{M-1} s_t^{(n)\prime}\log q_t^{(n)} \\
&-(T-1)\sum_{n=0}^{M-1}\sum_{m=0}^{M-1} w^{(n)}(m)\log W^{(n)}(m) \\
&-(T-1)\sum_{n=0}^{M-1}\sum_{m=0}^{M-1} w^{(n)}(m)\log r^{(n)}(m).
\end{aligned} \tag{6.11}$$

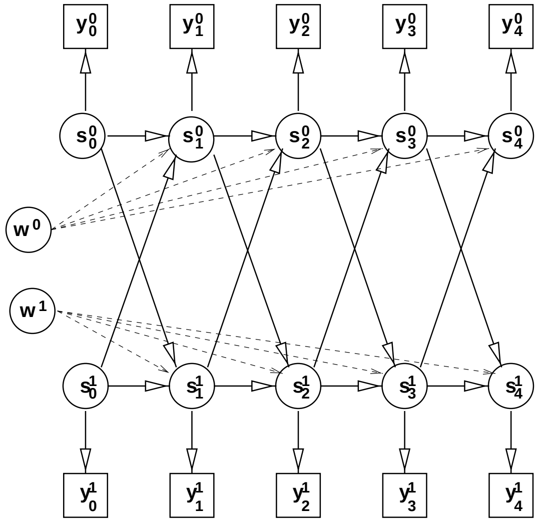


Figure 6.8 Dependency graph of a coupled HMM with adaptive intermodal weights.

The Bayesian network equivalent of this pdf is shown in Figure 6.9. The original network has been decoupled into M independent and overparameterized HMMs and M independent W-networks. This is achieved by introducing variational parameters q, c, and r, as shown in Figure 6.9.

Following now-familiar steps of the variational inference approach (see Section 2.2.3) leads to the optimal values of variational parameters:

$$\log q_\tau^{(l)} = \begin{cases} \sum_{m=0,m\neq l}^{M-1} \left\langle w^{(m)}(l) \right\rangle \log P^{(m,l)\prime} \left\langle s_1^{(m)} \right\rangle & \tau = 0 \\ \sum_{m=0,m\neq l}^{M-1} \left\langle w^{(l)}(m) \right\rangle \log P^{(l,m)} \left\langle s_{\tau-1}^{(m)} \right\rangle & \\ \quad + \sum_{m=0,m\neq l}^{M-1} \left\langle w^{(m)}(l) \right\rangle \log P^{(m,l)\prime} \left\langle s_{\tau+1}^{(m)} \right\rangle & 0 < \tau < T-1 \\ \sum_{m=0,m\neq l}^{M-1} \left\langle w^{(l)}(m) \right\rangle \log P^{(l,m)} \left\langle s_{T-2}^{(m)} \right\rangle & \tau = T-1 \end{cases} \tag{6.12}$$

$$c^{(l)} = \left\langle w^{(l)}(l) \right\rangle \tag{6.13}$$

$$\log r^{(l)}(k) = \frac{1}{T-1} \sum_{t=1}^{T-1} \mathrm{tr} \left\{ \left\langle s_t^{(l)} s_{t-1}^{(k)\prime} \right\rangle' \log P^{(l,k)} \right\}. \tag{6.14}$$

All equations hold for $k, l = 0, \ldots, M-1$. Intermediate steps of this derivation can be found in Appendix D.2.

Equation 6.12 for variational parameter q resembles Equation 6.8 that defines the parameter in the case of fixed weights. Here, however, the weights are substituted with their probabilities (or, equivalently, expectations $\langle \cdot \rangle$). Additional equations (6.13 and 6.14) are nonetheless necessary to account for the stochastic nature of the weights. In particular, Equation 6.14 compounds the existence of a “correlation” between modalities k and l. The term “correlation” here loosely refers to the similarity of two state distributions $\left\langle s_t^{(l)} \right\rangle$ and $\left\langle s_{t-1}^{(k)} \right\rangle$ with respect to the mapping $\log P^{(l,k)}$. The higher the similarity, the higher the r will be.

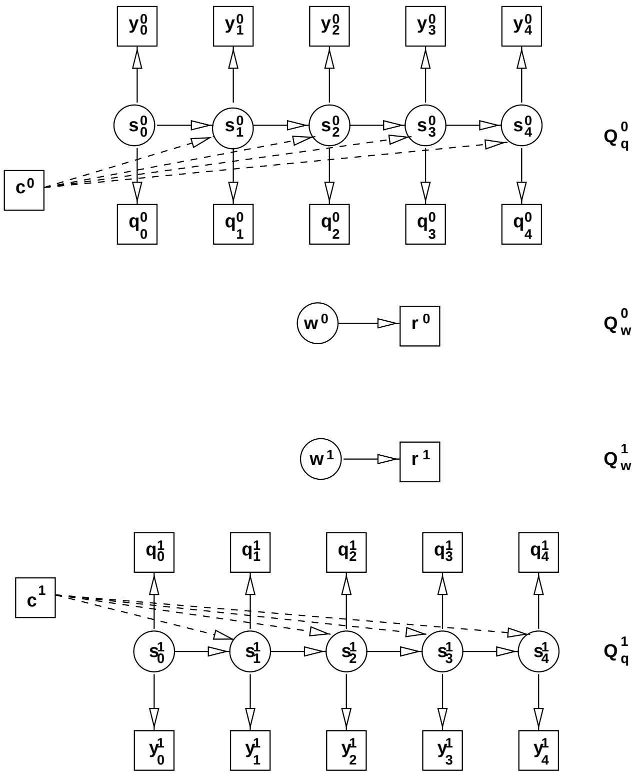


Figure 6.9 Factorized joint pdf for variational inference of a coupled HMM with adaptive intermodal weights. Factorization yields four independent networks that isolate the four sets of hidden variables of the original model.

Given this approximation, the inference algorithm becomes

```
Cost = $\infty$;
Initialize $\left\langle s_\cdot^{(m)} \right\rangle$ and $\langle w^{(m)} \rangle$ for all $m = 0, \ldots, M-1$;
while (error > maxError) {
    for ( $l = 0 : M-1$ ) {
        Update $q_\cdot^{(l)}$ from $\left\langle s_\cdot^{(m)} \right\rangle, m = 0, \ldots, l-1, l+1, \ldots, M-1$ and $\langle w^{(m)} \rangle$
                using Equation 6.12;
        Estimate $\left\langle s_\cdot^{(l)} \right\rangle$ from $y_\cdot^{(l)}$, $c^{(l)}$, and $q_\cdot^{(l)}$ using ordinary HMM inference on
                subnet $Q_q^{(l)}$;
    }
    Update $\log r^{(\cdot)}$ from $\left\langle s_\cdot^{(\cdot)} \right\rangle$ using Equation 6.14;
    Estimate $w^{(\cdot)}$ from $r^{(\cdot)}$ and $W$ on subnet $Q_w^{(\cdot)}$;
    Update $c^{(\cdot)}$ from $w^{(\cdot)}(\cdot)$ using Equation 6.13;
    Update Cost using Equation 6.15;
    error ← ( oldCost - Cost ) / Cost;
}
```

The cost term can be found as

$$\begin{aligned}
&\langle H - H_Q \rangle - \log Z_Q = \\
&\quad \sum_{t=1}^{T-1} \sum_{n=0}^{M-1} \left[- \sum_{m=0, m \neq n}^{M-1} \left\langle w^{(n)}(m) \right\rangle \mathrm{tr} \left\{ \log P^{(n,m)} \left\langle s_{t-1}^{(m)} s_t^{(n)\prime} \right\rangle \right\} + \left\langle s_t^{(n)} \right\rangle ' \log q_t^{(n)} \right] \\
&\quad + \sum_{n=0}^{M-1} \left\langle s_0^{(n)} \right\rangle ' \log q_0^{(n)} \\
&\quad + \sum_{n=0}^{M-1} \left\langle w^{(n)} \right\rangle ' \log r^{(n)} \\
&\quad - \sum_{n=0}^{M-1} \log Z_{Q_q}^{(n)} - \sum_{n=0}^{M-1} \log Z_{Q_w}^{(n)},
\end{aligned} \tag{6.15}$$

where $Z_{Q_q}(n)$ denotes the "probability" of observations in the nth sub-HMM and $Z_{Q_w}^{(n)}$ represents the "probability" of observations in the nth W-network.

6.3.2.3 Adaptive time-varying intermodal weights model

The adaptive time-varying weights model lifts the restriction of equal intermodal weights at all time instances. By allowing the time variation of weight factors, the model allows for variable levels of interaction between modalities at different time instances. However, in order to make this model complete one needs to define the type of temporal dependencies imposed on intermodal weights. We choose the hidden Markov model framework to describe the evolution of weights in time.

Formally, the joint pdf model is defined by the following Hamiltonian:

$$\begin{aligned}
H = & - \sum_{t=1}^{T-1} \sum_{n=0}^{M-1} \sum_{m=0}^{M-1} w_t^{(n)}(m) s_t^{(n)\prime} \log P^{(n,m)} s_{t-1}^{(m)} - \sum_{n=0}^{M-1} s_0^{(n)\prime} \log pi_0^{(n)} \\
& - \sum_{t=0}^{T-1} \sum_{n=0}^{M-1} \log Pr(y_t^{(n)} | s_t^{(n)}) \\
& - \sum_{t=1}^{T-1} \sum_{n=0}^{M-1} w_t^{(n)\prime} \log W^{(n)} w_{t-1}^{(n)} - \sum_{n=0}^{M-1} w_0^{(n)\prime} \log \rho_0^{(n)}.
\end{aligned} \tag{6.16}$$

The dynamic Bayesian network equivalent of this pdf is depicted in Figure 6.10. Parameters $W^{(n)}$ now describe transition pdfs for HMMs of intermodal weight factors, and $\rho_0^{(n)}$ is the initial pdf of intermodal weights of modality n. This pdf is assumed to be fixed to $\rho_0^{(n)}(k) = \delta(n-k)$, i.e., it is always the case that $w_0^{(n)}(k) = \delta(n-k)$.

The complexity of this model clearly calls for an approximate inference. Following the nature of the rest of this work, we choose the variational inference approach of Section 2.2.3. Accordingly, we need to

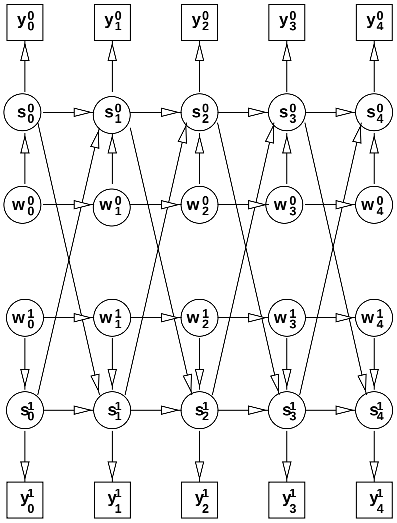


Figure 6.10 Dependency graph of a coupled HMM with time-varying adaptive weights. Evolution of intermodal factors is modeled as another dynamic Bayesian network.

define a class of parameterized (and factorized) distributions. The following Bayesian network topology naturally comes to mind:

$$
\begin{aligned}
H_Q = & -\sum_{t=1}^{T-1} \sum n = 0^{M-1} c_t^{(n)} s_t^{(n)\prime} \log Pr(n,n) s_{t-1}^{(n)} - \sum_{n=0}^{M-1} s_0^{(n)\prime} \log pi_0^{(n)} \\
& - \sum_{t=0}^{T-1} \sum_{n=0}^{M-1} \log Pr(y_t^{(n)} | s_t^{(n)}) \\
& - \sum_{t=0}^{T-1} \sum_{n=0}^{M-1} s_t^{(n)\prime} \log q_t^{(n)} \\
& - \sum_{t=1}^{T-1} \sum_{n=0}^{M-1} w_t^{(n)\prime} \log W^{(n)} w_{t-1}^{(n)} - \sum_{t=1}^{T-1} \sum_{n=0}^{M-1} w_t^{(n)} \log r_t^{(n)} \\
& - \sum_{n=0}^{M-1} w_0^{(n)\prime} \log \rho_0^{(n)}.
\end{aligned} \tag{6.17}
$$

Figure 6.11 depicts the dependency graph of this network. Similar to the time-invariant model, we factorized the original pdf into M HMM-like pdfs for each state-space modality (denoted in Figure 6.11 by $Q_q^{(\cdot)}$) and M HMM-like pdfs for temporal modeling of intermodal weights ($Q_w^{(\cdot)}$ in Figure 6.11). Variational parameters q, c, and r are introduced for that purpose.

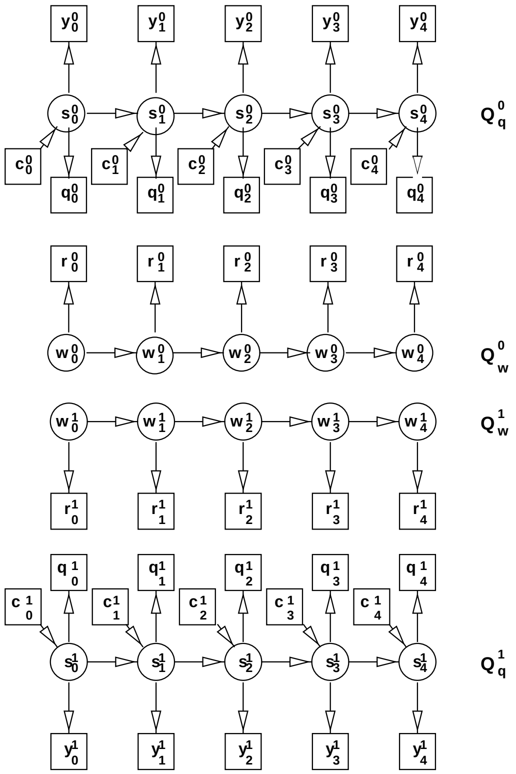


Figure 6.11 Factorized approximation of a coupled HMM with time-varying adaptive intermodal weights. Network consists of four independent subnets related to the four sets of hidden variables of the original model.

The application of Theorem 1 then yields optimal values of variational parameters that minimize the KL distance between the original pdf P and the approximating pdf Q:

$$\log q_\tau^{(l)} = \begin{cases} \sum_{m=0,m\neq l}^{M-1} \left\langle w_1^{(m)}(l) \right\rangle \log P^{(m,l)\prime} \left\langle s_1^{(m)} \right\rangle & \tau = 0 \\ \sum_{m=0,m\neq l}^{M-1} \left\langle w_\tau^{(l)}(m) \right\rangle \log P^{(l,m)} \left\langle s_{\tau-1}^{(m)} \right\rangle & \\ + \sum_{m=0,m\neq l}^{M-1} \left\langle w_{\tau+1}^{(m)}(l) \right\rangle \log P^{(m,l)\prime} \left\langle s_{\tau+1}^{(m)} \right\rangle & 0 < \tau < T-1 \\ \sum_{m=0,m\neq l}^{M-1} \left\langle w_{T-1}^{(l)}(m) \right\rangle \log P^{(l,m)} \left\langle s_{T-2}^{(m)} \right\rangle & \tau = T-1 \end{cases} \tag{6.18}$$

$$c_\tau^{(l)} = \left\langle w_\tau^{(l)}(l) \right\rangle \tag{6.19}$$

$$\log r_\tau^{(l)}(k) = \mathrm{tr} \left\{ \left\langle s_\tau^{(l)} s_{\tau-1}^{(k)}{}' \right\rangle ' \log P^{(l,k)} \right\} \tag{6.20}$$

with $l, k = 0, \ldots, M-1$.

The form and the steps of the approximate inference algorithm are identical to those of the time-invariant models in Section 6.3.2.2. The difference is in the parameter update rules, which now follow Equations 6.18 to 6.20. Moreover, to estimate $\left\langle w_\cdot^{(\cdot)} \right\rangle$ one needs to apply ordinary HMM inference in the appropriate $Q_w^{(\cdot)}$ network. In summary, the algorithm takes the following form:

```
Cost = $\infty$;
Initialize $\left\langle s_{\cdot}^{(m)} \right\rangle$ and $\left\langle w_{\cdot}^{(m)} \right\rangle$ for all $m = 0, \dots, M-1$;
while (error > maxError) {
    for ( $l = 0 : M-1$ ) {
        Update $q_{\cdot}^{(l)}$ from $\left\langle s_{\cdot}^{(m)} \right\rangle, m = 0, \dots, l-1, l+1, \dots, M-1$ and $\left\langle w_{\cdot}^{(m)} \right\rangle$
                using Equation 6.18;
        Estimate $\left\langle s_{\cdot}^{(l)} \right\rangle$ from $y_{\cdot}^{(l)}$, $c_{\cdot}^{(l)}$, and $q_{\cdot}^{(l)}$ using ordinary HMM inference
                on subnet $Q_q^{(l)}$;
    }
    Update $\log r_{\cdot}^{(\cdot)}$ from $\left\langle s_{\cdot}^{(\cdot)} \right\rangle$ using Equation 6.20;
    Estimate $w_{\cdot}^{(\cdot)}$ from $r_{\cdot}^{(\cdot)}$ using ordinary HMM inference on subnets $Q_w^{(\cdot)}$;
    Update $c_{\cdot}^{(\cdot)}$ from $w_{\cdot}^{(\cdot)}(\cdot)$ using Equation 6.19;
    Update Cost using Equation 6.21;
    error $\leftarrow$ ( oldCost - Cost ) / Cost;
}
```

The cost term can be found as

$$
\begin{aligned}
&\langle H - H_Q \rangle - \log Z_Q = \\
&\quad \sum_{t=1}^{T-1} \sum_{n=0}^{M-1} \left[- \sum_{m=0, m \neq n}^{M-1} \left\langle w_t^{(n)}(m) \right\rangle \operatorname{tr} \left\{ \log P^{(n,m)} \left\langle s_{t-1}^{(m)} s_t^{(n)\prime} \right\rangle \right\} + \left\langle s_t^{(n)} \right\rangle' \log q_t^{(n)} \right] \\
&\quad + \sum_{n=0}^{M-1} \left\langle s_0^{(n)} \right\rangle' \log q_0^{(n)} + \sum_{t=0}^{T-1} \sum_{n=0}^{M-1} \left\langle w_t^{(n)} \right\rangle' \log r_t^{(n)} \\
&\quad - \sum_{n=0}^{M-1} \log Z_{Q_q}^{(n)} - \sum_{n=0}^{M-1} \log Z_{Q_w}^{(n)},
\end{aligned}
\tag{6.21}
$$

where $Z_{Q_q}(n)$ denotes, as before, the "probability" of observations in the nth sub-HMM, and $Z_{Q_w}^{(n)}$ represents the "probability" of observations in the nth Q_w subnet.

6.4 Learning

Parameter estimation of coupled HMMs can be formulated as the problem of maximum likelihood learning in general Bayesian networks. This again leads one to the optimization approach of generalized EM of Section 2.3.

The inference phase in coupled HMMs, as discussed in the previous sections, provides one with sufficient statistics to update the model parameters. In this section, however, we discuss only the update equations for hidden state parameters of the models in question. Update equations for parameters of the observation pdfs follow exactly those derived for ordinary HMMs. Thus, depending on whether the observation pdfs are discrete, Gaussian, or mixture-of-Gaussians, for instance, the parameter equations will be exactly the same as those in Section 3.6. Of course, there will be M times as many equations as there were for a single HMM.

Consider first the case of a general coupled HMM as defined in Section 6.2. Each of M transition probability pdfs $Pr(s_t^{(n)}|s_{t-1}^{(0)}, \ldots, s_{t-1}^{(M-1)}), n = 0, \ldots, M-1$ can be viewed as a table with cells indexed by $M+1$ dimensional indices (corresponding to instances of $s_t^{(n)}, s_{t-1}^{(0)}, \ldots, s_{t-1}^{(M-1)}$). There are $N(n)N(0) \cdot \ldots \cdot N(M-1)$ entries in each table. Let us denote the table entry indexed by $(i_0, i_1, \ldots, i_M)$, $i_n \in \{0, \ldots, N(n)-1\}$ as

$$Pr(s_t^{(n)} = e_{i_0}, s_{t-1}^{(0)} = e_{i_1}, \ldots, s_{t-1}^{(M-1)} = e_{i_M}).$$

Thus, this is the probability that modality n is in state i_0 at time t (or, equivalently, $s_t^{(n)} = e_{i_0}$) given that the states of modalities 0 through $M-1$ at time $t-1$ are i_1 through i_M, respectively.

Application of the maximization step of the GEM algorithm leads to the following transition pdf update equation:

$$\begin{aligned} &Pr(s_t^{(n)} = e_{i_0}, s_{t-1}^{(0)} = e_{i_1}, \ldots, s_{t-1}^{(M-1)} = e_{i_M}) = \\ &\frac{\sum_{t=1}^{T-1} \left\langle s_t^{(n)} = e_{i_0}, s_{t-1}^{(0)} = e_{i_1}, \ldots, s_{t-1}^{(M-1)} = e_{i_M} \right\rangle}{\sum_{t=1}^{T-1} \sum_{i_n=0}^{N(n)} \left\langle s_t^{(n)} = e_{i_0}, s_{t-1}^{(0)} = e_{i_1}, \ldots, s_{t-1}^{(M-1)} = e_{i_M} \right\rangle}. \end{aligned} \tag{6.22}$$

Note that terms such as $\left\langle s_t^{(n)} = e_{i_0}, s_{t-1}^{(0)} = e_{i_1}, \ldots, s_{t-1}^{(M-1)} = e_{i_M} \right\rangle$ actually represent joint probabilities $Pr(s_t^{(n)} = e_{i_0}, s_{t-1}^{(0)} = e_{i_1}, \ldots, s_{t-1}^{(M-1)} = e_{i_M})$. Equation 6.22 is in fact a generalization of the transition pdf update equation of ordinary HMMs (see Section 3.6).

However, recall that all our coupled HMM topologies except for the naive one assume the factorization of state transition pdfs according to Equation 6.4. Now, the parameter update equation in 6.22 assumes a simpler form:

$$P^{(n,m)}(i,j) = \frac{\sum_{t=1}^{T-1} \left\langle s_t^{(n)}(i) s_{t-1}^{(m)}(j) \right\rangle \left\langle w_t^{(n)}(m) \right\rangle}{\sum_{t=1}^{T-1} \left\langle s_t^{(m)}(j) \right\rangle \left\langle w_t^{(n)}(m) \right\rangle}. \tag{6.23}$$

Recall that $P^{(n,m)}(i,j)$ is the (i,j)th entry of the (n,m)th state transition matrix, i.e., the probability that state j in modality m is followed by state i in modality n at the next time instance. The term $\left\langle s_t^{(n)}(i) \right\rangle$ denotes the ith component of vector $\left\langle s_t^{(n)} \right\rangle$, i.e., the probability that at time t the state of the nth modality is i. Note that the joint term $\left\langle s_t^{(n)}(i) s_{t-1}^{(m)}(j) \right\rangle$ can be decoupled as $\left\langle s_t^{(n)}(i) \right\rangle \left\langle s_{t-1}^{(m)}(j) \right\rangle$ when $n \neq m$. This is because the expectations are taken with respect to the factorized Q distribution where all modalities are mutually independent.

Besides the state transition pdfs, updates are also needed for parameters of the distributions associated with intermodal weights (in the cases when they are assumed to be stochastic). We first consider the case of adaptive but time-invariant weights (see Section 6.3.2.2). It is easy to show that, in this case, the parameter W has the following update:

$$W^{(n)}(m) = \left\langle w^{(n)}(m) \right\rangle \tag{6.24}$$

for $n, m = 0, \ldots, M-1$.

In the case of time-varying weight factors, the update equation for parameters of the intermodal weight HMM distributions follows the ordinary HMM parameter updates. Namely,

$$W^{(n)}(i,j) = \frac{\sum_{t=1}^{T-1} \left\langle w_t^{(n)} = i, w_{t-1}^{(n)} = j \right\rangle}{\sum_{t=1}^{T-1} \left\langle w_{t-1}^{(n)} = j \right\rangle}, \tag{6.25}$$

where $n, i, j = 0, \ldots, M-1$. As mentioned in Section 6.3.2.3, the initial weight distribution is fixed to

$$\rho_0^{(n)}(k) = \delta(n-k)$$

for $n, k = 0, \ldots, M-1$.

CHAPTER 7

ANALYSIS AND RECOGNITION OF HAND GESTURES USING DYNAMIC BAYESIAN NETWORKS

7.1 Introduction

Hand gestures are a means of nonverbal interaction among people. They range from the simple acts of using the hand to point at and move objects, to the more complex gestures that express our feelings and allow us to communicate ideas with others. To exploit the use of gestures in HCI, it is necessary to provide the means by which they can be interpreted by computers. The HCI interpretation of gestures requires that dynamic and/or static configurations of the human hand, arm, and even other parts of the human body be measurable by the machine. First attempts to solve this problem resulted in mechanical devices that directly measure hand and/or arm joint angles and spatial position. This group is best represented by the so-called *glove-based devices* [11, 12, 13, 14, 15]. Glove-based gestural interfaces require the user to wear a cumbersome device and generally carry a load of cables that connect the device to a computer. This hinders the ease and naturalness with which the user can interact with the computer-controlled environment.

Potentially, any awkwardness in using gloves and other devices can be overcome by video-based noncontact interaction techniques. These approaches use video cameras and computer vision techniques to interpret gestures, as depicted in Figure 7.1. The nonobstructiveness of the resulting vision-based interface is particularly relevant to HCI.

7.1.1 Definition of gestures

Outside the HCI framework, hand gestures cannot be easily defined. Webster's dictionary, for example, defines gestures as "the use of motions of the limbs or body as a means of expression; a movement usually of the body or limbs that expresses or emphasizes an idea, sentiment, or attitude." Psychological and social studies tend to narrow this broad definition and relate it to human expression and

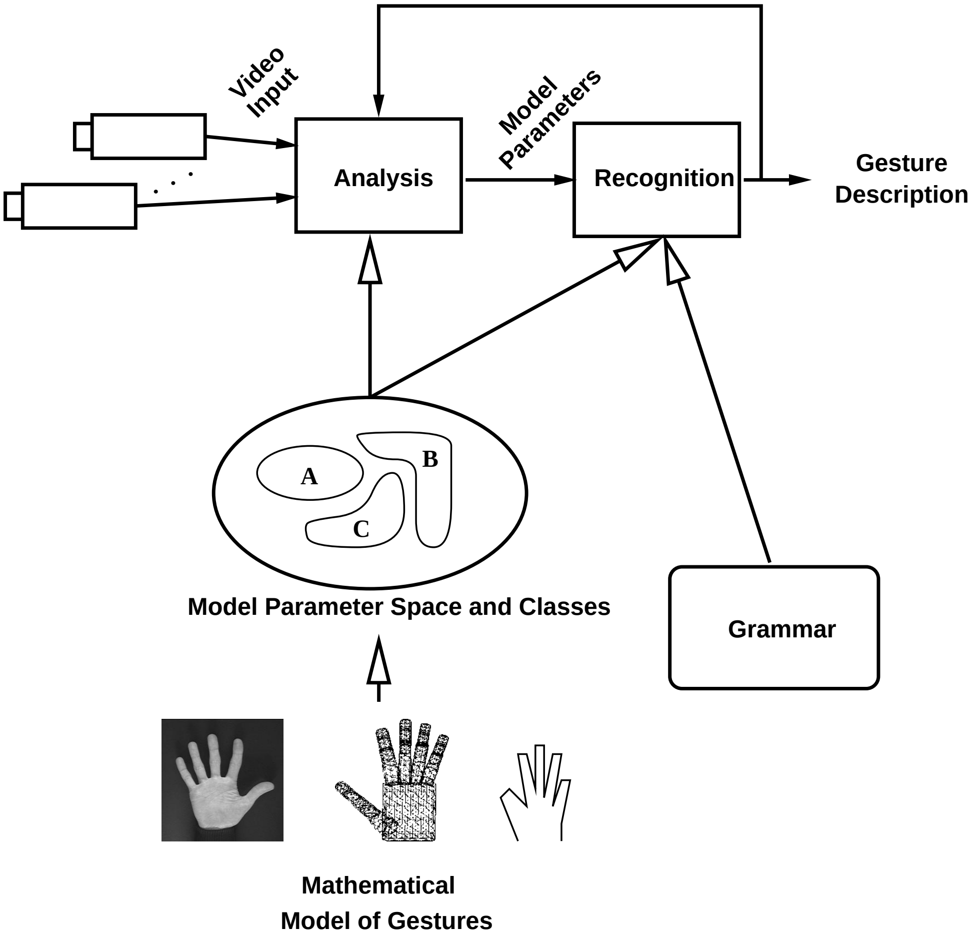


Figure 7.1 Vision-based gesture interpretation system. Visual images of gestures are acquired by one or more video cameras. They are processed in the analysis stage where the gesture model parameters are estimated. Using the estimated parameters and some higher-level knowledge, the observed gestures are inferred in the recognition stage.

social interaction [82]. However, in the domain of HCI the notion of gestures is somewhat different. In a computer-controlled environment, one wants to use the human hand both to mimic the natural use of the hand as a manipulator and to communicate to the machine (control of computer/machine functions through gestures).

Hand gestures are a means of communication, similar to spoken language. The production and perception of gestures can thus be described using a model commonly found in the field of spoken language recognition. An interpretation of this model, applied to gestures, is depicted in Figure 7.2. According to the model, gestures originate as a gesturer's mental concept, possibly in conjunction with other modalities such as speech. They are expressed through the motion of arms and hands. Observers perceive gestures as streams of visual images which they interpret using their knowledge of those gestures. Thus, the production and perception model of gestures has usually been summarized in the following

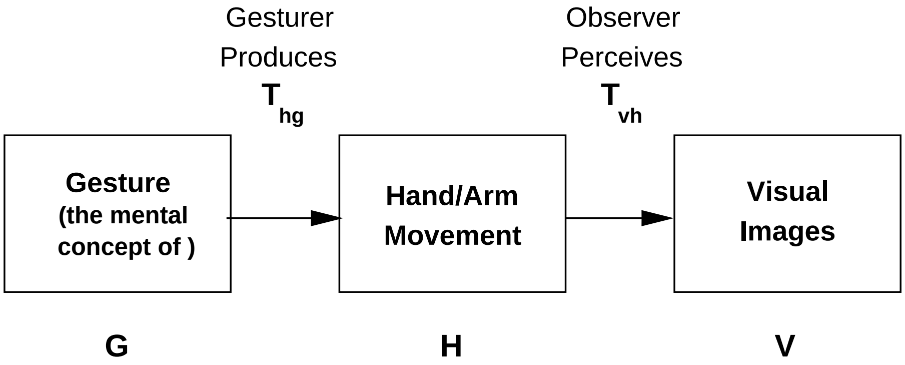


Figure 7.2 Production and perception of gestures. Hand gestures originate as a mental gestural concept G, are expressed (T_{hg}) through arm and hand motion H, and are perceived (T_{vh}) as visual images V.

form:

$$H = T_{hg}G, \tag{7.1}$$

$$V = T_{vh}H, \tag{7.2}$$

$$V = T_{vh}\left(T_{hg}G\right) = T_{vg}G. \tag{7.3}$$

Transformations $T_{\cdot}$ can be viewed as different models: T_{hg} is a model of hand or arm motion given gestural concept G, T_{vh} is a model of visual images given hand or arm motion H, and T_{vg} describes how visual images V are formed given some gesture G. The models are parametric, with the parameters belonging to their respective parameter spaces $\mathcal{M}_{\mathcal{T}_{\cdot}}$. In light of this notation, one can say that the aim of visual interpretation of hand gestures is to infer gestures G from their visual images V using a suitable gesture model T_{vg}, or

$$\hat{G} = T^{i}_{vg}V, \tag{7.4}$$

where T^{i}_{vg} denotes some ("inverse") mapping from space V to space G.

7.1.2 Gestural taxonomy

Several alternative taxonomies have been suggested in the literature that deal with psychological aspects of gestures. Kendon [82] distinguishes "autonomous gestures" (that occur independently of speech) from "gesticulation" (gestures that occur in association with speech). McNeill and Levy [79] recognize three groups of gestures: iconic and metaphoric gestures, and "beats." The taxonomy that seems most appropriate within the context of HCI was recently developed by Quek [83, 28]. A slightly modified version of the taxonomy is given in Figure 7.3. All hand/arm movements are first classified into two major classes: gestures and unintentional movements. Unintentional movements are those hand/arm movements that do not convey any meaningful information. Gestures themselves can have

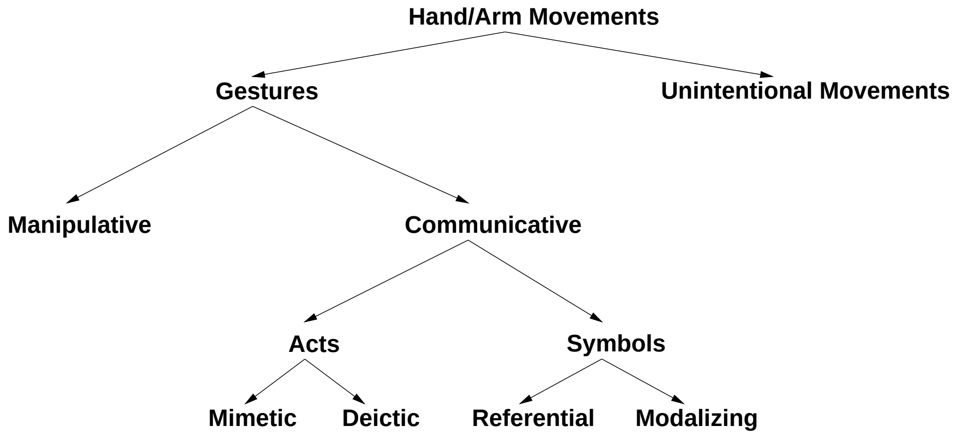


Figure 7.3 A taxonomy of hand gestures for HCI. Meaningful gestures are differentiated from unintentional movements. Gestures used for manipulation (examination) of objects are separated from the gestures which possess inherent communicational character.

two modalities: communicative and manipulative. Manipulative gestures are the ones used to act on objects in an environment (object movement, rotation, etc.). Communicative gestures, on the other hand, have an inherent communicational purpose. In a natural environment they are usually accompanied by speech. Communicative gestures can be either acts or symbols. Symbols are those gestures that have a linguistic role. They symbolize some referential action (for instance, circular motion of index finger may be a sign for a wheel) or are used as modalizers, often of speech ("Look at that wing!" and a modalizing gesture specifying that the wing is vibrating, for example). In the HCI context, symbols are, so far, one of the most commonly used gestures since they can often be represented by different static hand postures. Finally, acts are gestures that are directly related to the interpretation of the movement itself. Such movements are classified as either mimetic (which imitate some actions) or deictic (pointing acts).

7.1.3 Temporal modeling of gestures

Because human gestures are dynamic processes, it is important to consider their temporal characteristics. This may help in the temporal segmentation of gestures from other unintentional hand/arm movements. In terms of our general definition of hand gestures, this is equivalent to determining the so-called gesture interval. Surprisingly, psychological studies are fairly consistent about the temporal nature of hand gestures. Kendon [82] calls this interval a "gesture phrase." It has been established that three phases make a gesture: preparation, nucleus (peak or stroke [79]), and retraction. Preparation phase consists of a preparatory movement that sets the hand in motion from some resting position. The nucleus of a gesture has some "definite form and enhanced dynamic qualities" [82]. Finally, the hand either returns to the resting position or repositions for the new gesture phase. An exception to this rule is the so called "beats" (gestures related to the rhythmic structure of the speech).

The above discussion can guide one in the process of temporal discrimination of gestures. The three temporal phases are distinguishable through the general hand/arm motion: “preparation” and “retraction” are characterized by rapid change in position of the hand, while the “stroke,” in general, exhibits relatively slower but sometimes more periodic hand motion.

7.1.4 Spatial modeling of gestures

Gestures are observed as hand and arm movements, actions in 3D space. Hence, the description of gestures also involves the characterization of their spatial properties. In an HCI domain this characterization has so far been mainly influenced by the kind of application for which the gestural interface is intended. For example, some applications require simple models (like static image templates of the human hand in TV set control in [84]), while some others require more sophisticated ones (3D hand models used in [85, 86], for instance).

If one considers the gesture production and perception model suggested in Section 7.1.1, two possible approaches to gesture modeling may become obvious. One approach may be to try to infer gestures directly from the visual images observed, as stated by Equation 7.4. This approach has often been used to model gestures, and is usually denoted as *appearance-based* modeling. Such models include deformable templates [87, 88, 89, 90, 91], point distribution models [92], whole visual images of hands, arms, or body [93, 94, 95, 96], and silhouettes and contours [97, 98, 99, 100]. Another approach may result if the intermediate tool for gesture production is considered: the human hand and arm. In this case, a two-step modeling process may be followed:

$$\hat{H} = T^i_{vh} V \tag{7.5}$$

$$\hat{G} = T^i_{hg} \hat{H}. \tag{7.6}$$

In other words, one can first infer the motion and/or posture of the hand and arm $\hat{H}$ from their visual images V using some mapping T^i_{vh}. Following that, the inference of gestures $\hat{G}$ from the motion and posture model states $\hat{H}$ is achieved with a different mapping T^i_{hg}. Models which follow this approach are known as articulated models [86, 101, 102, 103, 104, 105, 106, 107, 108, 109].

7.1.5 Gesture analysis

The goal of the gesture analysis phase is to estimate the states of the gesture model H using measurements from the video images V of a human operator engaged in HCI. However, direct mapping of the images of hand or arm actions to the states of the hand or arm model using some mapping T^i_{vh} would usually be extremely complex. For instance, one would have to infer what the hand driving torque is, given a sequence of images of the hand in motion. In practice it is more convenient to introduce an intermediate step to this process. This is depicted in Figure 7.4. Namely,

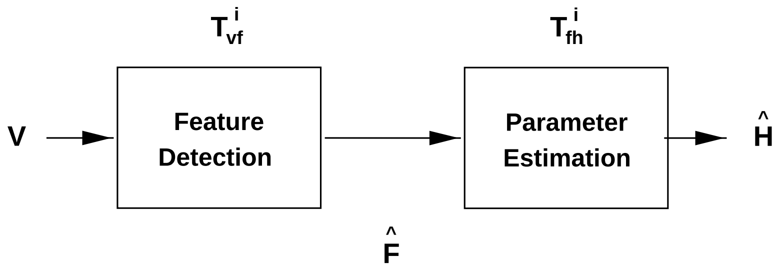


Figure 7.4 Analysis of hand gestures. In the analysis stage, features F are extracted from visual images V. Model states $\hat{P}$ are estimated and possibly predicted.

$$\hat{F} = T_{vf}^{i} V \tag{7.7}$$

$$\hat{H} = T_{fh}^{i} \hat{F}. \tag{7.8}$$

This means that two generally sequential tasks are involved in the analysis (see Figure 7.4). The first task involves "detection" or extraction of relevant image features from the raw image or image sequence. The second task uses these image features for computation of the model states and, consequently, the driving input.

7.1.5.1 Feature detection

The feature detection stage is concerned with detection of features which are used for estimating of the states of a chosen gestural model. In the detection process it is first necessary to localize the gesturer. Once the gesturer is localized, the desired set of features can be detected.

Two types of cues are most frequently used in the localization process: *color cues* and *motion cues.* Color cues are applicable because of the characteristic color signature of the human skin. The color signature is usually more distinctive and less sensitive to illumination changes in the hue-saturation space than in the standard (camera capture) RGB color space. Most of the color segmentation techniques rely on histogram matching [110] or employ a simple look-up table approach [111, 112] based on the training data for the skin and possibly its surrounding areas. However, the skin color matching schemes can often be unreliable in changing illumination or background conditions. Hence, many gesture recognition applications resort to the use of uniquely colored gloves or markers on hands/fingers [109, 113, 114, 115, 116]. The use of colored gloves makes it possible to localize the hand efficiently and even in real-time, but imposes an obvious restriction on the user and the interface setup.

Motion cues are also commonly applied to hand/arm localization, and they are used in conjunction with certain assumptions about the gesturer. For example, in the HCI context, it is usually the case that only one person gestures at any given time. Moreover, the gesturer is usually stationary with respect to the (also stationary) background. Hence, the main component of motion in the visual image is usually the motion of the arm/hand of the gesturer and can thus be used to localize her/him.

7.1.5.2 Model state estimation

Computation of the model states is the last stage of gesture analysis phase. In gesture recognition systems, this is followed by the recognition stage, as shown in Figure 7.4. For hand or arm tracking systems, however, the state computation stage usually produces the final output. The type of computation used depends on both the model type and the features that were selected.

In the case of 3D hand models, two sets of states are used—angular (joint angles) and linear (phalangae lengths and palm dimensions). The estimation of these kinematic states from the detected features is a complex and cumbersome task. Under certain assumptions the problem of finding the hand joint angles can be reduced to an inverse kinematics problem. Inverse kinematic problems are in general ill-posed, allow for multiple solutions, and are computationally expensive. Approximate solutions are therefore found in most case [85, 86, 110]. Once the hand model states are initially estimated, the state estimates can be updated using some kind of prediction/smoothing scheme. A commonly used scheme is Kalman filtering and prediction. Three major drawbacks are associated with the 3D hand model state estimation approach: the obvious computational complexity of the inverse kinematics, occlusions of features (the fingertips), and changes in scale that are difficult to adapt to. Finally, it should be pointed out that knowledge of the exact hand posture states seem unnecessary for the recognition of communicative gestures [83], although the exact role of 3D hand parameters in gesture recognition is not clear.

In the case of appearance-based models of the hand or arm the estimation of the states of such models usually coincides with the estimation of some compact description of the image or image sequence. Appearance models based on the visual images per se are often used to describe gestural actions: key frames [96], eigen-images [117], motion history images [93], etc. Deformable 2D template-based models are also often employed as the spatial models of hand and arm contours or even the whole human body [88, 89, 90, 92]. Finally, a wide class of appearance models uses silhouettes or gray-scale images of the hands. In such cases the model states attempt to capture a description of the shape of the hand while being relatively simple [26, 97, 98, 100, 118]. Like the other state estimation tasks, the reported estimation of motion states are usually based on simple Newtonian dynamics models and Kalman-based predictors.

7.1.6 Gesture recognition

Gesture recognition is the phase in which the data analyzed from the visual images of gestures is recognized as a specific gesture. This is figuratively depicted in Figure 7.5. In accordance to my previous notation, this can be formally written as

$$\hat{G} = T_{hg}^{i} \hat{H}. \tag{7.9}$$

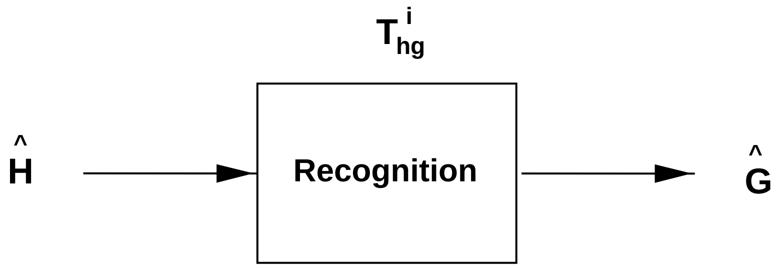


Figure 7.5 Recognition of hand gestures. States of the physical model H are used to infer the underlying gestural concept G.

Namely, the estimates of the physical model's states are used in some way to infer what the gestural concept is. For example, if one has the measurements of the hand and arm joint angles in time, one may be able to figure out what the gesture was that produced those measurements.

In the cases of certain gestural types, such as particular iconic gestures or certain symbols of American sign language (ASL), it may be sufficient to focus on static postures of the hand [119, 120]. However, in general, gestures are *spatio-temporal actions.* Because gestural actions possess this temporal context, one needs to resort to recognition of temporal patterns. Fortunately, a wide variety of techniques is available from the field of time series analysis. The main requirement for any pattern recognition technique used in gestural classification is that it be time-instance invariant and time-scale invariant. For example, a clapping gesture should be recognized as such whether it is performed slowly or quickly, now or in ten minutes. Again, numerous temporal pattern recognition techniques deal with such problems, and perhaps the most prominent of these emerges from the field of automatic speech recognition (ASR). Because both speech and gestures are means of natural human communication, an analogy drawn between them and computational tools developed for ASR is frequently used in gesture recognition. In particular, hidden Markov models and their different flavors have almost been the sole successful carriers of the gesture recognition task [26, 98, 100, 121].

7.2 General Dynamic Bayesian Network Gesture Model

Let me once again consider the model of gestural actions discussed in Section 7.1.1. Recall that in this model a gestural concept drives, through some physical process, the physical system of the human arm and hand. An observer perceives gestures by somehow measuring (visually or mechanically) the motion and posture of the gesturer's hands. This model can be graphically presented in the form of a system block diagram of Figure 7.6. In this diagram the state of some gestural concept is denoted by c. The state of the physical system is similarly denoted by x. Depending on the model of the physical system, this state could, for instance, represent values of the hand joint angles or maybe linear hand velocities. The influence of concept on the physical system is represented as some driving input u. Again,

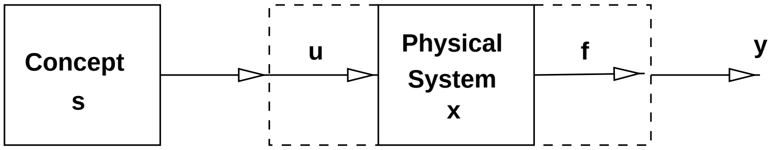


Figure 7.6 Block diagram of a gesture production system. Gestural concept c drives a physical system through an input u. System states x evolve according to some dynamics and are partially observed through intermediate features f as observations y.

depending on what model of the system one has in mind this could be, for example, the joint torque or linear force that causes hands to move. Finally, from an observer's point of view the hand motion gets perceived as an observation y. If one visually observes the hand motion through a camera CCD, y can then denote the gray or color levels of pixels in an image. What one really wants to know, however, is where the hand and arm are in 3D space. Hence, one represents that information as features f and attempts to infer them from observations y. Alternatively, one may be able to measure directly the spatial position of the moving hand (using a magnetic tracker device, for example). In that case, the observation y and the feature f may jointly represent a single noisy measurement of the hand position.

This representation of the gestural model is quite generic. At this point, however, we make the crucial assumption. We propose that

Any gestural action can be modeled as a dynamic Bayesian network.

In particular, we introduce a DBN topology depicted in Figure 7.7. This means that any gestural action of duration T observed through a set of observations $\mathcal{Y} = \{y_0, \ldots, y_{T-1}\}$ is generated by a pdf defined over the space of gestural concept states $\mathcal{C} = \{c_0, \ldots, c_{T-1}\}$ and physical (hand and arm) system states $\mathcal{X} = \{x_0, \ldots, x_{T-1}\}$. Formally,

$$Pr(\mathcal{C}, \mathcal{X}, \mathcal{Y}) = Pr(c_0) \prod_{t=1}^{T-1} Pr(c_t|c_{t-1}) Pr(x_t|x_{t-1}, c_t) \prod_{t=0}^{T-1} Pr(y_t|x_t). \tag{7.10}$$

In this formulation we have eliminated, without loss of generality and for sake of simplicity, the input variables u and the feature variables f.

Again, without loss of generality we assume that

- concept states c are discrete valued,
- states x of the physical system are continuous valued, and
- observations y are continuous or discrete valued.

Why do we impose those restrictions? Clearly, the common notion of a concept implies a set of discrete symbols. For instance, each gesture has three distinct gestural phases (see Section 7.1.3). Physical

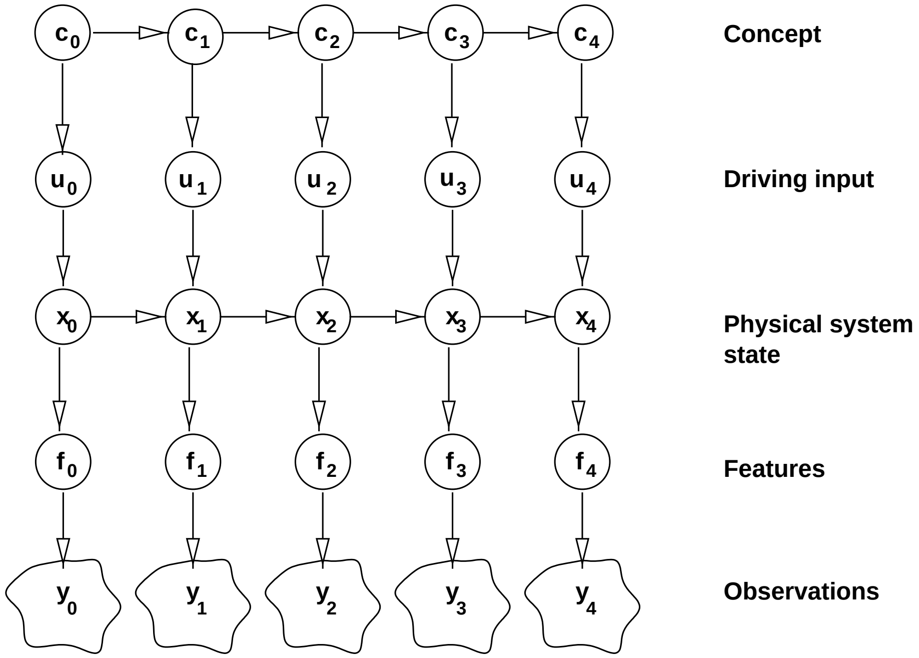


Figure 7.7 Dynamic Bayesian network representation of a gestural action.

systems, on the other hand, are "allowed" to take on a continuum of values. The hand, for example, can be positioned in any point of a portion of the 3D space, and hand velocity can pretty much take on any value within a certain range. Similarly, pixel intensity can in principle assume any value, even though it is usually quantized to an 8-bit range.

However, more important than the above restrictions are the ones imposed on the structure of state transitions in the model. We therefore assume that

- the gestural concept of every gestural action can be modeled as a Markov chain, and
- the dynamics of hand and arm can be modeled as a linear (or linearized) dynamic system (LDS).

The assumption regarding concept dynamics is plausible and in line with the usual concept models, but the linear dynamics assumption for the hand/arm motion model is potentially a problem. Section 7.1.4 surveyed a number of common, as we called them, spatial models of hand and arm motion. The most physically correct is the articulated dynamical model. However, this model is extremely complex and nonlinear. Moreover, it is questionable whether this model is at all useful for gesture recognition [83]. Therefore, let us say tentatively that the LDS assumption for the physical hand model is feasible.

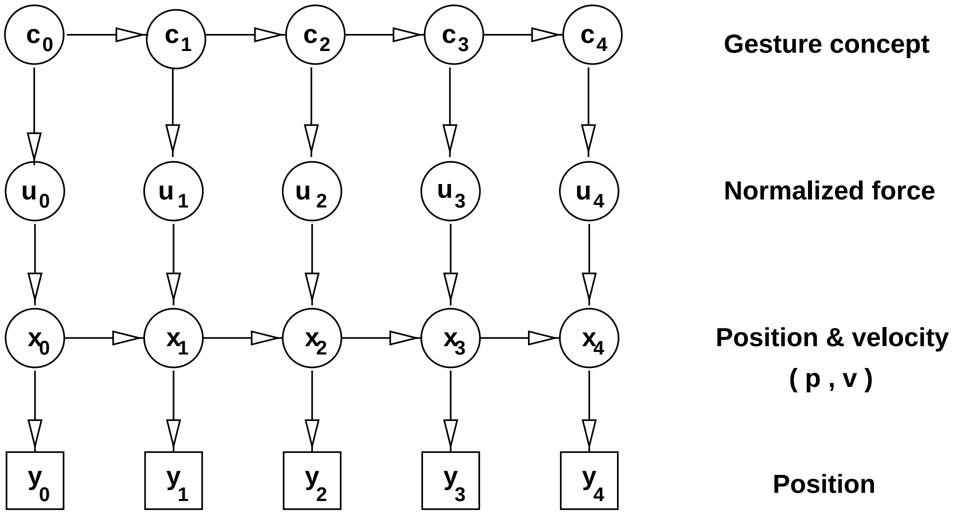


Figure 7.8 Dynamic Bayesian network representation of a directly observable gestural action, such as the ones measured by the computer mouse.

7.3 Directly Observed Hand Gestures

The notion of directly observed hand motion assumes that, as mentioned before, the observations carry noisy measurements of the spatial position of the human hand. Such cases occur, for instance, when the hand position is tracked with magnetic or ultra-sonic trackers or with a simple computer mouse. In those case there is, in general, no need to consider some intermediate features f as part of a gestural action model. Hence, the general hand gesture model from Section 7.2 reduces to the dependency graph of Figure 7.8. Recall that we encountered this dependency topology earlier. In fact, the above network, together with the Markov chain concept and LDS physical system state assumptions, was denoted the *mixed-state HMM* in Chapter 5. Hence, to analyze and recognize directly observed hand gestures one can immediately apply the inference, decoding, and learning techniques of mixed-state HMMs in Chapter 5. In the sections to follow we discuss in more detail peculiarities of mixed-state HMM techniques applied to hand gesture analysis and recognition.

7.3.1 Dynamics of hand motion

In my modeling of hand motion we impose simplified Newtonian dynamics on every hand- or arm-related measurement. Namely,

> The hand motion is modeled as the relative kinematic motion of a point-mass particle with *piecewise constant acceleration*, and with respect to some suitable stationary origin.

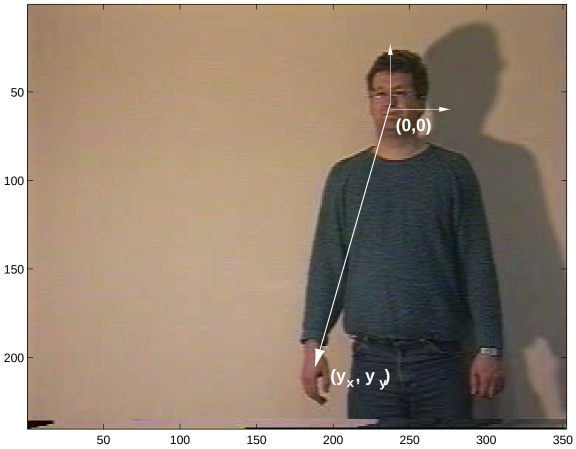


Figure 7.9 Hand position is measured relative to some stationary origin. In this case hand position (y_x, y_y) is defined with respect to head.

Note the fact that the motion is relative. For instance, in the case of global arm motion the head (or one shoulder) can be chosen as the stationary origins. In the case of computer mouse strokes, on the other hand, the motion is inherently relative to some workspace.

From the above assumptions follow the motion state equations

$$\dot{x}(t) = v(t), \text{ and} \tag{7.11}$$

$$\dot{v}(t) = u(t) + n_v(t), \tag{7.12}$$

where $x(t) = [x_x(t)x_y(t)]'$ is the relative position of the hand with respect to the head, $v(t)$ is a linear hand velocity, and $u(t)$ a piece-wise constant input (see Figure 7.9). Roughly, one can think of $u(t)$ as a mass-normalized force excerpted on the hand. Of course, the piece-wise constancy comes in place because we believe that, as stated in Section 7.2, the motion is driven by a concept which is discrete-valued. The term $n_v(t)$ represents a noise process that models uncertainty in input $u(t)$. Assume that the noise process is i.i.d. Gaussian with zero mean and constant variance q:

$$n_v(t) \sim \mathcal{N}(0, q).$$

Furthermore, assume that only position of the hand is observable:

$$y(t) = x(t) + n_x(t),$$

where n_x denotes the observation noise and $y(t)$ is the vector of observed hand position $y(t) = [y_x(t)y_y(t)]'$. Again, assume that this noise process is i.i.d. Gaussian, with zero mean and variance r:

$$n_x(t) \sim \mathcal{N}(0, r).$$

Discretization in time is obtained using zero-order hold approximation when uniformly sampling with period T_s (see [76], for example). This yields the set of state-space equations

$$\begin{aligned} x_{t+1} &= Ax_t + Bu_t + n_t \\ y_t &= Cx_t + w_t, \end{aligned} \tag{7.13}$$

where x_t is now the vector of positions and velocities sampled at time $t \cdot T_s$:

$$x_t = \begin{bmatrix} x(t \cdot T_s) \\ v(t \cdot T_s) \end{bmatrix}$$

u_t and y_t are sampled versions of $u(t)$ and $y(t)$, respectively, n_t is the state noise term obtained by sampling the piecewise constant process noise $n_v(t)$, and w_t is sampled from the noise process $n_x(t)$. It is easy to show that in the case of uncorrelated spatial noise processes[1] the state matrices for each spatial coordinate are

$$A = \begin{bmatrix} 1 & T_s \\ 0 & 1 \end{bmatrix} = A(T_s) \tag{7.14}$$

$$B = \begin{bmatrix} \frac{1}{2}T_s^2 \\ T_s \end{bmatrix} = B(T_s) \tag{7.15}$$

$$C = \begin{bmatrix} 1 & 0 \end{bmatrix} \tag{7.16}$$

$$Q = \begin{bmatrix} \frac{1}{3}T_s^3 & \frac{1}{2}T_s^2 \\ \frac{1}{2}T_s^2 & T_s \end{bmatrix} q = Q(T_s)\, q \tag{7.17}$$

$$R = r, \tag{7.18}$$

where the last two terms denote variances of the sampled noise processes

$$v_t \sim \mathcal{N}(0, Q),$$

and

$$w_t \sim \mathcal{N}(0, R).$$

Also recall from Chapter 5 that the piecewise constant input u_t is modeled as $u_t = Dc_t$, with a matrix of N_c input levels $D = [d_0 \dots d_{N_c-1}]$ and Markov chain dynamics of concept c_t.

7.3.2 Inference

Exact inference of the hidden concept and system states in the proposed DBN gesture model is in general intractable (see Chapter 5). In Chapter 5 we discussed several inference techniques that may be used to achieve tractable yet approximate inference. All those techniques, such as variational inference, are readily and directly applicable to this gesture model.

[1] "Uncorrelated spatial noise process" implies that the noise variance q is diagonal.

7.3.3 Parameter learning

Parameter learning of the directly observed hand gesture model follows directly from the generalized EM learning framework of mixed-state HMMs in Chapter 5. However, as a consequence of the Newtonian dynamics model, certain differences emerge in this specific case.

One exception is clearly that the linear system transition matrix A and observation matrix C are known and need not be estimated. More important differences, however, surface in the update equations for the LDS state covariance Q and input levels D. It is easy to show that the maximization step of the GEM algorithm yields the following parameter update equation for each spatial coordinate:

$$\begin{aligned} D &= \left(B'Q(T)^{-1}B\right)^{-1} B'Q(T)^{-1} \left(\sum_{t=1}^{T-1} \langle (x_t - Ax_{t-1}) c_t{}' \rangle \right) \left(\sum_{t=1}^{T-1} \langle c_t c_t{}' \rangle \right)^{-1} && (7.19) \\ q &= \frac{1}{2(T-1)} \mathrm{tr} \left\{ Q(T)^{-1} \sum_{t=1}^{T-1} \langle (x_t - Ax_{t-1})(x_t - Ax_{t-1})' \rangle - BD \langle c_t (x_t - Ax_{t-1})' \rangle \right\}. && (7.20) \end{aligned}$$

Seemingly, the update equation for the input levels D shows dependency on the state noise variance Q. However, in the case of noise uncorrelated across the spatial coordinates, D becomes dependent only on the fixed value $Q(T)$ and not q itself, as Equation 7.19 clearly shows. Update equations for all other parameters, including the observation noise variance r and concept transition probability distribution P remain as given in Section 5.5.

7.3.4 Initialization

Approximate inference as well as learning of model parameters are iterative schemes. Namely, they involve recursive minimization of the cost function that is only guaranteed not to increase the cost (see Section 2.3). In order to attain "best" estimates of hidden variable statistics and model parameters one needs a "good" initial estimate of those desired quantities.

Variational inference of a model's sufficient statistics relies on an initial estimate of either the concept state $\langle c_t \rangle$ or the LDS state $\langle x_t \rangle$ (see Chapter 5). Given that Newtonian dynamics are imposed on the motion of the hand, a good initial estimate of the LDS state can be obtained by modeling the concept-driven LDS as a one order higher noise driven system with adaptive noise variance [76]. Recall the LDS state formulation of Equation 7.13. Assuming that input u_t is constant and contains uncorrelated additive white noise $n_{u,t}$ we can write

$$\begin{aligned} \begin{bmatrix} x_{t+1} \\ u_{t+1} \end{bmatrix} &= \begin{bmatrix} A & B \\ 0 & I \end{bmatrix} \begin{bmatrix} x_t \\ u_t \end{bmatrix} + \begin{bmatrix} n_t \\ n_{u,t} \end{bmatrix} \\ y_t &= \begin{bmatrix} C & 0 \end{bmatrix} \begin{bmatrix} x_t \\ u_t \end{bmatrix} + w_t. \end{aligned}$$

The input noise variance $var(n_{u,t})$ is (at least initially) unknown. One therefore selects some initial variance value based on general knowledge of the hand's maneuverability. Given a set of observations $\mathcal{Y}$ one can apply Kalman filtering using the extended LDS formulation. At each step of Kalman filtering, however, one needs to adaptively adjust the unknown input noise variance so that the filter "stays on track." The adjustment criterion is usually based on the requirement that the normalized innovations squared remain $((y_t - C\langle x_t \rangle)' \operatorname{var}(x_t)^{-1} (y_t - C\langle x_t \rangle))$ within certain bounds [76]. Hence, at each forward Kalman propagation step the noise covariance is adjusted until the criterion is met. As the result of this procedure one obtains estimates of $\langle x_t \rangle$ and $\langle u_t \rangle$ that can be used to initialize variational inference recursions (see Section 5.3.2).

Initialization of the model's parameters is based on the initial estimates of the sufficient statistics $\langle x_t \rangle$ and $\langle c_t \rangle$. An estimate of $\langle x_t \rangle$ can be obtained using the outlined procedure. On the other hand, an initial estimate of $\langle c_t \rangle$ is usually designed based on the general concept model. For instance, assume that the concept model is chosen to be "left-to-right" [5]. Assume furthermore that the observed data is of duration T. Then $\langle c_t \rangle$ is initially assumed to be such that each concept state "covers" a nonoverlapping temporal segment of duration T/N_c, where N_c is the total number of concept states. One can then immediately apply the update equations (7.20 and 7.19) to obtain an initial set of parameter estimates.

7.3.5 Experiment: Analysis and recognition of hand gestures acquired by the computer mouse

The computer mouse has long been used as a means of simple human–computer interaction. Its basic point-and-click function has become an integral part of any graphical user interface (GUI). However, many applications could benefit from more than just a simple click or pointing. For instance, in a text editor application one may use one mouse motion to indicate a word deletion and another one to specify that a word needs to be highlighted. While interacting with a street map displayed on a computer screen it may be useful to specify the orientation of the map by drawing an arrow symbol in the desired direction. Finally, one can use a mouse or another pointing device (such as stylus) to write letters or symbols that are to be recognized by the machine. Applications like this have also been around for several years but are currently gaining large popularity due to advent of personal digital assistants (PDAs) with handwriting recognition capabilities. To achieve on-line written symbol recognition a plethora of simple and complex pattern recognition techniques has been employed ranging from neural networks [122] to hidden Markov models [123].

To demonstrate the feasibility of DBN-based gesture recognition we next consider a simple case of computer mouse–acquired hand movements. Our experiment considered four classes of symbols produced by the human hand moving a computer mouse: arrow, erase, circle, and wiggle symbols. Examples of

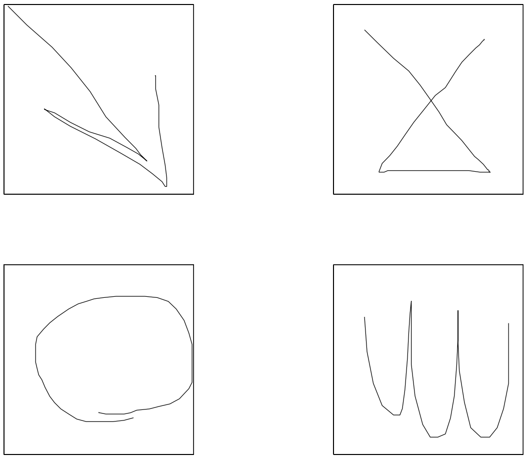


Figure 7.10 Examples of four symbols produced by computer mouse motions (left-to-right, top-to-bottom): “arrow,” “erase,” “circle,” and “wiggle.”

symbol classes are shown in Figure 7.10. The task in question was to model each of the four symbols with a combination of LDS and HMM. The LDS part, as outlined before, models dynamics of the mouse motion. The HMM, on the other hand, models the driving force (concept) that causes the motion. In the previous sections we discussed a complete, coupled model of LDS and HMM employed for this purpose, namely a mixed-state HMM. The model can be used to jointly infer the states of the driving concept c_t and the states of the LDS x_t using mixed-state HMM variational inference (see Chapter 5). In this experiment we contrasted this coupled model to two decoupled models:

- Decoupled adapted LDS and HMM. Namely, the LDS is adapted to “best” model the dynamics of the mouse motion of each symbol when the driving force u_t is assumed to be quasi-constant with additive white noise $u_t = u_{t-1} + n_{u,t}$ (see Section 7.3.4). The HMM is consequently employed to model the quasi-constant driving force u_t inferred by the LDS.

- Decoupled fixed LDS and HMM. In this case, the LDS is assumed to be fixed for all four symbols. In particular, we estimate the driving force using the numerical gradient approximation $u_t = grad(grad(x_t))$, where $grad(x_t) = \frac{x_{t+1}-x_{t-1}}{2 \cdot T_s}$. Again, an HMM is used to model the estimated driving force.

All three model classes are depicted in Figure 7.11.

For each of the three models we assumed the same concept state spaces. The number of concept states was determined to be related to the number of strokes necessary to produce each symbol. Thus, the concept model of the arrow symbol had eight states (two times four strokes), erase had six states, circle four states, and wiggle six. Furthermore, each concept state transition was limited to left-to-right transition: from current state the concept could only transition back to itself or to one other not-yet-

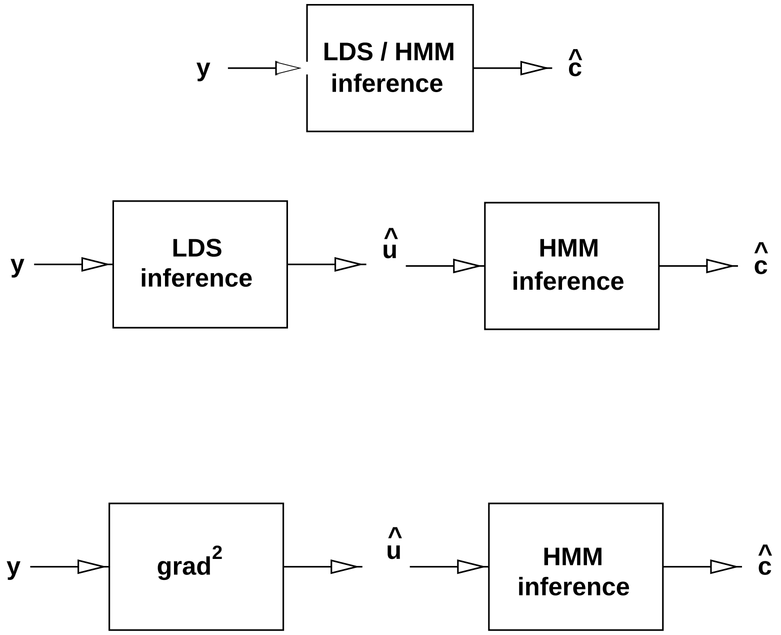


Figure 7.11 Three ways of modeling mouse-acquired symbols. From top to bottom: completely coupled LDS and HMM (mixed-state HMM), decoupled adapted LDS and HMM, and decoupled fixed LDS and HMM.

visited state. This implies that the concept probability transition matrix had nonzero elements only on the main diagonal and the first subdiagonal. In the two decoupled symbol models we chose to model the observations u_t of the concept models as variable mean Gaussian processes with identical variances at every concept state.[2]

The data set consisted of 136 examples of each symbol (thus, a total of 4×136 examples). Symbols were acquired from normalized[3] mouse movements sampled at 100-ms intervals. To test the models' performance we used the rotation error counting method [124]. Rotation error counting is a cross-validation training/testing procedure, a trade-off between the leave-one-out (LOO) method and the holdout error counting. Let the complete data set consist of N examples. The data set is partitioned into K disjoint subsets of cardinality N/K. Each model is then trained on $K - 1$ subsets and tested on the remaining one data subset. The total error count for every model is formed as the average misclassification frequency over the N/K test sessions. In our experiments we chose $1/4$ partition ($K = 4$) of each symbol's data set. Therefore, for every symbol in each one of the four training sessions a symbol model was trained on 3×36 data samples and tested on 36 positive and 3×136 negative examples. For each test sample and each symbol model the likelihood of the sample was appropriately obtained. For instance, in the case of mixed-state HMM modeled symbols, variational inference (see Chapter 5) with

[2] Even though it is a usual practice to allow the variance to vary from concept state to concept state, for sake of compatibility with the fixed variance mixed-state HMM we decided to keep the other HMMs' observation variances fixed.

[3] Symbol were scaled to $[0, 1] \times [0, 1]$ unit area and directionally aligned.

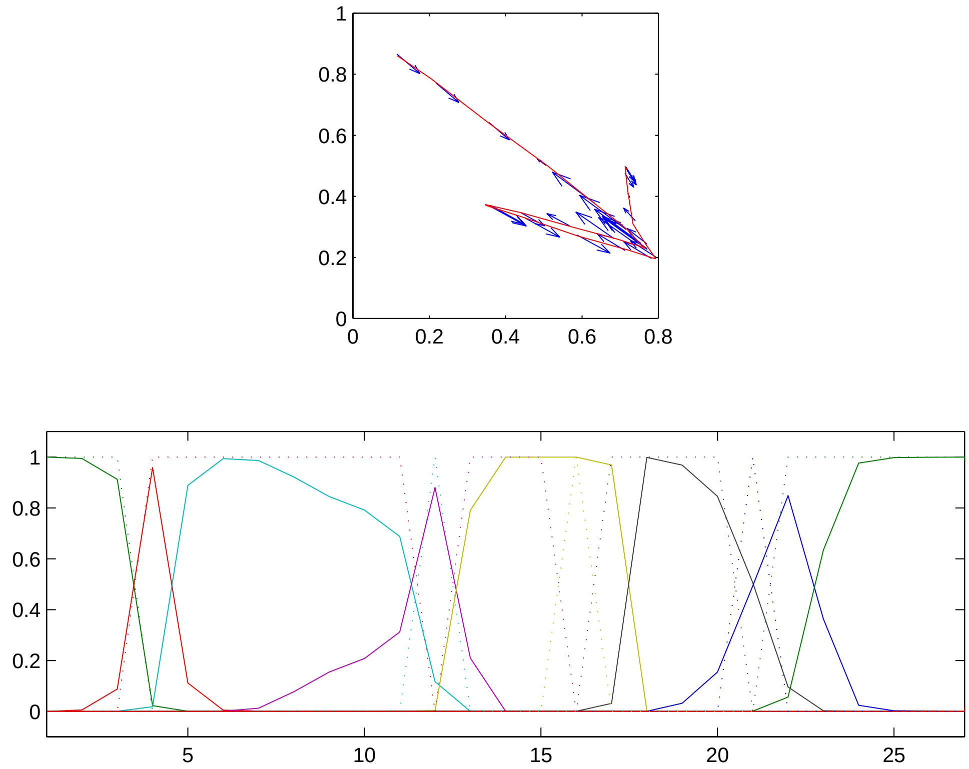


Figure 7.12 Estimates of concept states for "arrow" symbol. Top graph depicts the symbol and the estimated driving force. Bottom graph shows estimates of concept states obtained using variational inference (solid line) and Viterbi decoding (dashed line).

relative error threshold of .001 was used to estimate the lower bound on likelihood (upper bound on cost). One example of mixed-state HMM-based decoding of "arrow" symbol is shown in Figure 7.12. For a gradient-based LDS/HMM model, likelihood was obtained using standard HMM inference of Chapter 3.

Test error rates are summarized in Table 7.1 and shown in Figure 7.13. Each error estimate was obtained as a biased MAP estimate of a Bernoulli probability in a binomial distribution of error counts [125]. As is obvious from Table 7.1 and Figure 7.13, none of the three models (mixed-state HMM, decoupled adapted LDS/HMM, and decoupled fixed LDS/HMM) performs significantly better than the other two.

A second set of experiments was aimed at testing the model's classification performance under additive white noise corruption. Each example from the data set was corrupted by i.i.d. zero-mean Gaussian noise with standard deviation of 0.01. Examples of noisy symbols are shown in Figure 7.14. Previously trained models of the four symbols were now tested on the noisy data, in the same fashion as in the noise-free case. Classification results are summarized in Table 7.2 and Figure 7.15. Table 7.2 and Figure 7.15 in this case indicate that, with 95 percent confidence ($p = .05$), completely coupled mixed-state HMM models had significantly better performance than both fixed and adapted decoupled LDS/HMM

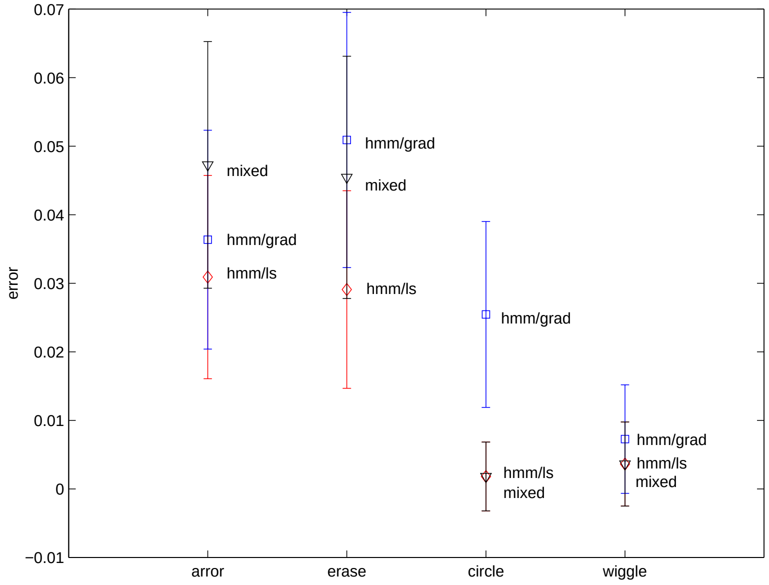


Figure 7.13 Classification error estimates of four mouse-acquired symbols. Shown are 95 percent confidence intervals for error counts. All three models (mixed, LDS/HMM, and grad/HMM) performed equally well in all cases but one. ("circle"). (▽ - mixed-state HMM, □ - fixed LDS/HMM, ◇ - adapted LDS/HMM.)

Table 7.1 Error estimates [%] and error estimate variances ([%]) for mouse symbol classification.

Model	Arrow	Erase	Circle	Wiggle
mixed-state HMM	4.73 (0.92)	4.55 (0.90)	0.18 (0.26)	0.36 (0.31)
gradient fixed LDS/HMM	3.64 (0.81)	5.09 (0.95)	2.55 (0.69)	0.73 (0.40)
decoupled adapted LDS/HMM	3.09 (0.76)	2.91 (0.74)	0.18 (0.26)	0.36 (0.31)

Table 7.2 Error estimates [%] and error estimate variances ([%]) for noisy mouse symbol classification.

Model	Arrow	Erase	Circle	Wiggle
mixed-state HMM	4.36 (0.89)	4.36 (0.89)	0.18 (0.26)	0.18 (0.26)
gradient fixed LDS/HMM	9.45 (1.25)	14.55 (1.51)	14.73 (1.51)	8.18 (1.18)
decoupled adapted LDS/HMM	24.91 (1.84)	25.09 (1.85)	0.55 (0.36)	35.64 (2.04)

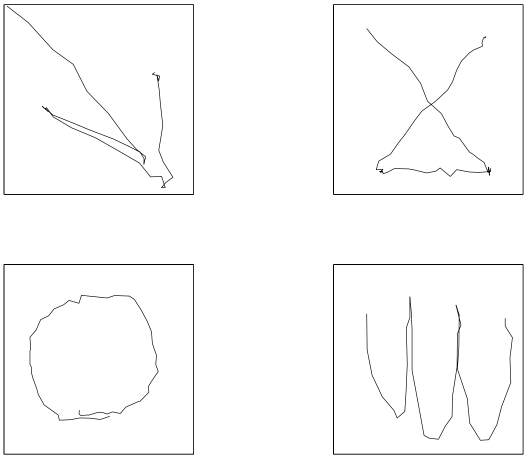


Figure 7.14 Samples of four mouse-acquired symbols corrupted by additive zero-mean white noise with standard deviation of 0.01.

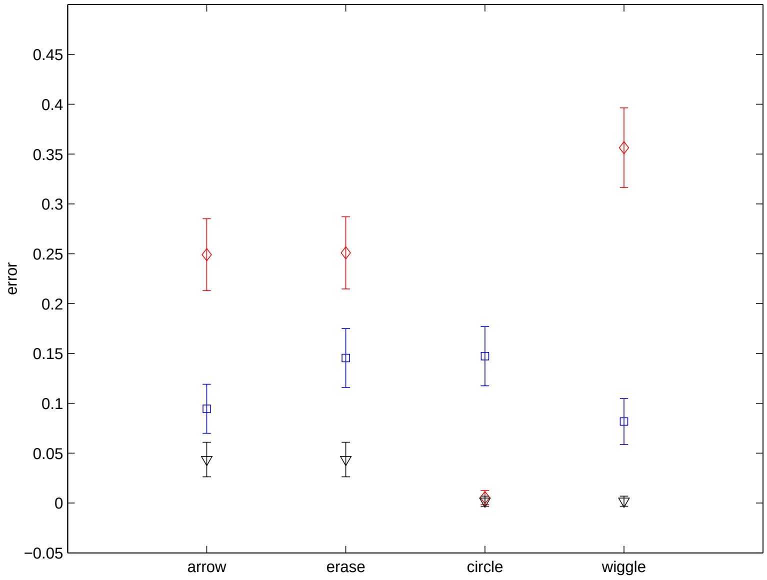


Figure 7.15 Classification error estimates of four noisy mouse symbols. Shown are 95 percent confidence intervals for error counts. Unlike the noise-free case, the coupled mixed-state HMM performed significantly better than the decoupled adapted and fixed LDS/HMM models. (▽ - mixed-state HMM, □ - fixed LDS/HMM, ◇ - adapted LDS/HMM.)

classifiers (with the exception of mixed-state and fixed LDS "circle" models). Of course, the trade-off is as always in increased computational complexity of the mixed-state models. Note, however, that on the average the iterative scheme of the mixed-state models required only about 5 to 10 iterations to converge (.001 relative change threshold).

7.4 Visually Observed Hand Gestures

In the previous section a model of directly measured hand motion was adopted. It is necessary now to consider how one can, from visual images of the human hand, infer the underlying gestural action. As mentioned in Section 7.1.5.1, most hand detection approaches rely on the unique appearance of the human skin in color space, as well as on the hand shape. We extend that approach and present it in the light of the DBN gesture model.

In the LDS gesture model of the previous section it was assumed that every observation corresponds, unambiguously, to a hand position at some given time. In other words, the observation could not have come from anything else but the hand position measurement. However, when a hand motion is observed visually, it is always the case that a number of pixels in any "hand" image does not belong to the hand. Such pixels could belong to a more or less stationary background or to other moving objects in the scene. Yet such measurements are present in our observation sets, and the simple LDS model must be modified to reflect this fact.

Recall from Chapter 4 the formulation of the mixture of DBN model. The model was formulated to deal with the case of multiple measurements with probabilistic model associations. The analogy between the mixture model and the one of visual hand tracking now becomes clear. Thus, we employ a *modified mixture of DBN* model for the visual tracking of the human hand.

Consider the case of the human hand moving according to the planar dynamics in Section 7.3. Suppose that this motion is observed by a stationary camera. Furthermore, assume that the background against which the hand is moving consists of stationary or almost stationary objects. A DBN model of a gestural action in such a scene can then be represented by the dependency diagram in Figure 7.16. The dependency graph in Figure 7.16 implies that every gestural action of duration T in a stationary background scene spatially sampled on M pixels can be thought of as generated by the following factorized pdf:

$$P = Pr(c_0) \prod_{t=1}^{T-1} Pr(c_t | c_{t-1}) Pr(x_t^{(f)} | x_{t-1}^{(f)}, c_t) \tag{7.21}$$

$$\times \prod_{t=0}^{T-1} Pr(x_t^{(b)}) \tag{7.22}$$

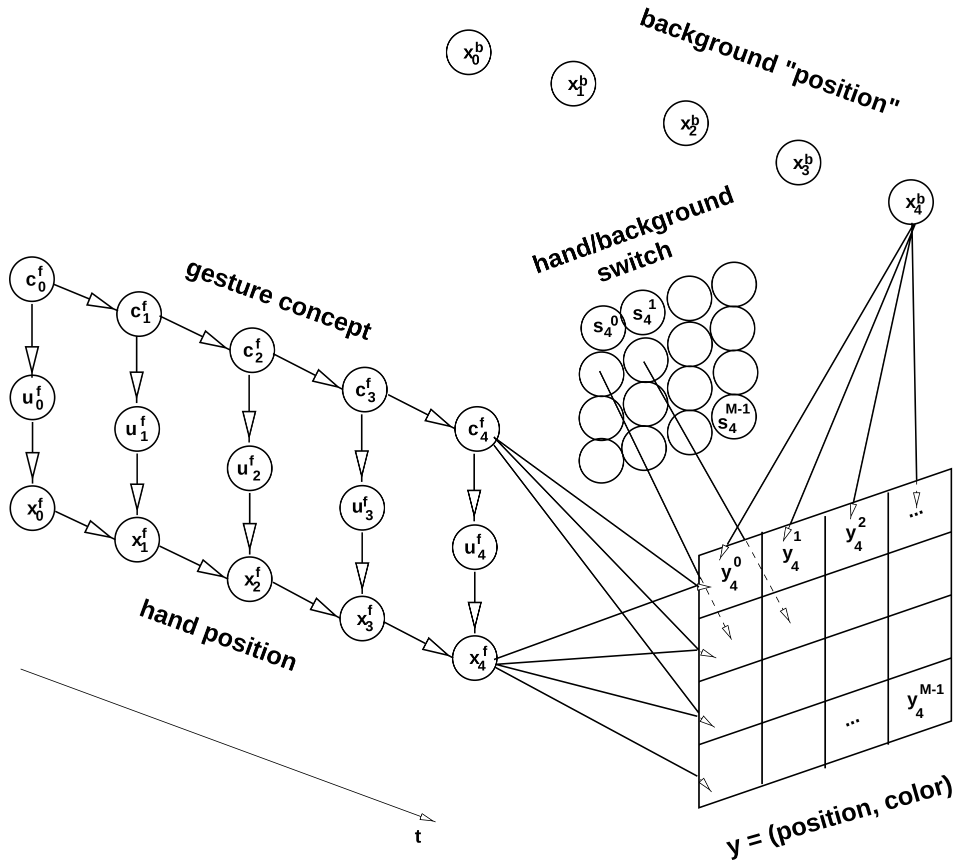


Figure 7.16 Bayesian network model for visual tracking of the human hand in scenes with stationary backgrounds. For convenience, only one time slice of observations and switching states is shown.

$$\times \prod_{t=0}^{T-1} \prod_{m=0}^{M-1} Pr(s_t^{(m)}) \tag{7.23}$$

$$\times \prod_{t=0}^{T-1} \prod_{m=0}^{M-1} Pr(y_t^{(m)} | x_t^{(f)}, x_t^{(b)}, s_t^{(m)}, c_t). \tag{7.24}$$

The main factors of the pdf correspond to

- linear dynamics of hand motion $x^{(f)}$ controlled by concept c (Equation 7.21),
- stationary dynamics of background $x^{(b)}$ (Equation 7.22), and
- observation model in image space y with associations controlled by association state s (Equations 7.23 and 7.24).

First consider the observation space and the observation association and generation model. From Figure 7.16 it follows that observation $y_t^{(m)}$ at time t corresponds to the mth pixel in the observation image.

Assuming that each pixel possesses *position and color* attributes one arrives at

$$y_t^{(m)} = \begin{bmatrix} \text{position of } m\text{th pixel at time t} \\ \text{color of } m\text{th pixel at time t} \end{bmatrix} = \begin{bmatrix} \chi_t^{(m)} \\ \varsigma_t^{(m)} \end{bmatrix}.$$

Thus, $\chi_t^{(m)}$ is the spatial coordinate of the m-th image pixel, $\chi_t^{(m)} = [\chi_{x,t}^{(m)} \quad \chi_{y,t}^{(m)}]'$. Similarly, $\varsigma_t^{(m)}$ is the color coordinate of the same pixel.

Furthermore, each pixel m in the image corresponds to either the hand or the background. The association of every pixel m with the hand or the background models is determined by the association state $s_t^{(m)}$. Namely, $s_t^{(m)}$ can take on values from the binary set $\{f, b\}$ according to distribution $Pr(s_t^{(m)})$, indicating whether the mth pixel belongs to the hand (f) or the background (b). Since every pixel can be associated with only one image object at any time t, we impose the following observation pdf structure:

$$\begin{aligned} Pr(y_t^{(m)}|x_t^{(f)}, x_t^{(b)}, s_t^{(m)} = f, c_t) &= Pr(y_t^{(m)}|x_t^{(f)}, c_t) \\ Pr(y_t^{(m)}|x_t^{(f)}, x_t^{(b)}, s_t^{(m)} = b, c_t) &= Pr(y_t^{(m)}|x_t^{(b)}, c_t). \end{aligned}$$

In particular, we choose the following pdf observation models:

Hand/foreground model. Assume that distribution of spatial and color attributes in an image of the human hand is conditionally independent[4] and has the following structure:

$$\begin{aligned} Pr(y_t^{(m)}|x_t^{(f)}, c_t) &= Pr(\chi_t^{(m)}, \varsigma_t^{(m)}|x_t^{(f)}, c_t) \\ &= Pr(\chi_t^{(m)}|x_t^{(f)}, c_t) Pr(\varsigma_t^{(m)}|x_t^{(f)}). \end{aligned}$$

Then, planar projection of the human hand at each time t has a Gaussian distribution in the image coordinate space with mean $x_t^{(f)}$ and one of N_c possible variances[5]:

$$Pr(\chi_t^{(m)}|x_t^{(f)}, c_t = i) = (2\pi)^{-\frac{1}{2}N_y} |R_i|^{-\frac{1}{2}} \exp\left\{ \left(\chi_t^{(m)} - Cx_t^{(f)}\right)' R_i{}^{-1} \left(\chi_t^{(m)} - Cx_t^{(f)}\right) \right\}, \qquad (7.25)$$

where, as before, $C = [1 \;\; 0]$ and $i = 0, \ldots, N_c - 1$ is the concept state index. Distribution of hand color is modeled by either Gaussian or discrete tabularized pdf $P_{\text{color}}^{(f)}$ in the color space of choice. Each pixel's color attribute is i.i.d. according to

$$Pr(\varsigma_t^{(m)}|x_t^{(f)}) = P_{\text{color}}^{(f)}(\varsigma_t^{(m)}).$$

Note that we assume stationary distribution of color attributes. Joint distribution of every pixel attribute (position and color) is now

$$Pr(y_t^{(m)}|x_t^{(f)}, c_t = i) = (2\pi)^{-\frac{1}{2}N_y} |R_i|^{-\frac{1}{2}} \exp\left\{ \left(\chi_t^{(m)} - Cx_t^{(f)}\right)' R_i{}^{-1} \left(\chi_t^{(m)} - Cx_t^{(f)}\right) \right\} P_{\text{color}}^{(f)}(\varsigma_t^{(m)}).$$

[4] Conditional independence of spatial and color attributes is assumed for convenience. However, a more realistic model may need to eliminate this assumption.

[5] This assertion is not completely correct. Given that an image coordinate space is bounded by image edges one needs to scale the Gaussian distribution by a normalization factor. The factor should guarantee that $\int_{\chi_t^{(m)} \in \text{image}} Pr(\chi_t^{(m)})\, d\chi_t^{(m)} = 1$. For image sizes much larger than the hand size and hand positions sufficiently far from image edges the normalization factor is very close to one.

Background model. Spatial distribution of background image pixels is uniform over the observation set (image)

$$Pr(\chi_t^{(m)}|x_t^{(b)}) = \frac{1}{M}.$$

The color of every background pixel is i.i.d. with a Gaussian or discrete tabularized pdf $P_{\text{color}}^{(b)}$:

$$Pr(\varsigma_t^{(m)}|x_t^{(b)}) = P_{\text{color}}^{(b)}(\varsigma_t^{(m)}).$$

Joint conditional distribution of background image pixel attributes is

$$Pr(y_t^{(m)}|x_t^{(b)}) = Pr(\chi_t^{(m)}, \varsigma_t^{(m)}|x_t^{(b)}) = \frac{1}{M} P_{\text{color}}^{(b)}(\varsigma_t^{(m)}),$$

as the color and spatial pixel attributes are again assumed conditionally independent.

Data association model. The data association pdf is assumed to be identical for all image pixels m at any fixed time instance, i.e.,

$$Pr(s_t^{(m)} = i) = P_{\text{assoc}}^{(t)}(i), \quad i \in \{f, b\},$$

where $P_{\text{assoc}}^{(t)}$ is the probability association table at time t. Therefore, for instance, the a priori probability of pixel m belonging to the hand at time t is $P_{\text{assoc}}^{(t)}(f)$.

Hand dynamics. What remains to be defined is the model of hand dynamics driven by gestural concepts. As before, assume that this model coincides with that of Section 7.3, a model derived from the mixed-state HMM model of Chapter 5. Hence,

$$Pr(x_t^{(f)}|x_{t-1}^{(f)}, c_t) = \mathcal{N}(Ax_{t-1} + BDc_t, Q),$$

and

$$Pr(c_t|c_{t-1}) = {c_t}' \; P_{\text{concept}} \; c_{t-1},$$

where P_{concept} denotes the concept state transition matrix.

7.4.1 Inference

The ultimate goal of gestural action modeling is the ability to infer the underlying gestural concept from a sequence of images of a moving hand. To achieve this goal one needs to efficiently handle the concept as well as the physical system state inference and decoding tasks. Given that the proposed model of gestural actions combines two intractable DBN models (mixture of DBNs and mixed-state DBN), it clearly follows that inference in the visually observed gestural action model is also not tractable. Thus, one needs to consider an approximate tractable inference solution.

Chapters 4 and 5 approached the intractable inference problems using structured variational inference. Therefore, it only makes sense to combine the two approximate solutions obtained in the above two

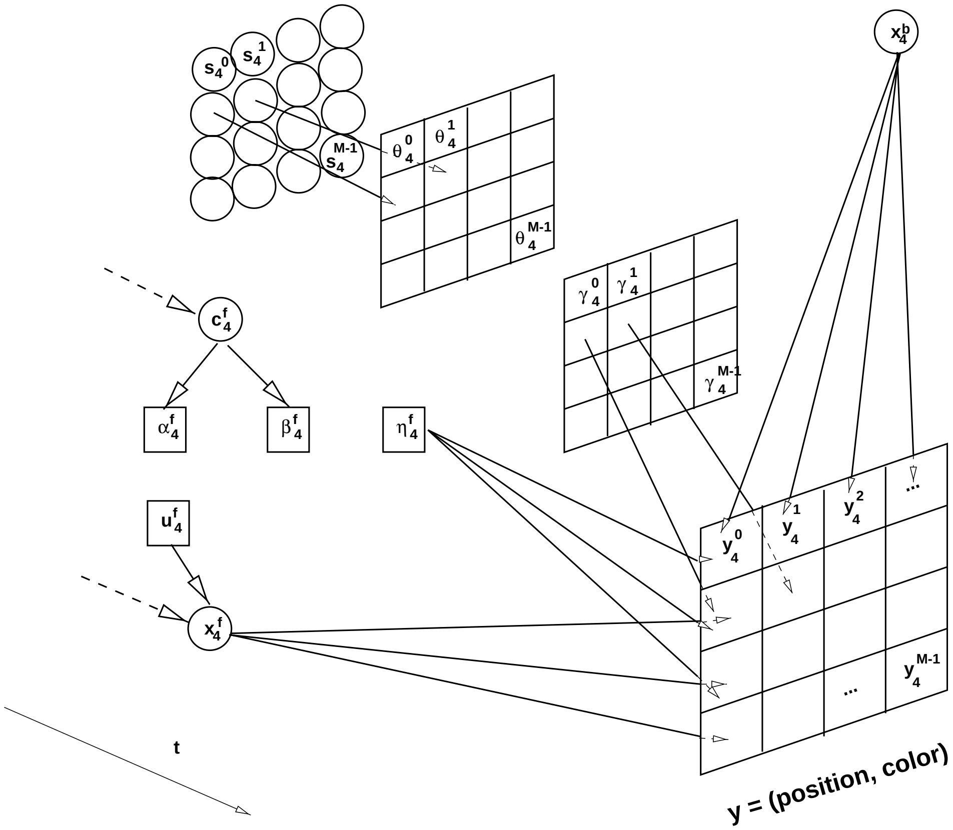


Figure 7.17 Factorized Bayesian network model for visual tracking of the human hand in scenes with stationary backgrounds. For convenience, only one time slice of observations and switching states is shown.

cases in order to solve my present inference problem. Combination of variational inference factorizations of Chapters 4 and 5 is trivial. Figure 7.17 now depicts the total factorization of joint pdf P. More precisely, factorized pdf Q can be written as

$$Q = Pr(c_0) \prod_{t=1}^{T-1} Pr(c_t|c_{t-1}) \prod_{t=0}^{T-1} Pr(\alpha_t|c_t) Pr(\beta_t|c_t \tag{7.26}$$

$$\times Pr(x_0^{(f)}|u_0) \prod_{t=1}^{T-1} Pr(x_t^{(f)}|x_{t-1}^{(f)}, u_t) \tag{7.27}$$

$$\times \prod_{t=0}^{T-1} Pr(x_t^{(b)}) \tag{7.28}$$

$$\times \prod_{t=0}^{T-1} \prod_{m=0}^{M-1} Pr(y_t^{(m)}|x_t^{(f)}, x_t^{(b)}, \gamma_t^{(m)}, \eta_t) \tag{7.29}$$

$$\times \prod_{t=0}^{T-1} \prod_{m=0}^{M-1} Pr(s_t^{(m)}) Pr(\theta_t^{(m)}|s_t^{(m)}). \tag{7.30}$$

Recall from the previous section the forms of specific pdfs that constitute the model. Given those particular forms, with the help of variational parameter derivations from Chapters 4 and 5, one easily arrives at the following set of fixed-point equations that yield optimal values of variational parameters α, β, γ, and η.

$$
\begin{aligned}
\alpha_\tau^{(i)} &= \exp\left\{ (Bd_i)' Q^{-1} \left(\left\langle x_\tau^{(f)} \right\rangle - A \left\langle x_{\tau-1}^{(f)} \right\rangle - \frac{1}{2} Bd_i \right) \right\} && (7.31) \\
u_\tau &= D \left\langle c_\tau \right\rangle && (7.32) \\
\gamma_\tau^{(m)} &= \left\langle s_\tau^{(m)} \right\rangle && (7.33) \\
\theta_\tau^{(m)}(f) &= \prod_{j=0}^{N_c-1} \left[(2\pi)^{-\frac{1}{2}N_y} |R_j|^{-\frac{1}{2}} \exp\left\{ -\frac{1}{2} \mathrm{tr}(R_j^{-1} \hat{R}_\tau^{(m)}) \right\} \right]^{\langle c_\tau(j) \rangle} && \\
&\quad P_{\text{color}}^{(f)}(\varsigma_\tau^{(m)}) && (7.34) \\
\theta_\tau^{(m)}(b) &= \frac{1}{M} P_{\text{color}}^{(b)}(\varsigma_\tau^{(m)}) && (7.35) \\
\beta_\tau(i) &= |R_i|^{-\frac{1}{2}} \exp\left\{ -\frac{1}{2} \mathrm{tr}(R_j^{-1} \hat{R}_\tau) \right\} && (7.36)
\end{aligned}
$$

where $i = 0, \ldots, N_c - 1$ refers concept states and $m = 0, \ldots, M-1$ corresponds to image pixels. Furthermore, $\hat{R}_\tau^{(m)}$ is an estimate of hand region (spatial) variance at time $t = \tau$ based on the mth image pixel:

$$
\hat{R}_\tau^{(m)} = \left\langle (\chi_\tau^{(m)} - Cx_\tau^{(f)})(\chi_\tau^{(m)} - Cx_\tau^{(f)})' \right\rangle .
$$

Similarly, $\hat{R}_\tau$ is an estimate of hand shape variance at time τ when all image pixels are taken into account:

$$
\hat{R}_\tau = \frac{\sum_{n=0}^{M-1} \hat{R}_\tau^{(n)} \left\langle s_\tau^{(n)}(f) \right\rangle}{\sum_{n=0}^{M-1} \left\langle s_\tau^{(n)}(f) \right\rangle}. \tag{7.37}
$$

The above set of fixed-point equations is, of course, in addition to the inference equation used to obtain sufficient statistics in each sub-net of factorization Q: $\langle c_\tau \rangle$, $\left\langle x_\tau^{(f)} \right\rangle$, $\left\langle x_\tau^{(f)} x_\tau^{(f)}{}' \right\rangle$, and $\left\langle s_\tau^{(m)} \right\rangle$. For instance, as outlined in Chapter 5, $\langle c_\tau \rangle$ is obtained from HMM inference on the sub-net with hidden state variables $c_\cdot$ and "observations" $\alpha_\cdot$ and $\beta_\cdot$. Similarly, one employs modified Kalman smoothing (see Chapter 4) on the set of observations $\{y_\cdot^{(0)}, \ldots, y_\cdot^{(M-1)}\}$ to obtain sufficient statistics $\left\langle x_\tau^{(f)} \right\rangle$ and $\left\langle x_\tau^{(f)} x_\tau^{(f)}{}' \right\rangle$. Finally, sufficient statistics $\left\langle s_\tau^{(m)} \right\rangle$ are calculated using the Bayesian estimate of Equation 4.7:

$$
\begin{aligned}
\left\langle s_\tau^{(m)}(f) \right\rangle &= \frac{\theta_\tau^{(m)}(f) P_{\text{assoc}}^{(\tau)}(f)}{\theta_\tau^{(m)}(f) P_{\text{assoc}}^{(\tau)}(f) + \theta_\tau^{(m)}(b) P_{\text{assoc}}^{(\tau)}(b)} \\
\left\langle s_\tau^{(m)}(b) \right\rangle &= \frac{\theta_\tau^{(m)}(b) P_{\text{assoc}}^{(\tau)}(b)}{\theta_\tau^{(m)}(f) P_{\text{assoc}}^{(\tau)}(f) + \theta_\tau^{(m)}(b) P_{\text{assoc}}^{(\tau)}(b)}.
\end{aligned}
$$

We summarize the newly obtained inference algorithm as follows:

```
error = $\infty$;
```

```
Initialize $\langle x^{(f)} \rangle$ and $\langle c \rangle$;
while (error > maxError) {
        Find $\alpha_\cdot$, $\beta_\cdot$, and $\theta_\cdot$ from $y_\cdot$, $\langle x_\cdot \rangle$, and $\langle c_\cdot \rangle$ using Equations 7.34,7.35,7.36 and 7.31;
        Estimate $\langle s_\cdot \rangle$ from $\theta_\cdot$ using Bayesian rule;
        Estimate $\langle c_\cdot \rangle$ from $\alpha_\cdot$ and $\beta_\cdot$ using HMM inference;
        Find $u_\cdot$ from $\langle c_\cdot \rangle$ using Equation 7.32;
        Find $\gamma_\cdot$ from $\langle s_\cdot \rangle$ using Equation 7.33;
        Estimate $\langle x_\cdot \rangle$ from $u_\cdot$ and $\gamma_\cdot$ using LDS inference;
        Update Cost i.e., bound on $P(\mathcal{Y})$;
        error ← ( oldCost - Cost ) / Cost;
}
```

7.4.2 Spatially localized analysis

In the generalized inference formulation of the previous section we subtlely introduced one assumption: at each time instance a complete scene image is used as an observation set. Yet, in most applications the hand occupies only a small portion of the whole scene (see Figure 7.9, for example). In such cases it is not necessary to use all available image pixels to estimate the hand dynamics and concept states.[6] It is sufficient to concentrate on the region of interest (ROI) within an image where the hand is most likely to be. Such an ROI is in fact easy to determine. Recall that the conditional distribution of the hand's spatial attribute at time t is Gaussian with mean $Cx_t^{(f)}$ and variance R_i (Equation 7.25). Therefore, the exponent of that distribution, $z = \left(\chi_t^{(m)} - Cx_t^{(f)}\right)' R_i^{-1} \left(\chi_t^{(m)} - Cx_t^{(f)}\right)$, has χ^2 distribution with N_y degrees of freedom. To find a spatial ROI where most of the hand pixels will lie one simply needs to determine a high confidence region of choice for z. Then, one only takes into account image pixels within that region, as the ROI now plays the role of the whole image of the previous section. The size and position of this ROI change in time. Hence, one needs to adjust the uniform spatial distribution of background pixels to reflect this change in size. Instead of analyzing an elliptical ROI, as defined by the confidence measure, it is sometimes more convenient to use a rectangular bounding box of the original ROI (see Figure 7.9).

7.4.3 Prediction and on-line inference

In the previous section we studied the classical inference problem for my model of visually observed hand gestures. The formulation of the problem assumes that all observation data points $\mathcal{Y} =$

[6] In fact, it is to one's disadvantage to use all pixels since, as will be seen in Section 7.4.4, the estimates of the hand association probability $P_{\text{assoc}}^{(t)}(f)$ will become significantly smaller than the corresponding background association probability estimates. This will in turn bias all future variable estimates towards background.

$\{y_0^{(0)}, \ldots, y_{T-1}^{(M-1)}\}$ are readily available and that one needs to estimate states of the corresponding hidden variables in the same time interval, i.e., 0 to $T-1$. However, in many cases new observation points arrive constantly. For example, to track a gesturing hand in a live video stream one needs to predict the hand's position in the next image frame as well as the ROI size. Given the topology of the gesture model it readily follows that the predicted position of the hand is

$$\hat{x}_T^{(f)} \quad = \quad \left\langle x_T^{(f)} | \mathcal{Y}_0^{T-1} \right\rangle = A \left\langle x_{T-1}^{(f)} \right\rangle + BDP_{\text{concept}} \left\langle c_{T-1} \right\rangle .$$

The predicted concept state is similarly

$$\hat{c}_T \quad = \quad \left\langle c_T | \mathcal{Y}_0^{T-1} \right\rangle = P_{\text{concept}} \left\langle c_{T-1} \right\rangle .$$

ROI size depends on the variance of the hand's spatial distribution. It is trivial to show that predicted hand shape variance becomes:

$$\hat{R}_T = \left\langle R | \mathcal{Y}_0^{T-1} \right\rangle = \sum_{j=0}^{N_c-1} R_i \hat{c}_T .$$

Finally, one would like to have an initial estimate of association pdf at $t = T$, $P_{\text{assoc}}^{(T)}$. Our model states, however, that the associations at time t only depend on image data at that particular time. We therefore assume that the change in association probabilities is small between two consecutive time instances:

$$\hat{P}_{\text{assoc}}^{(T)} = P_{\text{assoc}}^{(T-1)} .$$

Given the above four estimates $\hat{x}_T$, $\hat{c}_T$, $\hat{R}_T$, and $\hat{P}_{\text{assoc}}^{(T)}$, one has sufficient information to determine ROI at $t = T$. Once the new image data $y_T^{(0)}, \ldots, y_T^{(M-1)}$ has arrived one can use those estimates to initialize the approximate inference algorithm.

With the arrival of new data, the inference algorithm not only reestimates sufficient statistics at the current time (such as $\langle s_T \rangle$ and $\langle x_T \rangle$), but also "smoothes" all the old ones ($\langle s_t \rangle$, $t = 0, \ldots, T-1$, for example). This means that, at least in theory, one needs to run the inference algorithm on the whole data history $\{y_0, \ldots, y_T\}$. Needless to say, such a requirement is not practical for long data sequences. In practice, fortunately, it is not necessary to smooth out the whole data sequence. Rather, one only reestimates a "small" number of most recent variables. That "small" number is usually determined by the largest time constant of the network in question.

7.4.4 Learning

The DBN model of visually observed hand gestures possesses several classes of parameters, each corresponding to a particular subnet of the model. Concept parameter, for instance, is the concept state probability transition table P_{concept}. Hand motion parameters are the input force levels D and the hand state variance Q, while hand shape and color are parameterized with the shape variance R_i and hand color probability table $P^{(f)}_{\text{color}}$. Spatial distribution of background pixels is defined to be uniform, whereas its color is parameterized with the background color probability table $P^{(b)}_{\text{color}}$. Finally, the pixel-to-model association is determined by the probability table P_{assoc}.

To learn the above-mentioned parameters we resort again to the generalized EM learning introduced in Chapter 2. Earlier in Section 7.4 we explained the origins of the model—it is a combination of mixed-state HMM and mixture-of-DBN models from Chapters 4 and 5. Hence, parameter learning of the visually observed hand gesture model simply carries over from the parameter update solutions of the two simpler models. For the sake of completeness, we now state the parameter update equations using the new model's notation.

Concept parameters.

$$P_{\text{concept}} = \left(\sum_{t=1}^{T-1} \langle c_t c'_{t-1} \rangle \right) \text{diag} \left(\sum_{t=1}^{T-1} \langle c_t \rangle \right)^{-1} \tag{7.38}$$

Hand motion parameters.

$$D = \left(B' Q(T)^{-1} B \right)^{-1} B' Q(T)^{-1} \left(\sum_{t=1}^{T-1} \left\langle (x_t^{(f)} - A x_{t-1}^{(f)}) {c_t}' \right\rangle \right) \left(\sum_{t=1}^{T-1} \langle c_t {c_t}' \rangle \right)^{-1} \tag{7.39}$$

$$\begin{aligned} q = {} & \frac{1}{2(T-1)} \\ & \text{tr} \left\{ Q(T)^{-1} \sum_{t=1}^{T-1} \left\langle (x_t^{(f)} - A x_{t-1}^{(f)})(x_t^{(f)} - A x_{t-1}^{(f)})' \right\rangle - BD \left\langle c_t (x_t^{(f)} - A x_{t-1}^{(f)})' \right\rangle \right\}. \end{aligned} \tag{7.40}$$

Hand shape parameters.

$$R_i = \frac{\sum_{t=0}^{T-1} \sum_{m=0}^{M-1} \left(\chi_t^{(m)} \chi_t^{(m)\prime} - C \left\langle x_t^{(f)} \right\rangle \chi_t^{(m)\prime} \right) \left\langle s_t^{(m)}(f) \right\rangle \langle c_t(i) \rangle}{\sum_{t=0}^{T-1} \sum_{m=0}^{M-1} \left\langle s_t^{(m)}(f) \right\rangle \langle c_t(i) \rangle} \tag{7.41}$$

Hand color parameters.

$$P^{(f)}_{\text{color}} = \frac{\sum_{t=0}^{T-1} \sum_{m=0}^{M-1} \varsigma_t^{(m)} \left\langle s_t^{(m)}(f) \right\rangle}{\sum_{t=0}^{T-1} \sum_{m=0}^{M-1} \left\langle s_t^{(m)}(f) \right\rangle}. \tag{7.42}$$

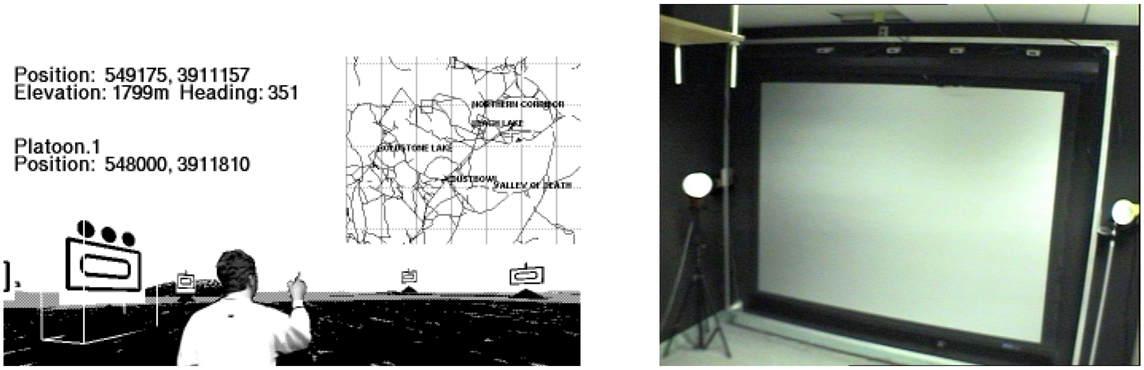
Background color parameters.

$$P^{(b)}_{\text{color}} = \frac{\sum_{t=0}^{T-1} \sum_{m=0}^{M-1} \varsigma_t^{(m)} \left\langle s_t^{(m)}(b) \right\rangle}{\sum_{t=0}^{T-1} \sum_{m=0}^{M-1} \left\langle s_t^{(m)}(b) \right\rangle}. \tag{7.43}$$

Association parameters.

$$P^{(t)}_{\text{assoc}}(f) = \frac{\sum_{m=0}^{M-1} \left\langle s_t^{(m)}(f) \right\rangle}{\sum_{m=0}^{M-1} \left\langle s_t^{(m)}(f) \right\rangle + \left\langle s_t^{(m)}(b) \right\rangle} \tag{7.44}$$

$$P^{(t)}_{\text{assoc}}(b) = \frac{\sum_{m=0}^{M-1} \left\langle s_t^{(m)}(b) \right\rangle}{\sum_{m=0}^{M-1} \left\langle s_t^{(m)}(f) \right\rangle + \left\langle s_t^{(m)}(b) \right\rangle} \tag{7.45}$$


Position: 549175, 3911157
Elevation: 1799m Heading: 351
Platoon.1
Position: 548000, 3911810
NORTHERN CORRIDOR
DUSTBOWL
VALLEY OF DEATH

(a) BattleView Display Control

(b) SmallWall Setup

Figure 7.18 Experimental testbed for integrated multimodal interaction. (a) User interaction with BattleView simulation environment. (b) SmallWall frontal projection setup.

7.4.5 Experiment: Analysis and recognition of visually perceived hand gestures

In this section we consider the task of visual hand tracking and hand gesture recognition. It appears that this task is, at least in principle, analogous to the task of recognizing the "mouse gestures" of Section 7.3.5. However, it introduces a significant new burden not present in the simpler case of "mouse gestures": the need for visual hand tracking of a gesturing hand.

The study of visually perceived hand gestures was conducted with one particular framework in mind. Instead of focusing on a large and vaguely defined class of natural communicative gestures, we chose to explore a novel domain of hand gestures for computer display control. Much like the mouse gestures, hand gestures for computer display control are aimed at further easing the burden of old fashioned human–computer interaction (HCI). However, free hand gestures eliminate the burden of device cables and allow more natural interaction. More importantly, unlike the natural communicative gestures, their display control counterpart can be highly structured and constrained. This fact was exploited to test the feasibility of the DBN-based visual gesture model.

The testbed application was a simple virtual display control task. In this application the user manipulates a virtual display (terrain or an object on virtual terrain, for instance) shown on the projection screen (see Figure 7.18(a)) using a set of eleven visually observed gestural actions. The set of commands is summarized in Table 7.3. For example, the user would perform the "move left" gestural action to change the viewing point of the 3D map along the horizontal axis. A "select" action would engage a pointer with which one could select an object of interest. "Rotate" actions would cause an object or a map to slowly rotate about some predefined axis. Finally, "stop" would terminate any currently engaged action.

Table 7.3 Gestural commands for virtual display control. Dynamic stroke gestural actions involve highly dynamic hand motion. Static stroke actions, on the other hand, exhibit limited hand motion during gesture stroke phases. Two-letter codes in parentheses are used to denote respective gestural actions.

	Commands			
Dynamic stroke				
	move left (LT)	move right (RT)	move up (UP)	move down (DN)
	move forward (FW)	move backward (BW)	rotate left (RL)	rotate right (RR)
Static stroke				
	stop (ST)	select (SL)	release (RS)	

All gestural actions of the human user are observed visually with a color camera strategically mounted on the display unit. This allows the system to obtain an unobstructed view of the freely moving user's hands and head. Examples of video shots of all eleven gestural action commands are depicted in Figure 7.19.

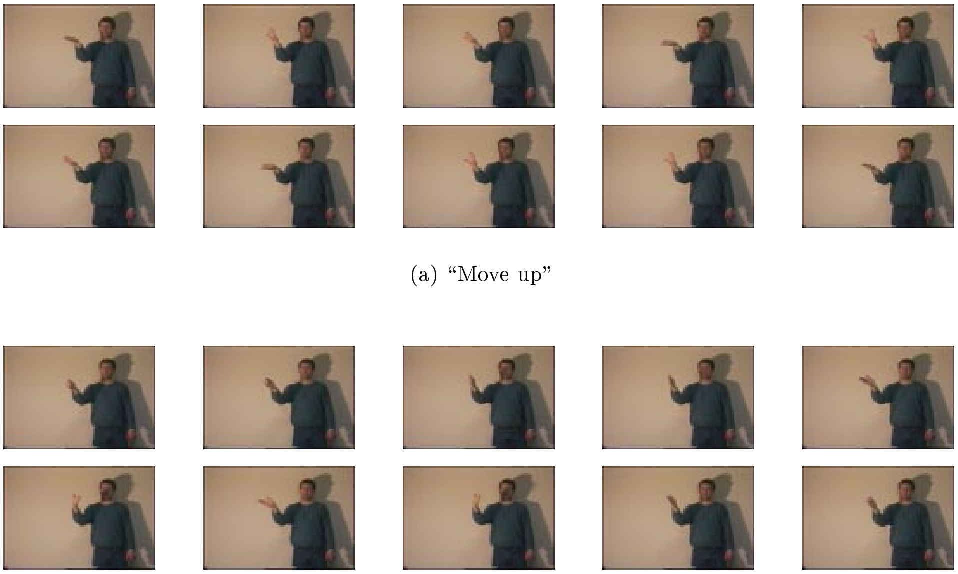
7.4.5.1 Initial Hand Tracking

The first task was to obtain initial estimates of trajectories of the moving hand without explicit knowledge of the underlying gestural concepts. This was needed because at the initial stage no gestural concept model was available. To achieve this task, we used a simplified (initializing) model of Section 7.4 for any hand motion. In the initializing model the mixed-state HMM structure (coupled concept/hand dynamics model) is replaced by "one order higher" dynamical system, similar to the initialization of the "mouse gesture" model (see Section 7.3.4). Again, in this simplification we assume that the hand motion driving force can be modeled as a noise-corrupted stationary state of the dynamical model. Since the concept model does not exist in this initial formulation, we assume that the hand shape variance R is allowed to vary freely with time, i.e., $R_t = \hat{R}_t$ as in Equation 7.37.

With this visual gesture model in mind one still needs to somehow select the initial model parameters: dynamic state noise variance q, foreground and background color distributions P_{color}, and assignment pdf P_{assign}. Dynamic state noise variance q was initially set to an arbitrary value and then adaptively adjusted to maintain 99.9 percent confident innovation estimate [76]. Background and foreground color distributions were estimated from color histograms of known (interactively selected) hand skin and background image patches. Finally, foreground and background object assignments were chosen to be equiprobable.

Concurrent with hand tracking, we also tracked the more stationary head in the same manner. The tracking algorithm now assumes the following form:

```
for ( $t = 0, \dots$ ) {
```



(a) “Move up”

(b) “Move left”

Figure 7.19 Examples of free hand gestural actions. Shown are selected frames of “move up” (a) and “move left” (b) gestural actions.

```
for ( head & hand ) {
    error = $\infty$;
    $\left\langle x_t^{(f)} \right\rangle = A \left\langle x_{t-1}^{(f)} \right\rangle + B \left\langle u_{t-1}^{(f)} \right\rangle$;
    while (error > maxError) {
        Find $\theta_t$ from $y_t$ and $\langle x_t \rangle$ using Equations 7.34 and 7.35;
        Estimate $\langle s_t \rangle$ from $\theta_t$ using Bayesian rule;
        Find $\gamma_t$ from $\langle s_t \rangle$ using Equation 7.33;
        Estimate $\left\langle x_t^{(f)} \right\rangle$ and $\langle u_t \rangle$ from $\gamma.$ using LDS filtering;
        Estimate $R_t$ using Equation 7.37;
        Update Cost i.e., bound on $P(\mathcal{Y})$;
        error $\leftarrow$ ( oldCost - Cost ) / Cost;
    }
}
}
```

Note that in the above tracking formulation all $\langle \cdot \rangle$ operators refer to filtered or predicted estimates of the appropriate variables and not to their smoothed values as is the case in the general inference algorithm. Of course, it is still possible, if necessary, to smooth out all the estimates (see Section 7.4.3).

This initial tracking procedure in general yielded satisfactory results. Several sequences of hand motions, each with more than 8000 frames, were tracked without any lost tracks. One example of an intermediate tracking step that includes foreground/background segmentation and spatial pdf reestimation is shown in Figure 7.20.

7.4.5.2 Recognition of gestural actions

Data set. The data set for the experimental session consisted of two video sequences of 40 gestural commands each. The two sequences are denoted GVS1 and GVS2.[7] Each of the forty commands in both video sequences was performed by a single gesturer and selected from the set of eleven possible actions (see Table 7.3). Table 7.4 shows the transcript of both GVS1 and GVS2.

Concept model. For every gestural action in the set, the following semantical structure was imposed:

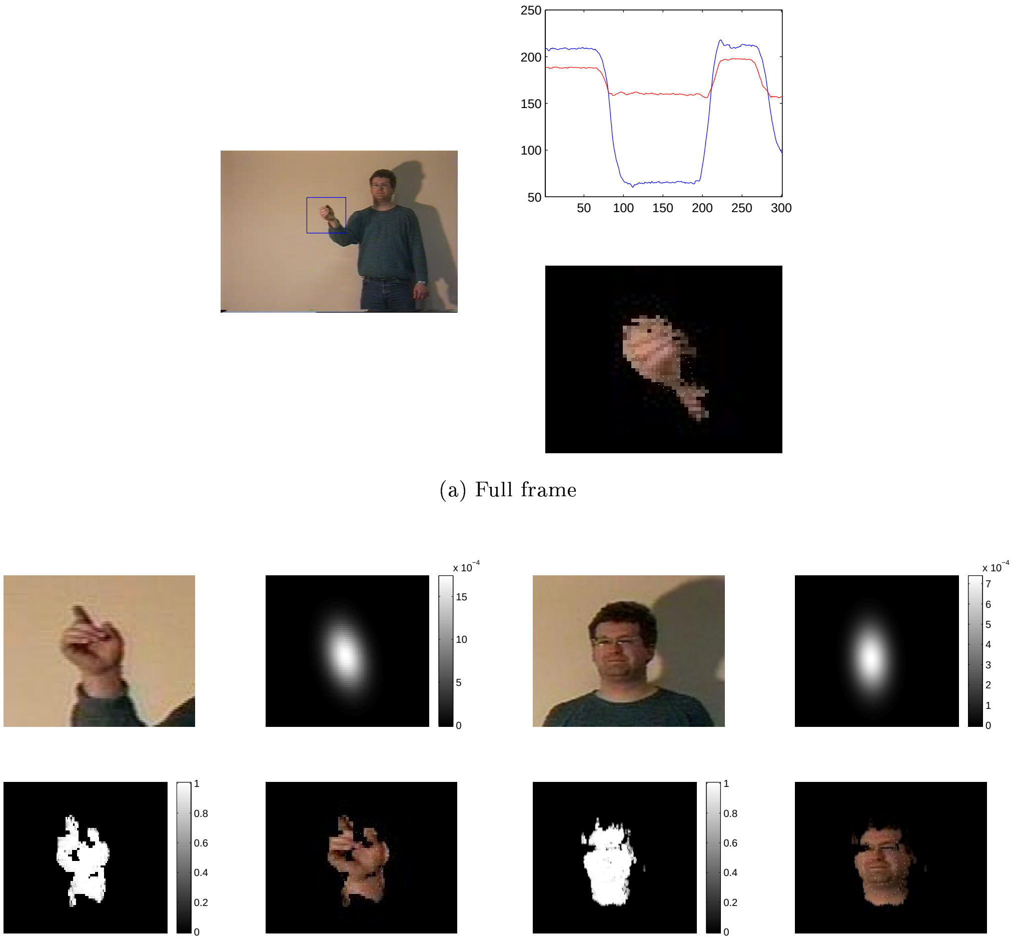
- Every gestural action consists of three phases: preparation, stroke, and retraction. This assumption directly adheres to the known temporal structure of natural gestures, as outlined in Section 7.1.3.
- Every repetitive dynamic gestural stroke consists of a sequence of basic gestural movements referred to here as *gestimes*. Each gestime represents one coherent unit of hand motion. For example, "move up" gestural stroke consists of a sequence of "up" and "down" gestimes. Furthermore, the number of repetitions of gestime pairs is assumed to have binomial distribution.

Complete semantics of the command language are depicted in the state diagram of Figure 7.21. Semantics define a simple probabilistic grammar where every gestural action is equally likely.[8] Action models of dynamic gestures such as "move up" and "mode down" that consist of identical gestimes differ in probabilistic weights associated with the unit transitions. Hence, a model of the "move up" gesture, for example, is defined as shown in Figure 7.22. The concept behind every gestural unit ("preparation," "retraction," static stroke, or gestime) is in turn assumed to be modeled as a discrete Markov chain, in accordance with the DBN-based visual gesture model. To reduce the complexity of inference we constrain all concept Markov chain models to have "left-to-right" transition probability matrices and a small number of states. In particular, "preparation" and "retraction" phase concepts are modeled as three-state chains, static stroke concepts as five state chains, whereas all gestimes had four state concept spaces. The above structures have been shown to yield the best inference results when compared to other similar concept model topologies.

Given the semantics of the gestural command language, it follows readily that every concept of every gestural action is itself a Markov chain. However, unlike the dynamics of isolated gestural units which

[7] GVS - gestural video sequence.

[8] In a more realistic situation some actions would be more likely than the others. Also, probabilistic weights could be imposed within action-pairs or triplets, thus constructing bigram or trigram probabilistic language models [126].



(a) Full frame

(b) Hand ROI

(c) Head ROI

Figure 7.20 Hand and head tracking using DBN. Figure (a) depicts frame 300 of one complete gestural video set. Shown are also hand trajectory estimates and hand ROI with segmented hand region. Elliptical lines are isolevels of estimated spatial distribution of the hand. Figures (b) and (c) are examples of hand and head segmentations, respectively. Left to right, top to bottom: ROIs, foreground spatial pdfs $Pr(\chi^{(\cdot)}_{\cdot} \mid x^{(f)}_{\cdot})$, foreground assignment estimates $\left\langle s^{(\cdot)}_{\cdot}(f) \right\rangle$, and foreground ROI pixel sets.

Table 7.4 Transcript of training (GVS1) and test (GVS2) gesture video sequences.

Index	Action
001	select
002	up
003	left
004	forward
005	down
006	stop
007	release
008	forward
009	right
010	backward
011	rotright
012	forward
013	select
014	rotleft
015	release
016	backward
017	right
018	select
019	up
020	left
021	down
022	release
023	rotright
024	forward
025	stop
026	select
027	rotleft
028	up
029	backward
030	down
031	release
032	stop
033	right
034	left
035	stop
036	up
037	forward
038	stop
039	backward
040	rotright

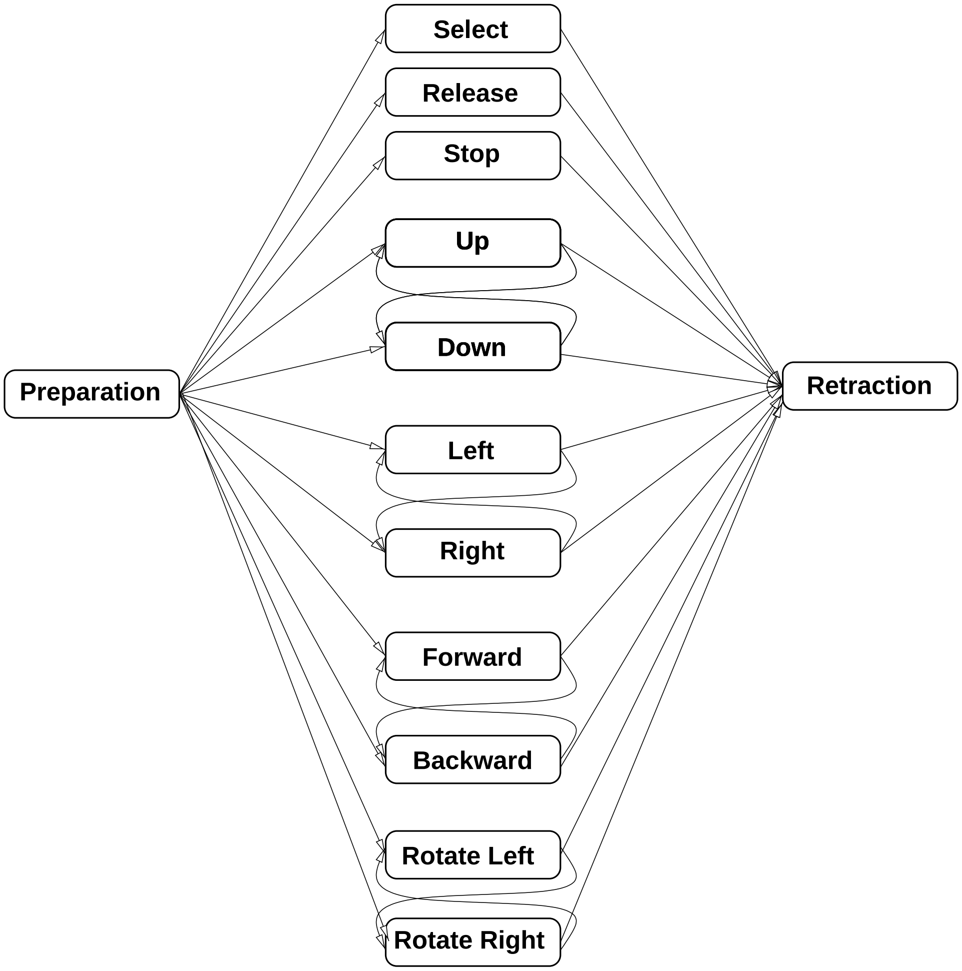


Figure 7.21 Grammar of gestural command language.

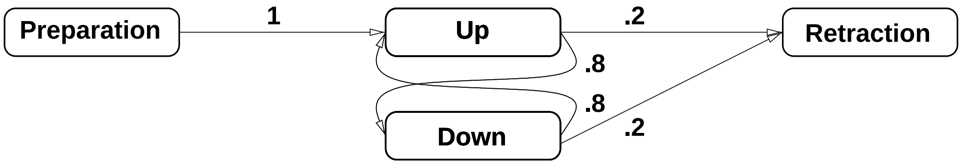


Figure 7.22 Model state diagram of "move up" gestural action.

are always (constrained to be) noncyclic (i.e., “left-to-right”), the dynamics of gestural action concepts can be cyclic. Cyclic behavior occurs in models of dynamic stroke gestural actions, such as “move left,” etc.

Hand dynamics. Hand motion dynamics of gestural action are, as before, modeled as linear dynamic systems (see beginning of Section 7.4). For simplicity, assume that the hand motion model of every gestural action is a fixed-parameter LDS. In other words, all units that make up any particular gestural action have the same motion model parameters. Whereas this assumption may negatively reflect on the dynamics of the hand itself, it makes the overall DBN model more tractable.

Silence model. In addition to models of eleven gestural actions, we defined a model of any nongestural action, referred to as the *silence model*. The silence model was assumed to have no motion dynamics and only a single concept state. Given the constrained structure of the data set, the silence model in essence corresponded to the resting hand position and played the role of a “hand-up/hand-down” detector.

Training. In the training phase, model parameters of gestural action concepts and the accompanying hand motions are learned from the sequences of images that exemplify those actions. The general learning algorithm for the DBN model was introduced in Section 7.4.4. Initial estimates of motion model parameters were obtained from the concept-decoupled tracking procedure outlined in Section 7.4.5.1. Concept parameters of every gestural unit including the coupling matrix D were initialized from the tracking estimates of the driving input force following the initialization technique for mixed-state HMMs (see Chapter 5). Parameters of the complete gestural actions were then reestimated in the general visual gesture model learning framework of Section 7.4.4. On the average, the 0.1 percent relative error in the cost function minimization was reached within ten iterations of the learning and inference algorithms.

Out of two video data sequences, one was designated the training set (GVS1) and the other the test set (GVS2).

Recognition. The task of gesture recognition was to accurately identify gestural actions in the test video sequence. Initially, gestural action periods were segmented from the video sequence using the “hand-up/hand-down” silence detector. Following this, periods of gestural activity were classified using the DBN-based gestural action models of eleven gestural commands.

As a base-line comparison we employed two decoupled models of gestural actions with sets of linear and nonlinear features derived from hand positional data augmented by available hand shape descriptors. In particular, one of the models was simply a decoupled version of our model where the HMM observations were modeling the input of the LDS and the hand shape as a mixture of Gaussian distributions (see Figure 7.23). In the other model a nonlinear mapping was introduced between the decoupled LDS and the HMM, as depicted in Figure 7.23. The mapping was essentially a linear-to-polar coordinate space transformation that was claimed to be successful in modeling hand gestural actions [121].

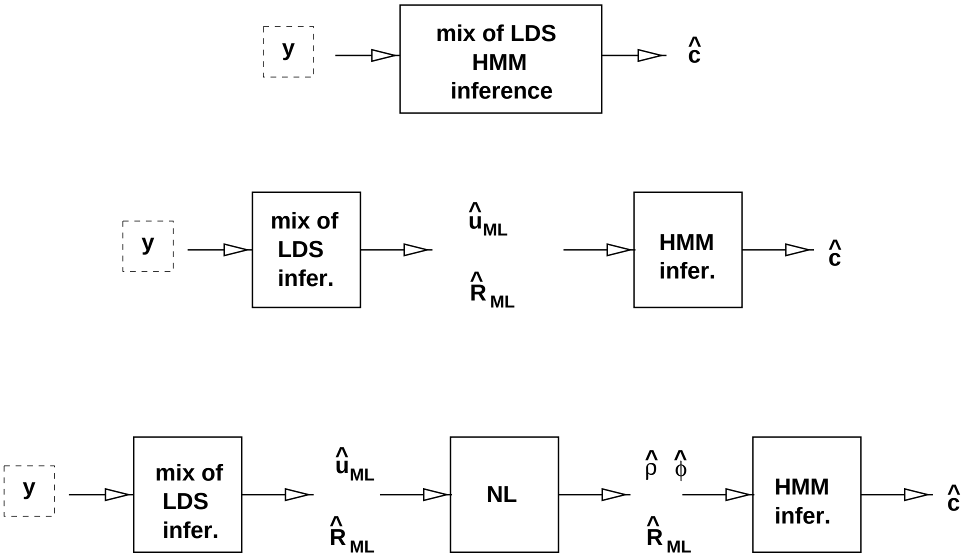


Figure 7.23 Block diagrams of three gestural action classifiers. Top to bottom: coupled mixture of LDS/HMM classifier uses approximate probabilistic inference to decode the states of gestural concepts $\hat{c}$ from images of moving hand y; decoupled LDS/HMM and its nonlinear counterpart estimate the hand motion driving force and hand shape without the knowledge of underlying gestural actions.

All three models were trained on the GVS1 set and tested on set GVS2. Table 7.5 shows the misclassification error estimates for the three types of gestural action models. Error estimates were obtained using standard counting methods (maximum likelihood estimates of binomial/Bernoulli model probabilities [125]). From Table 7.5 it seems that, as expected, the coupled mixture of LDS/HMM model performs better then the two decoupled models. However, strictly speaking, with a confidence level of $p = .05$, the performances of the three models cannot be distinguished (confidence intervals are about ±13 percent). This is clearly due to the lack of data since only 40 action sequences of eleven actions are available. Besides collecting and analyzing more data, additional discrimination of performances may be achieved through cross-validation tests.

To gain more insight in the gesture classification performance it is useful to consider the confusion matrix associated with the classification task. Confusion matrices in Tables 7.6, 7.7, and 7.8 depict cumulative classification results over all eleven action commands for the three tests. For instance, column 1 of Table 7.6 indicates that out of four occurrence of the action "select" in the test set, three were correctly recognized and one was misclassified as "release." It is clear from Table 7.6 that majority of misclassifications occur in the case of nonplanar forward/backward and rotate left/right actions. This is somewhat expected since we assume a planar model of hand motion when in fact the forward/backward actions are almost perpendicular to the camera plane. The other common source of errors stems from inadequate discrimination of opposite gestural actions. For instance, it is sometimes the case that a "move right" gestural action begins with a hesitant downward movement. Finally, static gestural actions

Table 7.5 Misclassification error estimates [%] and error estimate variances ([%]) for classification of 40 gestural actions from GVS2 data set.

Action	Coupled mixture of LDS/HMM Error[%] (Var[%])	Decoupled mixture of LDS/HMM Error[%] (Var[%])	Nonlinear decoupled model Error[%] (Var[%])
Select	25 (4.69)	25 (4.69)	25 (4.69)
Stop	40 (4.80)	60 (4.80)	80 (3.20)
Release	0 (0)	0 (0)	50 (6.25)
Up	0 (0)	0 (0)	25 (4.69)
Down	0 (0)	0 (0)	100 (0)
Left	0 (0)	33.3 (7.41)	66.7 (7.41)
Right	33.3 (7.41)	33.3 (7.41)	66.7 (7.41)
Forward	20 (3.20)	40 (4.80)	20 (3.20)
Backward	75 (4.69)	75 (4.75)	75 (4.69)
Rotate Left	50 (12.50)	100 (0)	100 (0)
Rotate Right	0 (0)	0 (0)	0 (0)
TOTAL	22.5 (0.43)	32.5 (0.55)	52.5 (0.62)

Table 7.6 Confusion table of gesture classification with coupled mixture of LDS/HMM model. Columns are original actions, rows are classified actions.

	SL	ST	RS	UP	DN	LT	RT	FW	BW	RL	RR
SL	**3**	0	0	0	0	0	0	0	0	0	0
ST	0	**3**	0	0	0	0	0	0	0	0	0
RS	1	1	**4**	0	0	0	0	1	0	0	0
UP	0	0	0	**4**	0	0	0	0	0	0	0
DN	0	0	0	0	**3**	0	0	0	0	0	0
LT	0	0	0	0	0	**3**	1	0	0	0	0
RT	0	0	0	0	0	0	**2**	0	0	0	0
FW	0	1	0	0	0	0	0	**4**	1	0	0
BW	0	0	0	0	0	0	0	0	**1**	0	0
RL	0	0	0	0	0	0	0	0	0	**1**	0
RR	0	0	0	0	0	0	0	0	2	1	**3**

Table 7.7 Confusion table of gesture classification with decoupled mixture of LDS/HMM model. Columns are original actions, rows are classified actions.

	SL	ST	RS	UP	DN	LT	RT	FW	BW	RL	RR
SL	**3**	0	0	0	0	0	0	0	0	0	0
ST	0	**2**	0	0	0	0	0	0	0	0	0
RS	1	2	**4**	0	0	0	0	2	0	0	0
UP	0	0	0	**4**	0	0	0	0	0	0	0
DN	0	0	0	0	**3**	0	0	0	0	0	0
LT	0	0	0	0	0	**2**	1	0	0	0	0
RT	0	0	0	0	0	0	**2**	0	0	0	0
FW	0	1	0	0	0	0	0	**3**	1	0	0
BW	0	0	0	0	0	0	0	0	**1**	0	0
RL	0	0	0	0	0	0	0	0	0	**0**	0
RR	0	0	0	0	0	1	0	0	2	2	**3**

Table 7.8 Confusion table of gesture classification with decoupled nonlinear model. Columns are original actions, rows are classified actions.

	SL	ST	RS	UP	DN	LT	RT	FW	BW	RL	RR
SL	**3**	1	2	0	0	0	0	0	0	0	0
ST	0	**1**	0	0	1	0	0	1	0	0	0
RS	0	1	**2**	0	0	0	0	0	1	0	0
UP	0	0	0	**3**	0	0	0	0	0	0	0
DN	0	0	0	0	**0**	0	0	0	0	0	0
LT	0	0	0	0	0	**1**	0	0	0	0	0
RT	0	0	0	0	1	1	**1**	0	0	1	0
FW	0	1	0	0	1	0	1	**4**	0	0	0
BW	0	0	0	0	0	0	0	0	**1**	0	0
RL	0	0	0	0	0	0	0	0	0	**0**	0
RR	1	1	0	1	0	1	1	0	2	1	**3**

("stop," "release," and "select") can only be distinguished by the hand posture itself. Unfortunately, the second-order shape model in use does not alway manage to capture enough information to robustly distinguish between the three hand shapes.

We conclude this example by noting that the relatively poor recognition performance of gestural actions stems from several factors: large variability in the actions themselves, unmatched motion model space (2D versus 3D), and insufficient shape feature descriptors. One would thus contend that without addressing the above issues first, automatic hand gesture recognition may remain far from being usable. Fortunately, as will be seen in Chapter 6, one can exploit modalities beyond the visual to bring even the seemingly poor gestural model back into the HCI field.

CHAPTER 8

MULTIMODAL ANALYSIS AND RECOGNITION OF HAND GESTURES AND SPEECH USING DYNAMIC BAYESIAN NETWORKS

8.1 Introduction

The interaction of humans with their environment (including other humans) involves multiple, concurrent modes of communication. However, when it comes to HCI only one interface device is usually used at a time—typing, clicking the mouse button, speaking, or pointing with a magnetic wand. The "ease" with which this unimodal interaction allows one to convey her intent to the computer is far from satisfactory. A number of reasons may lead one to consider multimodal HCI as opposed to the unimodal one. One may consider such reasons to be of practical, biological, and mathematical nature, as discussed in Section 1.2. Given the overwhelming evidence in favor of multimodal HCI, we next discuss different approaches to fusing multiple sensing modalities. The focus is on hand gestures and speech.

8.2 When to Integrate Gestures and Speech

Different sensing modalities yield disparate signal forms and rates. For instance, auditory signals occur at a rate of 20–20,000 Hz and are preprocessed by a human within 30 ms. Visual signals, on the other hand, are perceived at about 25 Hz, and their early processing takes on the order of 150 ms. That makes successful integration of such signals a difficult and challenging task.

The answers on how closely coupled the two modalities are mostly originate in psycho-behavioral studies concerned with the interaction of modalities. For instance, it is known that gestures and speech are intimately connected and are claimed to arise from a single mental concept [127]. Gestures occur synchronously with their semantically parallel speech units or just before them [127]. However, a question remains as to whether such coupling persists when the modalities are used for HCI. Several "Wizard of Oz" type studies have confirmed that it does. Similar conclusions have been drawn from a few usability studies that involved systems with novel gesture/speech interfaces. Oviatt [6] has, for example, extensively studied the interaction of drawing gestures and speech in palmtop HCI applications. She has concluded that the integration occurs on a semantic level where gestures are used to convey information on location while speech conveys the information on subject and action (verb) in a sentence. Overall, one can say that on a semantic level speech and gestures can have one of two relationships: *complementary* or *reinforcing.* The complementary role of speech and gestures often occurs in natural communication. For example, as noted by Oviatt [6], gestures are often used to specify spatial properties and relationships between objects, while speech is naturally better suited to express actions and certain nonspatial object attributes that can be hard to express through hand movements. When one wants to specify that an object's color should be changed to red, he usually points to an object and utters, "Make this red." On the other hand, in certain situations gestures and speech can play reinforcing roles. For instance, to make a person (or a virtual person) move leftward one can say "move left, move left" while simultaneously performing a "move left" gesture.

Another interesting perspective that may shed more light on the study of the levels of multimodal integration in HCI comes from the field of sensory data fusion. For the most part, three distinct levels of integration can be distinguished [47]: *data fusion*, *feature fusion*, and *decision fusion.* Data fusion is the lowest level of fusion. It involves integration of raw observations and can occur only in the case when the observations are of the same type. This type of fusion cannot be associated with the integration of gestures and speech because the two are observed using different types of sensors (video camera and microphone, for instance). What is known as feature fusion in sensory data literature is more commonly found in integration of modalities for HCI. It assumes that each stream of sensory data is first analyzed for features, after which the features themselves are fused. This type of fusion is appropriate for closely coupled and synchronized modalities. In general, feature-level fusion retains less detailed information than data fusion, but is also less sensitive to noise in raw data. The type of fusion most commonly found in HCI is the so-called decision-level fusion. Decision-level fusion is based on the fusion of individual mode decisions or interpretations. For example, once an arm movement is interpreted as a deictic (pointing) gesture and a spoken sentence is recognized as "Make this box white," the two can be fused to interpret that a particular object (box) needs to be painted white. Synchronization of modalities in this case pertains to synchronization of decisions on a semantic level. From numerous studies in the sensory fusion field, we know that decision fusion is qualified as the most robust and resilient to individual sensor failure. It has a low data bandwidth and is generally less computationally expensive than feature fusion. One disadvantage of decision-level fusion is that it potentially cannot recover from loss of information that occurs at lower levels of data analysis, and thus does not exploit the correlation between the modality streams at the lower integration levels.

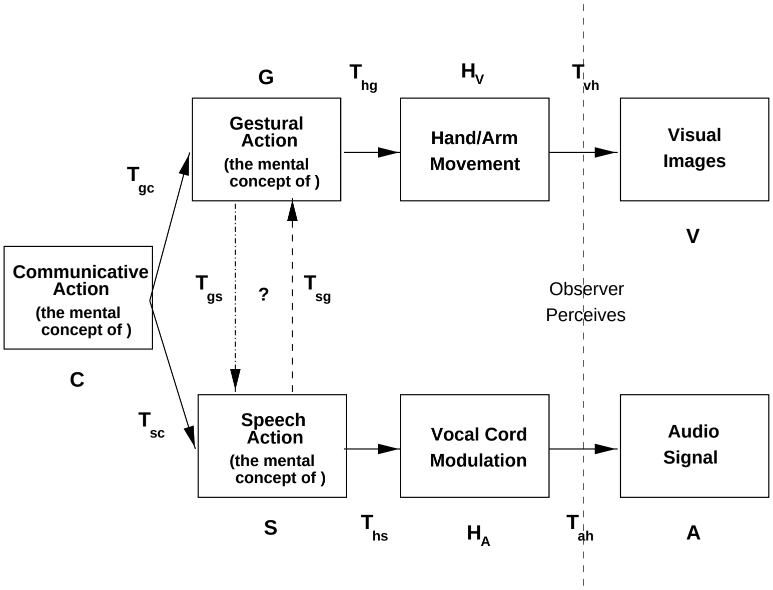


Figure 8.1 Generative multimodal model. Gestures and speech originate as a communication concept C. It is arguable whether and to what extent there is interaction between the two modal concepts. Speech and gestures are separated in the processing phase and expressed using different modalities H_V and H_A. Finally, they are perceived using hearing A and vision V.

Keeping in mind this general notion of possible semantic roles and fusion levels, we next focus on modeling the interaction between spoken language and hand gesturing. In particular, we propose a dynamic Bayesian network (DBN) framework that automatically extracts and models any possible interaction between the two modalities.

8.3 How to Integrate Gestures and Speech

As mentioned in the previous section, the level at which the integration is done strongly influences the actual computational mechanism used for the integration. To tackle the integration task, we propose an integration framework based on *dynamic Bayesian networks* (see Chapter 3).

8.3.1 A general integration model

To formulate the model for multimodal (speech/gesture) integration, consider first a feasible generative model, as depicted in Figure 8.1. The model separates the generation of speech and gestures at some early stage of a mental concept, satisfying the observations noted in Section 8.2. To construct an integration model for multimodal recognition, one can consider the inverse of the generative model. From visual images and audio signals, one would independently process the individual modalities and classify them to the point of obtaining some estimates of the individual multimodal concepts $\hat{G}$ and

$\hat{S}$. Finally, one would fuse them at that level to obtain an estimate of the communication concept $\hat{C}$. However, it is not clear whether links from one modality's concept lead to another modality. Moreover, if they do, how strongly do they influence each other? Hence, independent processing and recognition of individual modalities to the point of their respective concepts may not be the best solution.

Recall now the *coupled HMM* discussed in Chapter 6. The framework was motivated by one's goal to model interactions among systems with multiple and possibly different discrete state dynamics and different observation spaces. Of course, this formulation fits perfectly with the need to formalize variable interactions between spoken and gestural language concepts. With a single coupled HMM-based multimodal concept model one can describe interactions anywhere from tightly coupled to loosely interacting cases. Control over the interactivity of modes is accomplished through a combination of intermodal parameters or intermodal weights, denoted as w in Chapter 6, and intermodal probabilistic couplings $P^{(m,n)}, m \neq n$. It the next two sections we consider in more detail how one can impose desired interactivity by varying the structure of the intermodal parameters.

8.3.2 Feature-level coupling

The case of tightly coupled speech and gestures can occur when the two modalities play either of their two semantic roles (reinforcing and complementary). If speech and gestures complement each other, the close coupling may be exhibited as the close coupling of all states of their conceptual models G and S. However, even though their conceptual spaces are coupled it is still reasonable to assume that each mode has unique concept dynamics different from the other mode. Hence, the *mixed HMM model with stationary weights* of Section 6.3.2.2 can be employed to describe this case. The "intensity" of coupling can be measured through the value of intermodal weights $w^{(m)}(n)$ and intermodal conditional pdfs: the more tightly coupled the two modalities, the more highly peaked the pdfs. If the coupling is tight to the point of identical or almost identical concept dynamics, then the model reduces to the classical HMM with concatenated observations (see Section 6.3.1). On the other hand, the tight coupling can be exhibited only over small temporal segments. For instance, in the "move this tank to region A" command, the onset of deictic action can be correlated with the beginning of the word "this." In that case the *temporally adaptive mixed HMM* from Section 6.3.2.3 can be a better choice.[1] Based on the values of intermodal weights $w_t^{(m)}(n)$ one can pinpoint to the intervals of stronger and weaker couplings.

8.3.3 Decision-level coupling

Loose coupling of speech and gestures implies that while the two modalities still may describe the same concept ("move left," for instance) the concept states themselves are uncorrelated on the feature

[1] Note, however, that the mixed HMM with stationary weights can also be seen as a special case of the variable temporal weight model.

level. In the example of the "move left" action, this would mean that the dynamics of the gesturing hand have nothing in common with the dynamics of the verbal mode. Hence, given the action concept C, all concept states of the two modes G and S are independent. In terms of the usual HMM models this means that an independent HMM can be assigned to model each modality separately. However, the same effect can be achieved using a fixed-weight coupled HMM with intermodal weights set to zero, $w^{(m)}(n) = 0, m \neq n$. As in the case of feature-level couplings, the noninteractivity of weights can be learned from data. Thus, one can initially employ an adaptive-weight coupled HMM and "learn" that the modes are uncorrelated.

8.4 Previous Work

Interest in fusion of multiple modalities, such as speech and hand gestures, has been around for more than a decade. Numerous feature- and decision-level approaches have been studied, mostly with emphasis on practical, sometimes ad hoc, solutions.

8.4.1 Feature-level coupling

Feature fusion context was introduced into the multimodal realm with HMMs whose observations were modeled as concatenated multimodal feature vectors. Such essentially unimodal[2] integration architectures have been considered for the fusion of speech and lip movements [128]. However, such initial architectures did not perform well. One of the first DBN-like multimodal architectures was the Boltzmann zipper [80]. In Boltzmann zipper for each hidden state can "belong" to only one of the multiple modalities (audio or video, for example) but not to both, as is the case in the unimodal HMMs. This architecture was applied to bimodal speech recognition and shown to yield an improvement in coupled interpretation [17] over the unimodal approach. A basis for coupled HMM architectures was established by the work of Brand [81, 75]. He introduced a coupled HMM architecture similar to that of Chapter 6 with fixed, unbiased intermodal weights, and he developed a Viterbi-like approximate inference scheme.

8.4.2 Decision-level coupling

Coupling of multiple modalities on the decision-level is the most frequently followed approach to multimodal integration. It involves fusion of concepts (decisions) from the individual modes to form a unique multimodal concept. An underlying assumption of this type of fusion is that the basic features of the individual modes are not sufficiently correlated to be fused at the feature level. Most of the decision-level fusion mechanisms commonly found in HCI systems are based on the concept of *frames*.

[2] Unimodal here refers to the fact that the process dynamics are unimodal, i.e., both audio and video are assumed to have identical dynamics.

The concept of frames is commonly found in artificial intelligence literature. A frame is a unit of a knowledge source describing an object [129]. Each frame has a number of slots associated with it. The slots represent possible properties of the object, actions, or the object's relationship with other frames. This last property facilitates a mechanism for designing networks of frames for a particular context with links describing contextual semantics. Such networks are also known as semantic networks [130]. In the multimodal HCI context, different modalities can be associated with individual frame slots. Different modalities can describe particular properties of a virtual object. Speech can, for instance, designate the object's color while gestures can imply the object's location. This is a case of the complementary role of the modalities. It is also possible that multiple modalities indicate the same property of an object. In such cases, fusion can be achieved by selecting the property with the lowest joint cost. In the Bayesian framework, this is equivalent to choosing the highest prior or posterior joint probability. An alternative may be to consider the Dempster-Shafer combination of evidence [131].

Frame-based multimodal HCI systems have been utilized ever since the famed Bolt's "Put-that-there" system [132] that employed speech, gaze, and hand gestures to manipulate virtual objects. Nigay and Coutaz [133] proposed a frame-based multifeature system design space that emphasized duality in modalities' roles and occurrences. They distinguished between parallel and sequential use of multiple modalities that may have a complementary or reinforcing role. Many recent systems used similar mechanisms. For example, [122] used speech and pen gesture frame fusion to design an interface for a calendar program "Jeanie." As classifiers for the individual modes it employed MS-TDNNs [122] for gesture recognition and the JANUS [134] speech translation system for the recognition of speech. Another, more complicated frame-based architecture was developed as a part of QuickSet [123, 135], a multimodal interface for control of military simulations using hand-held personal digital assistants (PDAs). Utilizing the artificial neural network and hidden Markov model classifiers for concurrent gesture recognition, multiple modalities in QuickSet played re-enforcing roles. The modalities could automatically disambiguate each other using joint ML estimation. Alternatively, unimodal interaction could be enabled when one of the modes become unreliable. Numerous other systems, such as "Finger-pointer" [106], "Virtual-World" [136], ALIVE [137], "Smart Rooms" [138], and "Neuro Baby" [139, 140], utilized similar frame-based architecture for integration of speech and simple gestures.

Many simple framed-based approaches have also been implemented for bimodal (audio and video) speech recognition [128, 141, 142]. Such approaches basically assume one frame—one-slot networks for each of the two modalities. The slots describe phonemes observed through speech and lip movements. Two frames are fused by selecting the phoneme with the highest joint probability. The classifiers for the individual modes are commonly of the HMM type.

Table 8.1 Spoken commands for virtual display control. Dynamic and static actions are associated with their respective gestural counterparts.

	Commands			
Dynamic actions				
	move/go left	move/go right	move/go up	move/go down
	move/go forward	move/go backward	rotate left	rotate right
Static stroke				
	stop	select this	release	

8.5 Experiment: Coupled Interpretation of Hand Gestures and Speech for Display Control

Recall the visual gesture recognition experiment from Section 7.4.5. In it a set of hand gestural commands was utilized to control a large-scale display application. In this new experiment, in the identical setup, we augmented the set of gestural commands with a number of verbal actions. The verbal set was selected so as to globally reinforce the appropriate gestural commands. The decision to employ this type of interaction was twofold. As seen in Section 7.4.5, unimodal visual gesture recognition suffers from inadequacies related to visual analysis of hand movements. Hence, it was reasonable to assume that the presence of reinforcing speech can straightforwardly help enhance the recognition performance. Secondly, the reinforcing setup enabled me to easily investigate behavior of model parameters with some a-priori guess as to how they should behave.

Data set. In the new experimental setup each gestural action in data sets GVS1 and GVS2 was "duplicated" with one of the verbal commands from Table 8.1. Hence, a "rotate left" gesture was accompanied by "rotate left" utterance while the "select this" spoken command occurred at the same time as the "select" gesture.

Modeling. As with visual gestures, we employed two baseline "classical models" of multimodal integration for the purpose of comparison with the coupled HMM framework. One model enforced a priori the decision-level integration, i.e., we used a fixed-weight coupled HMM (see Section 6.3.2.1) with zero intermodal weights. The second baseline model, on the other hand, imposed strong feature-level coupling between concept states; a unimodal HMM with concatenated audio/video features was used for this purpose. The two baseline models were contrasted to fixed, adaptive time-invariant, and adaptive time-varying weight-coupled HMMs of Chapter 6. This is depicted in Figure 8.2. In addition to the five multimodal models, independent models of gestural and verbal actions were also constructed.

To exploit the possible correlation of speech and gestures, we focused on modeling the complete actions in question. Namely, each action expressed concurrently through audio and visual streams was

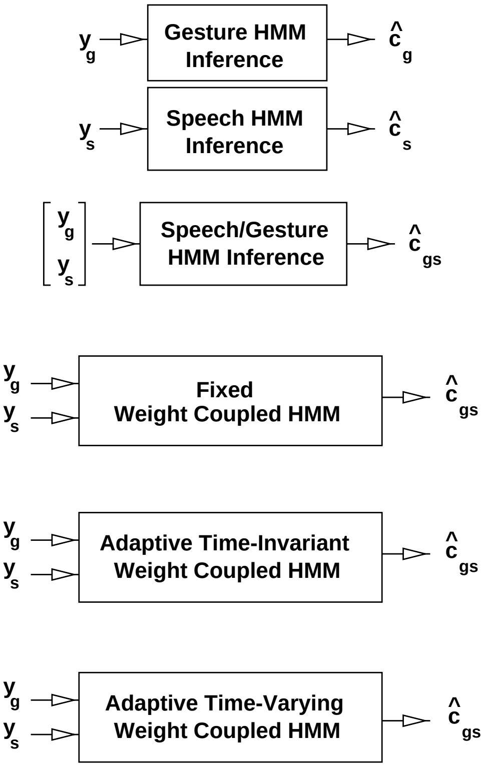


Figure 8.2 Competing inference models for bimodal gesture/speech recognition. Top to bottom: decision-level inference, highly coupled feature-level inference, naively coupled inference, coupled inference with adaptive time-invariant weight HMM, and coupled inference with adaptive time-varying weight HMM.

modeled using one of the five model types. Thus, the "move right" action was modeled as a single gesture/speech model that draws its observations from both audio and video streams.

Training. For all of the models in question we used auditory and visual features extracted independently of their respective concept models. Namely, audio features were selected to be ten MFCC coefficients and their temporal derivatives and were computed every 33 ms on frames of 50-ms duration (see [5] for details, for instance). Visual features were obtained using the decoupled hand tracking described in Section 7.4.5.1. Hence, they were the estimates of hand driving force and second-order hand shape descriptors.

Given the sets of available features derived from data set GVS1, each of the five models was trained using its own learning procedure. We consider the example of the "move left" action to outline the intermediate modeling steps for the five model types. All coupled models of "move left" were initialized using independent models of the underlying speech and visual hand movements. Hence, the gestural concept of "move left" from Section 7.4.5 was used for that purpose. The independent concept model of the verbal part of this action was obtained by combining the word-level HMMs of the words "move" and "left" with appropriate silence models. The decision-level model of the "move left" action therefore consisted of the above two models. A highly coupled unimodal model of "move left" was obtained when the independent gestural action model was retrained with concatenated audio/visual features. The two independent models also served as initial conditions for model parameter reestimation of the fixed- and the adaptive-weight coupled models (see Chapter 6). In the fixed-weight coupled model we imposed a higher degree of concept correlation by constraining all coupling weights to be equal, $w^{(n)}(m) = 0.5$.

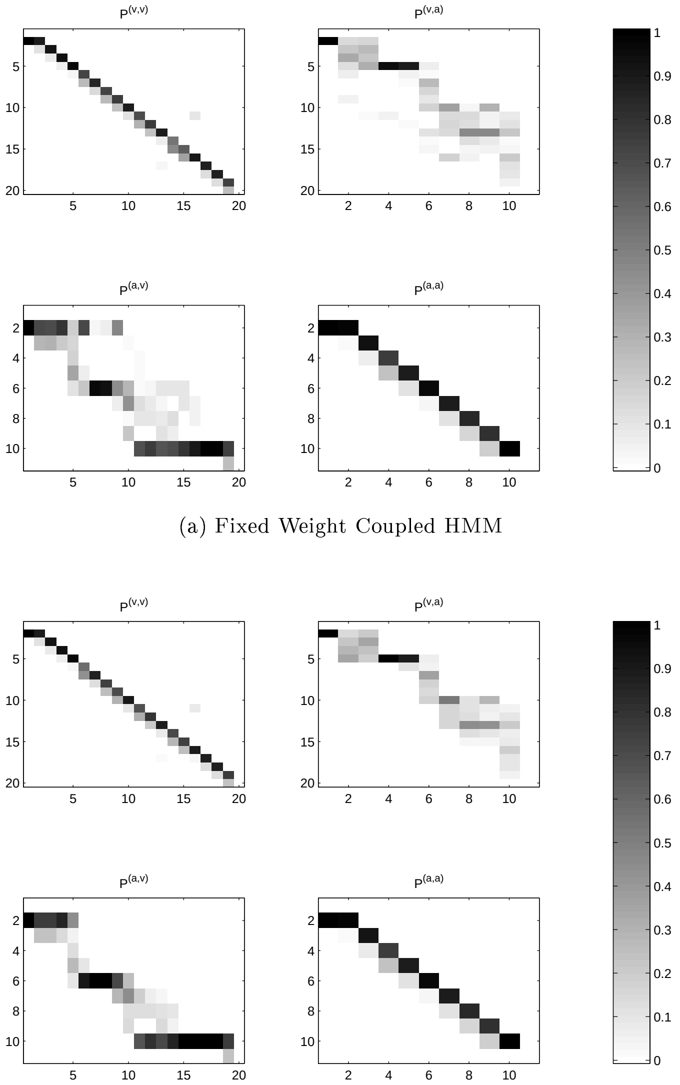
All parameter learning techniques converged, on the average, within 5 and 10 iterations to yield 0.1 percent relative error in the cost function. This is, of course, in addition to the number of iterations necessary for variational inference of the coupled model states. With 0.1 percent relative error threshold, variational inference usually converged within the first ten steps.

It is interesting to consider the results of model parameter training. In particular we focus on the state transition parameters of three coupled HMM networks. Figures 8.3(a), 8.3(b), and 8.3(c) depict state transition probability matrices, both intra- and intermodal, of the action "move/go up." As a reference, Figure 8.4 shows segmentation of one training example of the same action. One can conclude from these figures that, in fact, there is relatively little correlation between gestural and spoken concept states. The correlation exists primarily between periods of silence in speech and preparation/retraction phases of gestures. Low state level interaction between the two modalities is also confirmed through adapted weight values. Their estimated distributions are highly peaked around intramodal transitions. For example, for action "up" $\langle w^{(v)}(v) \rangle = 0.9950$ while $\langle w^{(v)}(a) \rangle = 0.0050$.

Recognition. As for recognition of gestural actions in Section 7.4.5, the task was to accurately interpret the actions of one user, this time by analyzing both his gestural hand movements and his verbal commands. Two sets of experiments were performed. One involved classification of unaltered audio/visual data from GVS2. The other set examined the influence of high levels of audio noise on joint action classification. Namely, the audio signal from GVS2 was corrupted with zero-mean white noise to yield a 5 dB signal-to-noise ratio.[3]

Classification performance results on the original GVS2 data are shown in Table 8.2. Misclassification rates are in general lower for all coupled methods than for the independent ones with the exception of the highly coupled unimodal model (AV2). This clearly confirms the fact that most correlation between speech and gestures tends to occur on the semantic level [143]. More insight into performance of our models can be gained by considering the second ("noisy") experimental task (see Table 8.3). In this task the isolated audio misclassification rate (A) increases to 37.5 percent, on the same order as the visual gesture classification. It is hoped that the joint gesture/speech recognition may do better than that, and in fact it does. Table 8.3 indicates that the coupled classification methods outperform the independent speech and gesture interpretations. Most notably, the adapted time-varying weight model achieves the lowest misclassification rate of 22.5 percent. The other two coupled models from Chapter 6

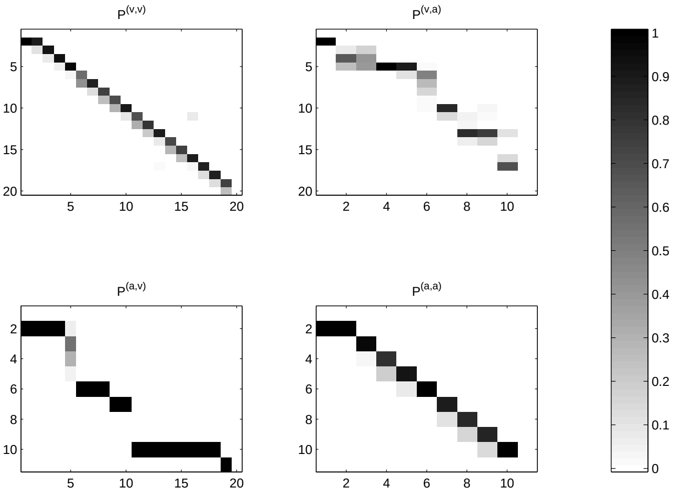
[3] The unaltered audio signal had a SNR of close to 30 dB.



(a) Fixed Weight Coupled HMM

(b) Adaptive Time-Invariant Weight Coupled HMM

Figure 8.3 State transition probability matrices for three coupled HMM models of speech/gesture action "up." Superscripts (v, v), (v, a), (a, v), and (a, a) denote visual intramodal, audio-to-visual intermodal, video-to-audio intermodal, and visual intramodal transition pdfs.



(c) Adaptive Time-Varying Weight Coupled HMM

Figure 8.3 Continued.

are comparable to the decision-level coupling method. Again, the highly coupled AV2 method falls out of the group of "good performers." Of course, one needs to keep in mind the computational complexity associated with the coupled model inference techniques. As is obvious from Chapter 6, every iteration of the variational inference algorithm involves two classical HMM inferences, one for each modality. On the average, convergence within 0.1 percent relative error in approximation was reached within five iterations. This in turn implies that on the average the complexity of the coupled inference is five-fold the complexity of the decoupled one. Finally, we include in the presentation of results the details of confusion matrices for all covered test cases (see Tables 8.4 to 8.15 in this chapter and Tables 7.6 through 7.8 in Chapter 7.)

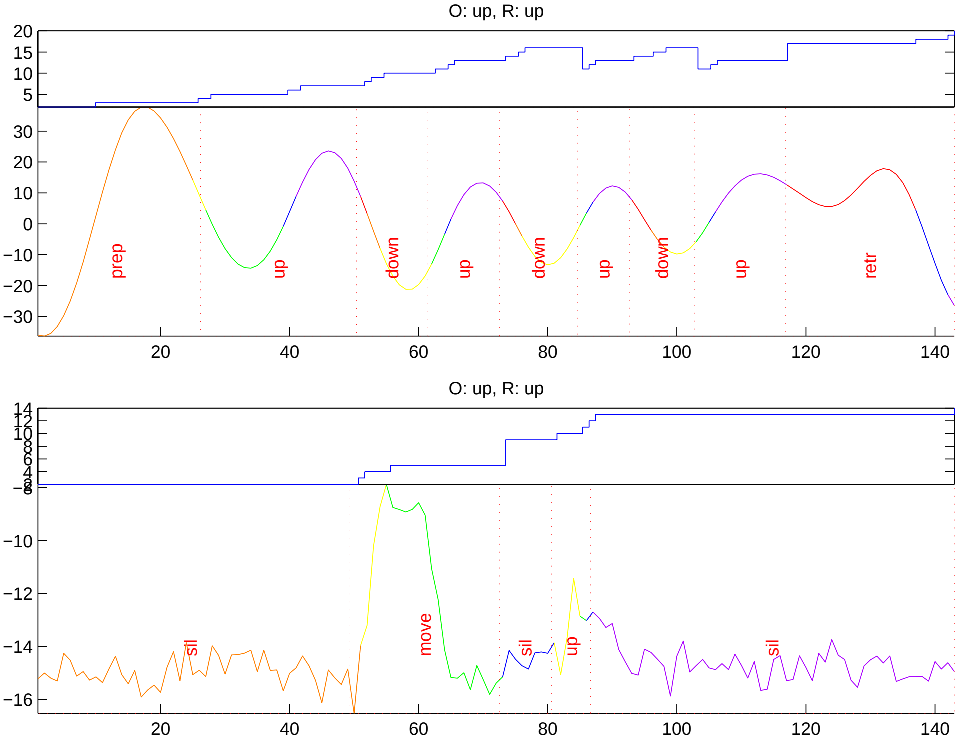


Figure 8.4 Segmentation of visual (top) and verbal (bottom) parts of action "up." Shown are gestime-level segmentations of visual unit force in x direction and word-level segmentations of the first MFCC coefficient of the verbal action. Also depicted (small graphs) are the concept states of the respective actions.

Table 8.2 Misclassification error estimates [%] and error estimate variances ([%]) for classification of 40 audio-visual actions from GVS2 data set. Columns correspond to different classification models: **A** - independent audio, **V** - independent video, **AV1** - decision-level coupled, **AV2** - state level coupled, **CAV1** - coupled with fixed weights, **CAV2** - coupled with adapted time-invariant weights, and **CAV3** - coupled with adapted time-varying weights.

Action	A	V	AV1	AV2	CAV1	CAV2	CAV3
SL	0 (0)	25 (4.68)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
ST	0 (0)	60 (4.80)	0 (0)	20 (3.20)	0 (0)	0 (0)	0 (0)
RS	25 (4.68)	0 (0)	25 (4.68)	0 (0)	0 (0)	0 (0)	25 (4.68)
UP	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
DN	33 (7.41)	0 (0)	0 (0)	100 (0)	100 (0)	100 (0)	0 (0)
LT	0 (0)	33 (7.41)	0 (0)	100 (0)	0 (0)	0 (0)	33 (7.41)
RT	0 (0)	33 (7.41)	0 (0)	33 (7.41)	0 (0)	0 (0)	0 (0)
FW	0 (0)	40 (4.80)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
BW	0 (0)	75 (4.68)	25 (4.68)	100 (0)	0 (0)	0 (0)	0 (0)
RL	100 (0)	100 (0)	100 (0)	100 (0)	0 (0)	0 (0)	0 (0)
RR	33 (7.41)	33 (7.41)	0 (0)	0 (0)	33 (7.41)	33 (7.41)	33 (7.41)
TOTAL	12.5 (2.73)	32.5 (0.55)	10 (0.22)	35 (0.57)	10 (0.22)	10 (0.22)	7.5 (0.17)

Table 8.3 Misclassification error estimates [%] and error estimate variances ([%]) for classification of 40 audio-visual actions from GVS2 data set with 5dB SNR on audio.

Action	A	V	AV1	AV2	CAV1	CAV2	CAV3
SL	25 (4.68)	25 (4.68)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
ST	100 (0)	60 (4.80)	100 (0)	100 (0)	100 (0)	100 (0)	80 (3.20)
RS	0 (0)	0 (0)	0 (0)	100 (0)	0 (0)	0 (0)	25 (4.68)
UP	25 (4.68)	0 (0)	0 (0)	0 (0)	100 (0)	0 (0)	0 (0)
DN	100 (0)	0 (0)	100 (0)	33 (7.41)	33 (7.41)	100 (0)	66.7 (7.41)
LT	0 (0)	33 (7.41)	0 (0)	100 (0)	0 (0)	0 (0)	0 (0)
RT	0 (0)	33 (7.41)	0 (0)	67.7 (7.41)	0 (0)	0 (0)	0 (0)
FW	0 (0)	40 (4.80)	0 (0)	40 (4.80)	0 (0)	0 (0)	0 (0)
BW	100 (0)	75 (4.68)	50 (6.25)	100 (0)	50 (6.25)	50 (6.25)	0 (0)
RL	100 (0)	100 (0)	0 (0)	100 (0)	0 (0)	0 (0)	50 (12.5)
RR	33 (7.41)	33 (7.41)	33 (7.41)	33 (7.41)	33 (7.41)	33 (7.41)	33 (7.41)
TOTAL	37.5 (0.58)	32.5 (0.55)	27.5 (.50)	60 (0.60)	30 (0.52)	27.5 (.50)	22.5 (0.44)

Table 8.4 Confusion table of audio action classification with an HMM. Columns are original actions, rows are classified actions.

	SL	ST	RS	UP	DN	LT	RT	FW	BW	RL	RR
SL	**4**	0	0	0	0	0	0	0	0	0	0
ST	0	**5**	0	0	0	0	0	0	0	0	0
RS	0	0	**3**	0	0	0	0	0	0	0	0
UP	0	0	0	**4**	1	0	0	0	0	0	0
DN	0	0	0	0	**2**	0	0	0	0	0	0
LT	0	0	0	0	0	**3**	0	0	0	0	0
RT	0	0	0	0	0	0	**3**	0	0	0	1
FW	0	0	1	0	0	0	0	**5**	0	0	0
BW	0	0	0	0	0	0	0	0	**4**	0	0
RL	0	0	0	0	0	0	0	0	0	**0**	0
RR	0	0	0	0	0	0	0	0	0	2	**2**

Table 8.5 Confusion table of audio/video action classification under decision-level fusion.

	SL	ST	RS	UP	DN	LT	RT	FW	BW	RL	RR
SL	**4**	0	0	0	0	0	0	0	0	0	0
ST	0	**5**	0	0	0	0	0	0	0	0	0
RS	0	0	**3**	0	0	0	0	0	0	0	0
UP	0	0	0	**4**	0	0	0	0	0	0	0
DN	0	0	0	0	**3**	0	0	0	0	0	0
LT	0	0	0	0	0	**3**	0	0	0	0	0
RT	0	0	0	0	0	0	**3**	0	0	0	0
FW	0	0	1	0	0	0	0	**5**	1	0	0
BW	0	0	0	0	0	0	0	0	**3**	0	0
RL	0	0	0	0	0	0	0	0	0	**0**	0
RR	0	0	0	0	0	0	0	0	0	2	**3**

Table 8.6 Confusion table of audio/video action classification with unimodal HMMs.

	SL	ST	RS	UP	DN	LT	RT	FW	BW	RL	RR
ST	0	**4**	0	0	0	0	0	0	0	0	0
RS	0	0	**4**	0	0	0	0	0	0	0	0
UP	0	0	0	**4**	0	0	0	0	0	0	0
DN	0	0	0	0	**0**	0	0	0	0	0	0
LT	0	0	0	0	0	**0**	0	0	0	0	0
RT	0	0	0	0	0	0	**2**	0	0	0	0
FW	0	1	0	0	3	3	1	**5**	4	0	0
BW	0	0	0	0	0	0	0	0	**0**	0	0
RL	0	0	0	0	0	0	0	0	0	**0**	0
RR	0	0	0	0	0	0	0	0	0	2	**3**

Table 8.7 Confusion table of audio/video action classification with fixed-weight coupled HMM model. Columns are original actions, rows are classified actions.

	SL	ST	RS	UP	DN	LT	RT	FW	BW	RL	RR
SL	**4**	0	0	0	0	0	0	0	0	0	0
ST	0	**5**	0	0	0	0	0	0	0	0	0
RS	0	0	**4**	0	0	0	0	0	0	0	0
UP	0	0	0	**4**	3	0	0	0	0	0	0
DN	0	0	0	0	**0**	0	0	0	0	0	0
LT	0	0	0	0	0	**3**	0	0	0	0	0
RT	0	0	0	0	0	0	**3**	0	0	0	0
FW	0	0	0	0	0	0	0	**5**	0	0	0
BW	0	0	0	0	0	0	0	0	**4**	0	0
RL	0	0	0	0	0	0	0	0	0	**2**	1
RR	0	0	0	0	0	0	0	0	0	0	**2**

Table 8.8 Confusion table of audio/video action classification using coupled HMM with variable time-invariant weights.

	SL	ST	RS	UP	DN	LT	RT	FW	BW	RL	RR
SL	**4**	0	0	0	0	0	0	0	0	0	0
ST	0	**5**	0	0	0	0	0	0	0	0	0
RS	0	0	**4**	0	0	0	0	0	0	0	0
UP	0	0	0	**4**	3	0	0	0	0	0	0
DN	0	0	0	0	**0**	0	0	0	0	0	0
LT	0	0	0	0	0	**3**	0	0	0	0	0
RT	0	0	0	0	0	0	**3**	0	0	0	0
FW	0	0	0	0	0	0	0	**5**	0	0	0
BW	0	0	0	0	0	0	0	0	**4**	0	0
RL	0	0	0	0	0	0	0	0	0	**2**	1
RR	0	0	0	0	0	0	0	0	0	0	**2**

Table 8.9 Confusion table of audio/video action classification using coupled HMM with variable time-varying weights.

	SL	ST	RS	UP	DN	LT	RT	FW	BW	RL	RR
SL	**4**	0	1	0	0	0	0	0	0	0	0
ST	0	**5**	0	0	0	0	0	0	0	0	0
RS	0	0	**3**	0	0	0	0	0	0	0	0
UP	0	0	0	**4**	1	0	0	0	0	0	0
DN	0	0	0	0	**2**	0	0	0	0	0	0
LT	0	0	0	0	0	**3**	0	0	0	0	0
RT	0	0	0	0	0	0	**3**	0	0	0	0
FW	0	0	0	0	0	0	0	**5**	0	0	0
BW	0	0	0	0	0	0	0	0	**4**	0	0
RL	0	0	0	0	0	0	0	0	0	**2**	1
RR	0	0	0	0	0	0	0	0	0	0	**2**

Table 8.10 Confusion table of noisy audio action classification with an HMM.

	SL	ST	RS	UP	DN	LT	RT	FW	BW	RL	RR
SL	**3**	0	0	0	0	0	0	0	0	0	0
ST	0	**0**	0	0	0	0	0	0	0	0	0
RS	0	0	**4**	0	0	0	0	0	0	0	0
UP	0	0	0	**3**	0	0	0	0	0	0	0
DN	0	0	0	0	**0**	0	0	0	0	0	0
LT	1	4	0	1	3	**3**	0	0	3	0	0
RT	0	0	0	0	0	0	**3**	0	1	0	0
FW	0	0	0	0	0	0	0	**5**	0	0	0
BW	0	0	0	0	0	0	0	0	**0**	0	0
RL	0	1	0	0	0	0	0	0	0	**2**	1
RR	0	0	0	0	0	0	0	0	0	0	**2**

Table 8.11 Confusion table of noisy audio/video action classification under decision-level fusion.

	SL	ST	RS	UP	DN	LT	RT	FW	BW	RL	RR
SL	**4**	0	0	0	0	0	0	0	0	0	0
ST	0	**0**	0	0	0	0	0	0	0	0	0
RS	0	2	**4**	0	0	0	0	0	0	0	0
UP	0	0	0	**4**	3	0	0	0	0	0	0
DN	0	0	0	0	**0**	0	0	0	0	0	0
LT	0	0	0	0	0	**3**	0	0	0	0	0
RT	0	0	0	0	0	0	**3**	0	0	0	0
FW	0	2	0	0	0	0	0	**5**	2	0	0
BW	0	0	0	0	0	0	0	0	**2**	0	0
RL	0	1	0	0	0	0	0	0	0	**2**	1
RR	0	0	0	0	0	0	0	0	0	0	**2**

Table 8.12 Confusion table of noisy audio/video action classification with unimodal HMMs.

	SL	ST	RS	UP	DN	LT	RT	FW	BW	RL	RR
SL	**4**	4	4	0	0	0	0	0	1	0	0
ST	0	**0**	0	0	0	0	0	0	0	0	0
RS	0	0	**0**	0	0	0	0	0	0	0	0
UP	0	0	0	**4**	1	0	0	0	0	0	0
DN	0	0	0	0	**2**	0	0	0	0	0	0
LT	0	0	0	0	0	**0**	0	0	0	0	0
RT	0	1	0	0	0	0	**1**	0	0	0	1
FW	0	0	0	0	0	0	0	**3**	1	0	0
BW	0	0	0	0	0	0	0	0	**0**	0	0
RL	0	0	0	0	0	0	0	0	0	**0**	0
RR	0	0	0	0	0	3	2	2	2	2	**2**

Table 8.13 Confusion table of audio/video action classification with fixed-weight coupled HMM model. Columns are original actions, rows are classified actions.

	SL	ST	RS	UP	DN	LT	RT	FW	BW	RL	RR
SL	**4**	0	0	0	0	0	0	0	0	0	0
ST	0	**0**	0	0	0	0	0	0	0	0	0
RS	0	3	**4**	0	2	0	0	0	0	0	0
UP	0	0	0	**4**	1	0	0	0	0	0	0
DN	0	0	0	0	**0**	0	0	0	0	0	0
LT	0	1	0	0	0	**2**	0	0	0	0	0
RT	0	0	0	0	0	1	**3**	0	0	0	0
FW	0	1	0	0	0	0	0	**5**	2	0	0
BW	0	0	0	0	0	0	0	0	**2**	0	0
RL	0	0	0	0	0	0	0	0	0	**2**	1
RR	0	0	0	0	0	0	0	0	0	0	**2**

Table 8.14 Confusion table of noisy audio/video action classification using coupled HMM with variable time-invariant weights.

	SL	ST	RS	UP	DN	LT	RT	FW	BW	RL	RR
SL	**4**	0	0	0	0	0	0	0	0	0	0
ST	0	**0**	0	0	0	0	0	0	0	0	0
RS	0	4	**4**	0	0	0	0	0	0	0	0
UP	0	0	0	**4**	3	0	0	0	0	0	0
DN	0	0	0	0	**0**	0	0	0	0	0	0
LT	0	0	0	0	0	**3**	0	0	0	0	0
RT	0	0	0	0	0	0	**3**	0	0	0	0
FW	0	1	0	0	0	0	0	**5**	2	0	0
BW	0	0	0	0	0	0	0	0	**2**	0	0
RL	0	0	0	0	0	0	0	0	0	**2**	1
RR	0	0	0	0	0	0	0	0	0	0	**2**

Table 8.15 Confusion table of noisy audio/video action classification using coupled HMM with variable time-varying weights.

	SL	ST	RS	UP	DN	LT	RT	FW	BW	RL	RR
SL	**4**	1	1	0	0	0	0	0	0	0	0
ST	0	**1**	0	0	0	0	0	0	0	0	0
RS	0	1	**3**	0	0	0	0	0	0	0	0
UP	0	0	0	**4**	2	0	0	0	0	0	0
DN	0	0	0	0	**1**	0	0	0	0	0	0
LT	0	0	0	0	0	**3**	0	0	0	0	0
RT	0	0	0	0	0	0	**3**	0	0	0	0
FW	0	0	0	0	0	0	0	**5**	0	0	0
BW	0	2	0	0	0	0	0	0	**4**	0	0
RL	0	0	0	0	0	0	0	0	0	**1**	1
RR	0	0	0	0	0	0	0	0	0	1	**2**

CHAPTER 9

CONCLUSIONS

Rapid development of novel computing, communication, and display technologies has brought into focus the inadequacies of existing HCI techniques. Keyboards and mice remain the major HCI modes just as ten years ago, yet the information flow between the user and computer has grown tremendously. Thus in recent years there has been a surge of interest in novel modalities for HCI that will potentially resolve the interaction bottleneck. Despite the abundance of interaction devices, the level of naturalness and efficiency of HCI has remained low. This is due in particular to the lack of robust sensory data interpretation techniques. For instance, automatic speech recognition (ASR) still performs satisfactorily only in highly restrictive, single-user, low-noise setups. Natural gesture interpretation is confined to its infancy. Moreover, even though it is clear that natural human-to-human interaction involves multiple communication modalities, current HCI narrowly focuses on interpretation of single sensing modalities. Potential benefits of multimodal interaction are numerous: reinforcement of an individual mode's interpretations, substitution of missing or unreliable HCI modes, etc. However, joint interpretation of multiple modalities is not a trivial task.

This dissertation presents a probabilistic approach to analysis and interpretation of data acquired by computer sensory modalities. The approach is based on modeling of user's intentions, actions, and their realizations using the framework of *dynamic Bayesian networks* or DBNs. Dynamic Bayesian networks are a generalization of the successful statistical time-series models such as hidden Markov models, commonly found in ASR, and Kalman filters, the essential tools of dynamic systems. Hence, it is natural to consider the DBN models as a basis of the general spatio-temporal action interpretation task. The framework of Bayesian networks provides one with a mathematically rigorous foundation for model learning and data classification in light of probabilistic Bayesian inference. It also gives one a powerful set of tools and techniques to generalize from and build upon the basic models. Three cases of novel DBN-inspired complex temporal models were introduced in this dissertation: *mixtures of DBNs* in Chapter 4, *mixed-state DBNs* in Chapter 5, and *coupled HMMs* in Chapter 6.

A mixture of dynamic Bayesian networks can describe a set of dynamic systems that all draw their observations from a common pool of data. This situation occurs if one attempts to infer the motion of several objects, whose appearances are all captured in a single image frame. The tasks of associating every observations from the pool with an object and inferring the object's state were solved in this dissertation from the perspective of Bayesian inference. Moreover, the same perspective yields learning of the model parameters a well-defined and easy task.

Mixed-state DBNs are introduced to bridge the gap between discrete- and continuous-state dynamic models. For instance, HMMs have excelled greatly in modeling discrete language concepts behind natural processes such as speech. Continuous-state dynamic systems, on the other hand, are the basic descriptors of the physical systems employed in production and perception of speech or hand movements—the carriers of concepts. Until now, the modeling of the concepts and physical systems has largely remained separated. Under the auspices of the dynamic Bayesian network framework in Chapter 5, this dissertation formulated the *mixed-state DBN model*, a unification of the concept HMMs and dynamic systems. Such networks yield an optimal solution for joint modeling of temporal events whose concepts remain highly coupled with the physical systems.

Finally, DBNs allow us to model the concepts that are expressed and sensed through a number of different but concurrent modalities. With that in mind three coupled DBN architectures were established in Chapter 6. Coupled HMMs are suitable for modeling of multiple modality observations coming from different concepts with varying levels of coupling. As before, Bayesian network inference provides an elegant framework to efficiently infer from multimodal data what the underlying concepts are. Moreover, the same framework yields adapted parameters that accurately model the level of interaction among the concepts.

Experimental validation of the proposed approaches was done in a setup for control of a virtual display. Attention was focused on modeling of visually perceived free hand gestures and speech. Hand gestures and speech are often used in conjunction in natural human-to-human communication. A combination of the *mixture of DBNs* and *mixed-state networks* was used in Chapter 7 to describe concepts and realizations of visually perceived hand gestures. We obtained encouraging results in the domain of restricted unimodal gesture interpretation. Interpretation of a small gestural command set with this model achieved recognition rates of close to 80 percent, an improvement of 8 to 20 percent over standard techniques. Furthermore, the same model was used for concurrent estimation of hand motion dynamics, a crucial step in gestural analysis. By doing so we introduced the constraints of higher-level knowledge, usually only present in the recognition phase, to dynamic system state estimation.

Finally, we considered the role of DBNs in multimodal interpretation within the HCI context. Classical unimodal concept models, such as HMMs when applied directly to multiple observation domains, perform satisfactorily if the concepts behind multiple observations are either highly independent or very dependent. Coupling between speech and gestures, two most expressive communication modalities, can be efficiently tackled within the *coupled HMM* framework. Experiments described in Chapter 8 confirmed that, compared to the extremes of highly coupled and highly decoupled concept models, adaptive coupled HMM structures perform favorably. This is particularly emphasized in situations where neither of the two modalities is dominant. Whereas independent classification of speech and gestures yielded misclassification rates in the vicinity of 35 percent the coupled inference brought them down to 22.5 percent. Of course, the price to pay was not always small. Adaptive coupled methods add a computational burden of iterative inference that can be overwhelming. In essence, they require that the decoupled inference be repeated a number of times. In my experiments this only meant a five-fold increase in complexity. More practical situations, however, may require that the trade-off between computational complexity and recognition performance be evaluated on a case-by-case basis.

9.1 Future Work

The use of Bayesian networks for modeling and interpretation of spatio-temporal actions, especially computer-sensed human communication modalities, is well-founded and general enough to open the door to a variety of possible extensions. This dissertation addressed some initial aspects and introduced basic models of complex communication processes. Future work can be carried on with the same DBN philosophy in mind. In particular, two possible avenues can be taken.

One may focus on improved modeling of visually observed hand motions. The model presented in this work in Chapter 7 simplifies the motion of the arm and the hand to the one of an independent Newtonian object—the hand itself. To model an articulated structure such as the arm one can construct a mixture-of-DBN model where the dynamic systems influence each other. For example, one ("root") dynamic system can be devoted to modeling the motion and appearance of the human torso whereas the other dynamic systems can model the motion and shape of the upper and the lower arm with respect to the torso and each other. Equivalent models have in fact been presented lately in some computer vision literature, but without the rigorous mathematical framework that DBN offers. In addition, as presented in Chapter 7, the DBN framework allows one not only to specify generic motion dynamics but also to include the models of concepts that drive those dynamics. This can be used to model not only the appearance of communicative concepts but also a number of natural human actions such as running, walking, etc.

Another avenue of work stems from the modeling of coupled communication concepts. Coupled HMMs of Chapter 6 in the form presented in this dissertation emphasize coupling of fairly low-level concept states. Nonetheless, the same architecture can be utilized to describe interaction of modes on a higher, possibly semantic level. This can of course be applied to more general speech/gesture language modeling. On the other hand, the tools of approximate analysis of coupled HMMs can serve as a basis for a whole suite of similar models. For instance, some of those models can address the interacting modalities that occur at naturally different sampling rates. Finally, the merger of multiple coupled hidden Markov modeled concepts with the mixed-state DBNs has enormous future potential deserving another thorough look.

Conclusions and tools of this study are general enough to be used by future researchers as a basis for novel probabilistic information fusion architectures. The author hopes that this study serves as an important stepping stone in the quest for a thorough mathematical foundation behind the ever practical HCI.

APPENDIX A

VARIATIONAL INFERENCE THEOREM

Consider the following form of distribution Q:

$$Q(\mathcal{X}) = \frac{e^{-H_Q(\mathcal{X},\mathcal{Y}|\eta)}}{Z_Q}, \tag{A.1}$$

where H_Q is the Hamiltonian of the distribution, and Z_Q is the normalization factor chosen such that $\sum_{\mathcal{X}} Q(\mathcal{X}) = 1$. Furthermore, let H_Q be defined as

$$H_Q(\mathcal{X},\mathcal{Y}|\eta) = -\sum_i f_i(\mathcal{X},\mathcal{Y}) g_i(\eta). \tag{A.2}$$

In other words, consider a particular family of exponential distributions Q whose sufficient statistics are $\langle f_i \rangle$. Most of the commonly encountered distributions (Gaussian, etc.) belong to this family. For instance, for a normally distributed x with mean μ and variance Σ, the Hamiltonian is

$$H_Q(x|\mu,\Sigma) = \frac{1}{2}\left[\log(2\pi) + \log(|\Sigma|) + (x-\mu)'\Sigma^{-1}(x-\mu)\right]. \tag{A.3}$$

The goal of variational inference is to minimize the Kullback–Leibler (KL) divergence $D(Q||P)$ by varying the parameters η. Formally,

$$\eta^* \;=\; \arg\min_{\eta} \sum_{\mathcal{X}} Q(\mathcal{X}|\eta) \log \frac{P(\mathcal{X}|\mathcal{Y})}{Q(\mathcal{X}|\eta)}. \tag{A.4}$$

The following theorem from Ghahramani [56] leads to a set of necessary conditions that η has to satisfy in order to minimize the KL divergence between the distributions.

Theorem 1 *For any distribution $P(\mathcal{X}|\mathcal{Y})$ defined over a set of variables $\mathcal{X}$, where $H(\mathcal{X},\mathcal{Y})$ is defined so that*

$$P(\mathcal{X}) = \frac{1}{Z} e^{-H(\mathcal{X},\mathcal{Y})}$$

and any approximating distribution Q in the exponential family parameterized by η, the KL divergence $D(Q||P)$ can be minimized by iteratively solving the set of following fixed point equations:

$$\frac{\partial \langle H_Q(\mathcal{X},\mathcal{Y})\rangle}{\partial \langle f_i(\mathcal{X},\mathcal{Y})\rangle} - \frac{\partial \langle H(\mathcal{X},\mathcal{Y})\rangle}{\partial \langle f_i(\mathcal{X},\mathcal{Y})\rangle} = 0, \ \forall i, \tag{A.5}$$

where $\langle\cdot\rangle$ is taken over the approximating distribution Q.

Proof. From the definition of KL divergence it follows that

$$\begin{aligned} D(Q||P) &= \sum_{\mathcal{X}} Q(\mathcal{X}) \log \frac{P(\mathcal{X})}{Q(\mathcal{X})} \\ &= \langle H(\mathcal{X},\mathcal{Y})\rangle - \langle H_Q(\mathcal{X},\mathcal{Y})\rangle - \log Z_Q + \log Z. \end{aligned} \tag{A.6}$$

To minimize D one first needs to find its derivative with respect to parameter η:

$$\frac{dD}{d\eta} = \frac{d\langle H(\mathcal{X},\mathcal{Y})\rangle}{d\eta} - \frac{d\langle H_Q(\mathcal{X},\mathcal{Y})\rangle}{d\eta} - \frac{d\log Z_Q}{d\eta}.$$

We now look specifically at each of the three terms in this expression. Given the form of distribution Q from Equation A.2 the terms can be expanded as

$$\frac{d\langle H\rangle}{d\eta} = \sum_i \frac{\partial\langle H\rangle}{\partial\langle f_i\rangle}\frac{d\langle f_i\rangle}{d\eta}$$

$$\begin{aligned} \frac{d\langle H_Q\rangle}{d\eta} &= \frac{d}{d\eta}\left[-\sum_{\mathcal{X}}\sum_i f_i(\mathcal{X},\mathcal{Y})g_i(\eta)Q(\mathcal{X})\right] \\ &= -\sum_{\mathcal{X}}\sum_i f_i(\mathcal{X},\mathcal{Y})\left[\frac{dg_i(\eta)}{d\eta}Q(\mathcal{X}) + g_i(\eta)\frac{dQ(\mathcal{X})}{d\eta}\right] \\ &= -\sum_i \frac{dg_i(\eta)}{d\eta}\langle f_i(\mathcal{X},\mathcal{Y})\rangle - \sum_i g_i(\eta)\frac{d}{d\eta}\langle f_i(\mathcal{X},\mathcal{Y})\rangle \end{aligned}$$

$$\begin{aligned} \frac{d\log Z_Q}{d\eta} &= \frac{1}{Z_Q}\frac{d}{d\eta}\left(\sum_{\mathcal{X}} e^{-H_Q(\mathcal{X},\mathcal{Y})}\right) \\ &= -\frac{1}{Z_Q}\sum_{\mathcal{X}} e^{-H_Q(\mathcal{X},\mathcal{Y})}\frac{dH_Q(\mathcal{X},\mathcal{Y})}{d\eta} \\ &= \sum_i\sum_{\mathcal{X}} Q(\mathcal{X})\frac{dg_i(\eta)}{d\eta}f_i(\mathcal{X},\mathcal{Y}) \\ &= \sum_i \frac{dg_i(\eta)}{d\eta}\langle f_i(\mathcal{X},\mathcal{Y})\rangle. \end{aligned}$$

Combining the three terms together yields

$$\begin{aligned} \frac{dD}{d\eta} &= \sum_i\left(g_i(\eta) + \frac{\partial\langle H\rangle}{\partial\langle f_i\rangle}\right)\frac{d\langle f_i\rangle}{d\eta} \\ &= \sum_i\left(\frac{\partial\langle H_Q\rangle}{\partial\langle f_i\rangle} - \frac{\partial\langle H\rangle}{\partial\langle f_i\rangle}\right)\frac{d\langle f_i\rangle}{d\eta}. \end{aligned}$$

Finally, setting $dD/d\eta$ to zero leads to expression in Equation A.5. □

Note (i): The solution of the KL divergence minimization satisfies the expectation step of GEM algorithm in Section 2.3. Clearly,

$$\frac{dB(P,Q,\theta)}{d\eta} = \frac{dD}{d\eta} = \frac{d\langle H(\mathcal{X},\mathcal{Y})\rangle}{d\eta} - \frac{\langle H_Q(\mathcal{X},\mathcal{Y})\rangle}{d\eta} - \frac{d\log Z_Q}{d\eta}.$$

Hence, variational inference (in general) provides an optimal solution for the expectation step of GEM.

Note (ii): Theorem 1 also holds in the case of deterministic annealing variant of GEM. Distribution Q in expressions $\langle \cdot \rangle$ is, nevertheless, raised to the power of $\beta = 1/T_{\text{anneal}}$.

APPENDIX B

MIXTURE OF DYNAMIC BAYESIAN NETWORKS

From the definitions of the joint and factorized pdf Hamiltonians in Equations 4.3 and 4.4 it follows that

$$\begin{aligned}
&\langle H - H_Q \rangle = \\
&\quad \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \sum_{t=0}^{T-1} \left(\left\langle s_t^{(m)}(n) \right\rangle - h_t^{(m)}(n) \right) \left\langle \log Pr(y_t^{(m)} | x_t^{(n)}) \right\rangle \\
&\quad + \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \sum_{t=0}^{T-1} \left\langle s_t^{(m)}(n) \right\rangle \log q_t^{(m)}(n). \qquad \text{(B.1)}
\end{aligned}$$

To minimize the divergence between the original pdf P and the approximating pdf Q, according to Theorem 1, partial derivatives of $\langle H - H_Q \rangle$ with respect to the sufficient statistics of Q need to vanish. Sufficient statistics of distribution Q are $\left\langle s_t^{(m)}(n) \right\rangle$ and the statistics determined by the observation distribution $Pr(y_t^{(m)} | x_t^{(n)})$. However, regardless of what the observation pdf sufficient statistics are, the quantity $\langle H - H_Q \rangle$ vanishes by choosing

$$h_t^{(m)}(n) = \left\langle s_t^{(m)}(n) \right\rangle = Pr(s_t(m) = n).$$

On the other hand, partial derivative of $\langle H - H_Q \rangle$ with respect to $\left\langle s_t^{(m)}(n) \right\rangle$ is

$$\frac{\partial \langle H - H_Q \rangle}{\partial \left\langle s_t^{(m)}(n) \right\rangle} = \log q_t^{(m)}(n) - \left\langle \log Pr(y_t^{(m)} | x_t^{(n)}) \right\rangle .$$

Equating the above quantity with zero yields

$$q_t^{(m)}(n) = \exp \left\langle \log \left(Pr(y_t^{(m)} | x_t^{(n)}) \right) \right\rangle .$$

APPENDIX C

MIXED-STATE DYNAMIC BAYESIAN NETWORKS

Given the definitions of the coupled DBN Hamiltonian from Equation 5.8 and the approximating DBN Hamiltonian from Equation 5.11 and Theorem 1 of variational inference parameters from Section 2.2.3, one finds

$$\begin{aligned}\langle H - H_Q \rangle = & \sum_{t=1}^{T-1} \left(\langle x_t \rangle - A \langle x_{t-1} \rangle \right)' Q^{-1} \left(u_t - D \langle s_t \rangle \right) \\ & + \frac{1}{2} \sum_{t=1}^{T-1} \operatorname{tr} \left\{ D' Q^{-1} D \langle s_t s_t' \rangle \right\} - \frac{1}{2} \sum_{t=1}^{T-1} \left(u_t \right)' Q^{-1} \left(u_t \right) \\ & + \langle x_0 \rangle Q_0^{-1} \left(u_0 - D \langle s_0 \rangle \right) + \frac{1}{2} \operatorname{tr} \left\{ D_0' Q_0^{-1} D_0 \langle s_0 s_0' \rangle \right\} - \frac{1}{2} \left(u_0 \right)' Q_0{}^{-1} \left(u_0 \right) \\ & + \sum_{t=0}^{T-1} \langle s_t \rangle' \log q_t. \end{aligned} \tag{C.1}$$

Derivatives of the above quantity with respect to the sufficient statistics of Q, $\langle x_\tau \rangle$ and $\langle s_\tau \rangle$[1], are

$$\frac{\partial \langle H - H_Q \rangle}{\partial \langle x_\tau \rangle} = \begin{cases} Q^{-1} \left(u_\tau - D \langle s_\tau \rangle \right) & \tau = T-1 \\ Q^{-1} \left(u_\tau - D \langle s_\tau \rangle \right) + A' Q^{-1} \left(u_{\tau+1} - D \langle s_{\tau+1} \rangle \right) & 0 < \tau < T-1 \\ Q_0^{-1} \left(u_\tau - D \langle s_\tau \rangle \right) + A' Q^{-1} \left(u_{\tau+1} - D \langle s_{\tau+1} \rangle \right) & \tau = 0 \end{cases} \tag{C.2}$$

and

$$\frac{\partial \langle H - H_Q \rangle}{\partial \langle s_\tau(i) \rangle} = \begin{cases} -d_i' Q^{-1} \left(\langle x_\tau \rangle - A \langle x_{\tau-1} \rangle \right) + \frac{1}{2} d_i' Q^{-1} d_i + \log q_\tau(i) & 0 < \tau < T \\ -d_{0i} Q_0^{-1} \langle x_0 \rangle + \frac{1}{2} d_0{}_i' Q_0^{-1} d_{0i} + \log q_0(i) & \tau = 0 \end{cases} . \tag{C.3}$$

Setting the above partial derivatives to zero results in the fixed-point variational parameter equations of Section 5.3.2.

[1] Derivatives of $\left\langle H - H_Q \right\rangle$ with respect to $\langle x_\tau x_\tau' \rangle$, $\left\langle x_\tau x_{\tau-1}' \right\rangle$, and $\left\langle s_\tau s_{\tau-1}' \right\rangle$ are zero

APPENDIX D

COUPLED HIDDEN MARKOV MODELS

D.1 Fixed-Weights Coupled Hidden Markov Model

To find the optimal values of variational parameters for the fixed-weight factorized model, one employs Theorem 1. The mean difference of Hamiltonians of the original and the approximating network is

$$\begin{aligned} &\langle H - H_Q \rangle = \\ &\quad -\sum_{t=1}^{T-1} \sum_{n=0}^{M-1} \sum_{m=0, m\neq n}^{M-1} w^{(n)}(m) \left\langle s_t^{(n)} \right\rangle' \log P^{(n,m)} \left\langle s_{t-1}^{(m)} \right\rangle \\ &\quad +\sum_{t=0}^{T-1} \sum_{n=0}^{M-1} \left\langle s_t^{(n)} \right\rangle' \log q_t^{(n)}. \end{aligned}$$

Sufficient statistics of each of the HMM submodels $Q_q^{(l)}$ are $\left\langle s_\cdot^{(l)} \right\rangle$ and $\left\langle s_\cdot^{(l)} s_{\cdot-1}^{(l)}{}' \right\rangle$. Taking a partial derivative of the mean Hamiltonian difference with respect to $\left\langle s_\tau^{(l)} \right\rangle$ yields

$$\frac{\partial \langle H - H_Q \rangle}{\partial \left\langle s_\tau^{(l)} \right\rangle} = \begin{cases} -\sum_{m=0,m\neq l}^{M-1} w^{(m)}(l) \log P^{(m,l)\prime} \left\langle s_1^{(m)} \right\rangle + \log q_\tau^{(l)} & \tau = 0 \\ -\sum_{m=0,m\neq l}^{M-1} w^{(l)}(m) \log P^{(l,m)} \left\langle s_{\tau-1}^{(m)} \right\rangle & \\ -\sum_{m=0,m\neq l}^{M-1} w^{(m)}(l) \log P^{(m,l)\prime} \left\langle s_{\tau+1}^{(m)} \right\rangle + \log q_\tau^{(l)} & 0 < \tau < T-1 \\ -\sum_{m=0,m\neq l}^{M-1} w^{(l)}(m) \log P^{(l,m)} \left\langle s_{T-2}^{(m)} \right\rangle + \log q_\tau^{(l)} & \tau = T-1. \end{cases} \tag{D.1}$$

The partial derivative with respect to $\left\langle s_\tau^{(l)} s_{\tau-1}^{(l)}{}' \right\rangle$ is always zero, for any τ.

D.2 Adaptive-Weights Coupled Hidden Markov Model

The mean difference of Hamiltonians of the original distribution P defined in Equation 6.10 and the approximating distribution Q defined in Equation 6.11 is

$$\langle H - H_Q \rangle =$$

$$\begin{aligned}
&\sum_{t=1}^{T-1}\sum_{n=0}^{M-1}\Big[(c^{(n)}-\left\langle w^{(n)}(n)\right\rangle \operatorname{tr}\left\{\log P^{(n,n)}\left\langle s_{t-1}^{(n)}s_t^{(n)\prime}\right\rangle\right\}\\
&-\sum_{m=0,m\neq n}^{M-1}\left\langle w^{(n)}(m)\right\rangle\left\langle s_t^{(n)\prime}\right\rangle\log P^{(n,m)}\left\langle s_{t-1}^{(m)}\right\rangle\Big]\\
&+\sum_{t=0}^{T-1}\sum_{n=0}^{M-1}\left\langle s_t^{(n)}\right\rangle'\log q_t^{(n)}\\
&+(T-1)\sum_{n=0}^{M-1}\left\langle w^{(n)}\right\rangle'\log r^{(n)}.
\end{aligned}$$

Partial derivatives with respect to sufficient statistics of the distribution Q are

$$\frac{\partial\left\langle H-H_Q\right\rangle}{\partial\left\langle s_\tau^{(l)}s_{\tau-1}^{(l)}{}'\right\rangle}=c^{(n)}-\left\langle w^{(n)}(n)\right\rangle, \tag{D.2}$$

$$\frac{\partial\left\langle H-H_Q\right\rangle}{\partial\left\langle s_\tau^{(l)}\right\rangle}=\begin{cases}
-\sum_{m=0,m\neq l}^{M-1}\left\langle w^{(m)}(l)\right\rangle\log P^{(m,l)\prime}\left\langle s_1^{(m)}\right\rangle+\log q_\tau^{(l)} & \tau=0\\
-\sum_{m=0,m\neq l}^{M-1}\left\langle w^{(l)}(m)\right\rangle\log P^{(l,m)}\left\langle s_{\tau-1}^{(m)}\right\rangle & \\
-\sum_{m=0,m\neq l}^{M-1}\left\langle w^{(m)}(l)\right\rangle\log P^{(m,l)\prime}\left\langle s_{\tau+1}^{(m)}\right\rangle+\log q_\tau^{(l)} & 0<\tau<T-1\\
-\sum_{m=0,m\neq l}^{M-1}\left\langle w^{(l)}(m)\right\rangle\log P^{(l,m)}\left\langle s_{T-2}^{(m)}\right\rangle+\log q_\tau^{(l)} & \tau=T-1
\end{cases} \tag{D.3}$$

and

$$\frac{\partial\left\langle H-H_Q\right\rangle}{\partial\left\langle w^{(l)}(k)\right\rangle}=-\sum_{t=1}^{T-1}\operatorname{tr}\left\{\left\langle s_t^{(l)}s_{t-1}^{(k)}{}'\right\rangle'\log P^{(l,k)}\right\}+(T-1)\log r^{(l)}(k). \tag{D.4}$$

Partial derivatives with respect to the remaining sufficient statistics are zero.

D.3 Adaptive Time-Varying Coupled Hidden Markov Weights Model

The mean difference of Hamiltonians in the time-varying model takes a form similar to that of the time-invariant case:

$$\begin{aligned}
&\left\langle H-H_Q\right\rangle=\\
&\sum_{t=1}^{T-1}\sum_{n=0}^{M-1}\Big[(c_t^{(n)}-\left\langle w_t^{(n)}(n)\right\rangle \operatorname{tr}\left\{\log P^{(n,n)}\left\langle s_{t-1}^{(n)}s_t^{(n)\prime}\right\rangle\right\}\\
&-\sum_{m=0,m\neq n}^{M-1}\left\langle w_t^{(n)}(m)\right\rangle\left\langle s_t^{(n)\prime}\right\rangle\log P^{(n,m)}\left\langle s_{t-1}^{(m)}\right\rangle\Big]\\
&+\sum_{t=0}^{T-1}\sum_{n=0}^{M-1}\left\langle s_t^{(n)}\right\rangle'\log q_t^{(n)}\\
&+\sum_{t=0}^{T-1}\sum_{n=0}^{M-1}\left\langle w_t^{(n)}\right\rangle'\log r_t^{(n)}.
\end{aligned}$$

Partial derivatives with respect to the sufficient statistics of pdf Q which are not identically zero are given as

$$\frac{\partial \langle H - H_Q \rangle}{\partial \left\langle s_\tau^{(l)} s_{\tau-1}^{(l)}{}' \right\rangle} = c_t^{(n)} - \left\langle w_t^{(n)}(n) \right\rangle , \tag{D.5}$$

$$\frac{\partial \langle H - H_Q \rangle}{\partial \left\langle s_\tau^{(l)} \right\rangle} = \begin{cases} -\sum_{m=0,m\neq l}^{M-1} \left\langle w_1^{(m)}(l) \right\rangle \log P^{(m,l)}{}' \left\langle s_1^{(m)} \right\rangle + \log q_\tau^{(l)} & \tau = 0 \\ -\sum_{m=0,m\neq l}^{M-1} \left\langle w_\tau^{(l)}(m) \right\rangle \log P^{(l,m)} \left\langle s_{\tau-1}^{(m)} \right\rangle & \\ -\sum_{m=0,m\neq l}^{M-1} \left\langle w_{\tau+1}^{(m)}(l) \right\rangle \log P^{(m,l)}{}' \left\langle s_{\tau+1}^{(m)} \right\rangle + \log q_\tau^{(l)} & 0 < \tau < T-1 \\ -\sum_{m=0,m\neq l}^{M-1} \left\langle w_{T-1}^{(l)}(m) \right\rangle \log P^{(l,m)} \left\langle s_{T-2}^{(m)} \right\rangle + \log q_\tau^{(l)} & \tau = T-1 \end{cases} \tag{D.6}$$

and

$$\frac{\partial \langle H - H_Q \rangle}{\partial \left\langle w_\tau^{(l)}(k) \right\rangle} = -\mathrm{tr} \left\{ \left\langle s_t^{(l)} s_{t-1}^{(k)}{}' \right\rangle{}' \log P^{(l,k)} \right\} + \log r_\tau^{(l)}(k). \tag{D.7}$$

REFERENCES

[1] J. A. Adam, "Virtual reality," *IEEE Spectrum*, vol. 30, no. 10, pp. 22–29, 1993.

[2] A. G. Hauptmann and P. McAvinney, "Gesture with speech for graphics manipulation," *Int'l Journal Man-Machine Studies*, vol. 38, pp. 231–249, Feb. 1993.

[3] H. Rheingold, *Virtual Reality.* New York: Summit Books, 1991.

[4] S. Mann, "Wearable computing: A first step toward personal imaging," *IEEE Computer*, vol. 30, pp. 25–32, Feb. 1997.

[5] L. R. Rabiner and B. Juang, *Fundamentals of Speech Recognition.* Englewood Cliffs, NJ: Prentice Hall, 1993.

[6] S. Oviatt, A. DeAngeli, and K. Kuhn, "Integration and synchronization of input modes during multimodal human-computer interaction," Center for Human-Computer Communication, Oregon Graduate Institute, 1997, http://www.cse.ogi.edu/CHCC.

[7] A. Waibel, M. T. Vo, P. Duchnowski, and S. Manke, "Multimodal interfaces," *Artificial Intelligence Review*, vol. 10, pp. 299–319, Aug. 1995.

[8] B. A. Myers, "A brief history of human computer interaction technology," Human Computer Interaction Institute, School of Computer Science, Carnegie Mellon University, Tech. Rep. CMU-CS-TR-96-163, 1996

[9] F. H. Raab, E. B. Blood, T. O. Steiner, and H. R. Jones, "Magnetic position and orientation tracking system," *IEEE Trans. on Aerospace and Electronic Systems*, pp. 709–718, 1979.

[10] R. Azuma, "Tracking requirements for augmented reality," *Communications of the ACM*, vol. 36, no. 7, pp. 50–52, 1993.

[11] T. Baudel and M. Baudouin-Lafon, "Charade: Remote control of objects using free-hand gestures," *Communications of the ACM*, vol. 36, no. 7, pp. 28–35, 1993.

[12] S. S. Fels and G. E. Hinton, "Glove-Talk: A neural network interface between a Data-Glove and a speech synthesizer," *IEEE Transactions on Neural Networks*, vol. 4, pp. 2–8, Jan. 1993.

[13] D. L. Quam, "Gesture recognition with a DataGlove," in *Proc. the 1990 IEEE National Aerospace and Electronics Conf.*, vol. 2, 1990, pp. 27–30.

[14] D. J. Sturman and D. Zeltzer, "A survey of glove-based input," *IEEE Computer Graphics and Applications*, vol. 14, pp. 30–39, Jan. 1994.

[15] C. Wang and D. J. Cannon, "A virtual end-effector pointing system in point-and-direct robotics for inspection of surface flaws using a neural network based skeleton transform," in *Proc. IEEE Int'l Conf. on Robotics and Automation*, vol. 3, 1993, pp. 784–789.

[16] B. H. Juang, "Speech recognition in adverse environments," *Computer Speech and Language*, vol. 5, pp. 275–294, 1991.

[17] D. Stork and H.-L. Lu, "Speechreading by Boltzmann zippers," in *Machines that Learn*, 1996, pp. 156–177.

[18] R. Sharma, T. S. Huang, V. I. Pavlović, Y. Zhao, Z. Lo, S. Chu, K. Schulten, A. Dalke, J. Phillips, M. Zeller, and W. Humphrey, "Speech/gesture interface to a visual computing environment for molecular biologists," in *Proc. Int'l Conf. on Pattern Recognition*, 1996, pp. 322–326.

[19] V. I. Pavlovic, R. Sharma, and T. S. Huang, "Visual interpretation of hand gestures for human-computer interaction: A review," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 732–756, July 1997.

[20] F. Hatfield, E. A. Jenkins, M. W. Jennings, and G. Calhoun, "Principles and guidelines for the design of eye/voice interaction dialogs," in *Proc. the Third Annual Symposium on Human Interaction with Complex Systems*, 1996, pp. 10–19.

[21] T. E. Hutchinson, "Computers that sense eye position on the display," *Computer*, vol. 26, pp. 65–67, July 1993.

[22] R. J. K. Jacob, "What you look at is what you get," *Computer*, vol. 26, pp. 65–67, July 1993.

[23] I. A. Essa and A. P. Pentland, "Coding analysis, interpretation, and recognition of facial expressions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 757–763, July 1997.

[24] D. M. Gavrila and L. S. Davis, "Towards 3-d model-based tracking and recognition of human movement: a multi-view approach," in *Proc. Int'l Workshop Automatic Face and Gesture Recognition*, 1995, pp. 272–277.

[25] C. R. Wren, A. Azarbayejani, T. Darrell, and A. P. Pentland, "Pfinder: Real-time tracking of the human body," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 780–785, July 1997.

[26] V. I. Pavlovic, R. Sharma, and T. S. Huang, "Gestural interface to a visual computing environment for molecular biologists," in *Proc. Int'l Conf. Automatic Face and Gesture Recognition*, 1996, pp. 30–35.

[27] R. Sharma and J. Molineros, "Computer vision-based augmented reality for guiding manual assembly," *Presence: Teleoperators and Virtual Environments*, vol. 6, pp. 292–317, June 1997.

[28] F. K. H. Quek, "Eyes in the interface," *Image and Vision Computing*, vol. 13, pp. 78–91, Aug. 1995.

[29] R. Sharma, T. S. Huang, and V. I. Pavlović, "A multimodal framework for interacting with virtual environments," in *Human Interaction with Complex Systems,* C. A. Ntuen, E. H. Park, and J. H. Kim, Eds.., Kluwer Academic Publishers, 1996, pp. 53–71.

[30] C. Lansing and G. W. McConkie, "A new method for speechreadng research: Tracking observers' eye movements," *Journal of the Academy of Rehabilitative Audiology*, vol. 28, pp. 25–43, 1994.

[31] R. M. Satava and S. B. Jones, "Virtual environments for medical training and education," *PRESENCE: Teleoperators and Virtual Environments*, vol. 6, no. 2, pp. 139–146, 1997.

[32] S. L. Delp, P. Loan, C. Basdogan, and J. M. Rosen, "Surgical simulation: An emerging technology for training in emergency medicine," *PRESENCE: Teleoperators and Virtual Environments*, vol. 6, no. 2, pp. 147–159, 1997.

[33] M. Bergamasco, "Haptic interfaces: The study of force and tactile feedback systems," in *Proc. IEEE Int'l Workshop on Robot on Robot and Human Communication*, 1995, pp. 15–20.

[34] T. R. Sheridan, *Telerobotics, Automation, and Human Supervisory Control.* Cambridge, MA: MIT Press, 1992.

[35] Z. A. Keirn and J. I. Aunon, “Man-machine communications through brain-wave processing,” *IEEE Engineering in Medicine and Biology Magazine*, vol. 9, pp. 55–57, 1990.

[36] D. J. McFarland, G. W. Neat, R. F. Read, and J. R. Wolpaw, “An EEG-based method for graded cursor control,” *Psychobiology*, vol. 21, pp. 77–81, 1993.

[37] V. T. Nasman, G. L. Calhoun, and G. R. McMillan, “Brain-actuated control and hmds,” in *Head Mounted Displays, Optical & Electro-optical Engineering,* New York: McGraw Hill, 1997, pp. 285–312.

[38] W. Putnam and R. B. Knapp, “Real-time computer control using pattern recognition of the electromyograms,” in *Proc. 15th Annual Int’l Conf. Engineering in Medicine and Biology Society*, vol. 15, 1993, pp. 1236–1237.

[39] H. S. Lusted and R. B. Knapp, “Controlling computers with neural signals,” *Scientific American*, pp. 82–87, Oct. 1996.

[40] T. Elbert, B. Rockstroh, W. Lutzenberger, and W. Birbaumer, *Self-Regulation of the Brain and Behavior.* New York: Springer-Verlag, 1984.

[41] S. Suryanarayanan and N. R. Reddy, “EMG-based interface for position tracking and control in VR environments and teleoperation,” *PRESENCE: Teleoperators and Virtual Environments*, vol. 6, no. 3, pp. 282–291, 1997.

[42] A. M. Junker, J. H. Schnurer, D. F. Ingle, and C. W. Downey, “Loop-closure of the visual cortex response,” Armstrong Aerospace Medical Research Laboratory, Wright-Patterson Air Force Base, OH, Tech. Rep. AAMRL-TR-88-014, 1988.

[43] D. W. Patmore and R. B. Knapp, “A cursor controller using evoked potentials and EOG,” in *Proc. RESNA Eighteen Annual Conf.*, 1995, pp. 702–704.

[44] Z. Ghahramani, “Learning dynamic Bayesian networks,” in *Adaptive processing of temporal information, Lecture notes in artificial intelligence,* C. L. Giles and M. Gori, Eds. New York, NY: Springer-Verlag, 1997.

[45] P. R. Cohen, M. Darlymple, F. C. N. Pereira, J. W. Sullivan, R. A. G. Jr., J. L. Schlossberg, and S. W. Tyler, “Synergic use of direct manipulation and natural language,” in *Proc. CHI’89*, 1989, pp. 227-234.

[46] R. R. Murphy, “Biological and cognitive foundations of intelligent data fusion,” *Trans. SMC*, vol. 26, pp. 42–51, Jan. 1996.

[47] D. L. Hall and J. Llinas, “An introduction to multisensor data fusion,” *Proc. IEEE*, vol. 85, pp. 6–23, Jan. 1997.

[48] S. L. Lauritzen, *Graphical Models.* New York: Oxford University Press, 1996.

[49] G. F. Cooper, “The computational complexity of probabilistic inference using Bayesian belief networks,” *Artificial Intelligence*, vol. 42, pp. 393–405, 1990.

[50] J. Pearl, “Fusion, propagation, and structuring in belief networks,” *Artificial Intelligence*, vol. 29, pp. 241–288, 1986.

[51] B. Frey, *Graphical Models for Machine Learning and Digital Communication.* Cambridge, MA: MIT Press, 1998.

[52] J. M. Hammersley and D. C. Handscomb, *Monte Carlo Methods.* London: Chapman and Hall, 1964.

[53] R. M. Neal, *Bayesian Learning for Neural Networks.* New York: Springer-Verlag, 1996.

[54] G. E. Hinton, P. Dayan, B. J. Frey, and R. M. Neal, “The wake–sleep algorithm for unsupervised neural networks,” *Science*, vol. 268, pp. 1158–1161, 1995.

[55] Z. Ghahramani and M. I. Jordan, "Factorial hidden Markov models," *Machine Learning*, vol. 29, pp. 245–273, 1997.

[56] Z. Ghahramani, "On structured variational inference," Department of Computer Science, University of Toronto, Tech. Rep. CRG-TR-97-1, 1997.

[57] J. Binder, D. Koller, S. J. Russell, and K. Kanazawa, "Adaptive probabilistic networks with hidden variables," in *Machine Learning*, 1998. To appear. Available at http://robotics.stanford.edu/ koller.

[58] E. Bauer, D. Koller, and Y. Singer, "Update rules for parameter estimation in Bayesian networks," in *Proc. Uncertainty in Artificial Intelligence,* 1997, pp. 235-239.

[59] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of Royal Statistical Society, Series B*, vol. 39, pp. 1–38, 1977.

[60] R. J. Hathaway, "Another interpretation of the EM algorithm for mixture of distribution," *Statistics and Probability Letters*, vol. 4, pp. 53–56, 1986.

[61] R. M. Neal and G. E. Hinton, "A new view of the EM algorithm that justifies incremental and other variants," Department of Computer Science, University of Toronto, Tech. Rep., 1993.

[62] T. M. Cover and J. A. Thomas, *Elements of Information Theory.* New York: John Wiley & Sons, 1991.

[63] C. F. J. Wu, "On the convergence properties of the EM algorithm," *Annals of Statistics*, vol. 11, no. 1, pp. 95–103, 1983.

[64] N. Ueda and R. Nakano, "Deterministic annealing variant of the EM algorithm," in *Advances in Neural information processing systems 7* G. Tesauro and J. Alspector, Eds. Morgan Kaufmann, 1995.

[65] Z. Ghahramani and G. E. Hinton, "Switching state-space models," submitted for publication, 1998, http://www.cs.toronto.edu/ghahramani.

[66] E. Seneta, *Non-negative Matrices and Markov Chains.* New York: Springer-Verlag, 1981.

[67] A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimal decoding algorithm," *IEEE Trans. Information Theory*, vol. 13, pp. 260–269, 1967.

[68] R. J. Elliot, L. Aggoun, and J. B. Moore, *Hidden Markov Models: Estimation and Control.* New York: Springer-Verlag, 1995.

[69] B. D. O. Anderson and J. B. Moore, *Optimal Filtering.* Englewood Cliffs, NJ: Prentice-Hall, 1979.

[70] R. E. Kalman and R. S. Bucy, "New results in linear filtering and prediction," *Journal Basic Engineering (ASME)*, vol. D, no. 83, pp. 95–108, 1961.

[71] D. Q. Mayne, "A solution to the smoothing problem for linear dynamic systems," *Automatica*, vol. 4, pp. 73–92, 1966.

[72] D. C. Fraser and J. E. Potter, "The optimum linear smoother as a combination of two optimum linear filters," *IEEE Trans. Automatic Control*, vol. AC-14, pp. 387–390, Aug. 1969.

[73] H. E. Rauch, "Solutions to the linear smoothing problem," *IEEE Trans. Automatic Control*, vol. AC-8, pp. 371–372, Oct. 1963.

[74] R. H. Shumway, *Applied Statistical Time Series Analysis.* Englewood Cliffs, NJ: Prentice Hall, 1988.

[75] M. Brand and N. Oliver, "Coupled hidden Markov models for complex action recognition," in *Proc. Computer Vision and Pattern Recognition,* 1997, pp. 201–206.

[76] Y. Bar-Shalom and T. E. Fortmann, *Tracking and Data Association.* Orlando, FL: Academic Press, 1988.

[77] T. Kirubarajan and Y. Bar-Shalom, "Low observable target motion analysis using amplitude information," *IEEE Trans. Aerospace and Electronics Systems*, vol. 32, pp. 1367–1384, Oct. 1996.

[78] R. H. Shumway and D. S. Stoffer, "Dynamic linear models with switching," *Journal of the American Statistical Association*, vol. 86, pp. 763–769, Sept. 1991.

[79] D. McNeill and E. Levy, "Conceptual representations in language activity and gesture.," in *Speech, place and action: Studies in deixis and related topics,* J. Jarvella and W. Klein, Eds. New York: John Wiley & Sons, 1982.

[80] M. E. Hennecke, D. G. Stork, and K. V. Prasad, "Visionary speech: Looking ahead to practical speechreading systems," in *Proc. Speechreading by Man and Machine: Models, Systems and Applications Workshop*, 1995, pp. 331-352.

[81] M. Brand, "Source separation with coupled hidden Markov models," Vision and Modeling Group, MIT Media Lab, Tech. Rep. TR 427, 1997.

[82] A. Kendon, "Current issues in the study of gesture," in *The Biological Foundations of Gestures: Motor and Semiotic Aspects,* J.-L. Nespoulous, P. Peron, and A. R. Lecours, Eds. Lawrence Erlbaum Assoc., 1986.

[83] F. K. H. Quek, "Toward a vision-based hand gesture interface," in *Proc. Virtual Reality Software and Technology Conf.*, 1994, pp. 17–31.

[84] W. T. Freeman and C. D. Weissman, "Television control by hand gestures," in *Proc. Int'l Workshop Automatic Face and Gesture Recognition*, 1995, pp. 179–183.

[85] J. J. Kuch and T. S. Huang, "Vision based hand modeling and tracking," in *Proc. Fifth Int'l Conf. Computer Vision*, 1995, pp. 666-671.

[86] J. Lee and T. L. Kunii, "Constraint-based hand animation," in *Models and Techniques in Computer Animation*, Tokyo: Springer-Verlag, 1993.

[87] R. Cipolla and N. J. Hollinghurst, "Human-robot interface by pointing with uncalibrated stereo vision," *Image and Vision Computing*, vol. 14, pp. 171–178, March 1996.

[88] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham, "Active shape models - their training and application," *Computer Vision and Image Understanding*, vol. 61, pp. 38–59, Jan. 1995.

[89] S. X. Ju, M. J. Black, and Y. Y. Oob, "Cardboard people: A parameterized model of articulated image motion," in *Proc. Int'l Conf. Automatic Face and Gesture Recognition*, 1996, pp. 38–43.

[90] C. Kervrann and F. Heitz, "Learning structure and deformation modes of nonrigid objects in long image sequences," in *Proc. Int'l Workshop Automatic Face and Gesture Recognition*, 1995, pp. 11-15.

[91] A. Lanitis, C. J. Taylor, T. F. Cootes, and T. Ahmed, "Automatic interpretation of human faces and hand gestures using flexible models," in *Proc. Int'l Workshop Automatic Face and Gesture Recognition*, 1995, pp. 98–103.

[92] T. Heap and D. Hogg, "Towards 3D hand tracking using a deformable model," in *Proc. Int'l Conf. Automatic Face and Gesture Recognition*, 1996, pp. 140–145.

[93] A. F. Bobick and J. W. Davis, "Real-time recognition of activity using temporal templates," in *Proc. Int'l Conf. Automatic Face and Gesture Recognition*, 1996, pp. 149-154.

[94] J. L. Crowley, F. Berard, and J. Coutaz, "Finger tacking as an input device for augmented reality," in *Proc. Int'l Workshop Automatic Face and Gesture Recognition*, 1995, pp. 195–200.

[95] T. Darrell and A. P. Pentland, “Attention-driven expression and gesture analysis in an interactive environment,” in *Proc. Int’l Workshop Automatic Face and Gesture Recognition*, 1995, pp. 135–140.

[96] T. Darrell, I. Essa, and A. Pentland, “Task-specific gesture analysis in real-time using interpolated views,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 18, no. 12, pp. 1236–42, 1996.

[97] W. T. Freeman and M. Roth, “Orientation histograms for hand gesture recognition,” in *Proc. Int’l Workshop Automatic Face and Gesture Recognition*, 1995, pp. 210-214.

[98] J. Schlenzig, E. Hunter, and R. Jain, “Vision based hand gesture interpretation using recursive estimation,” *Proc. Twenty-Eight Asilomar Conf. Signals, Systems, and Computer*, 1994, pp. 1267-1272.

[99] J. Schlenzig, E. Hunter, and R. Jain, “Recursive identification of gesture inputs using hidden Markov models,” in *Proc. Second IEEE Workshop Applications of Computer Vision*, 1994, pp. 187–194.

[100] T. E. Starner and A. Pentland, “Visual recognition of American sign language using hidden Markov models,” in *Proc. Int’l Workshop Automatic Face and Gesture Recognition*, 1995, pp. 189–194.

[101] Y. Azoz, L. Devi, and R. Sharma, “Vision-based human arm tracking for gesture analysis using multimodal constraint fusion,” in *Proc. Advanced Display Federated Laboratory Symposium*, 1997, pp. 53–57.

[102] A. Azarbayejani, C. Wren, and A. Pentland, “Real-time 3-D tracking of the human body,” in *Proc. IMAGE’COM 96*, 1996, pp. 120–126.

[103] E. Clergue, M. Goldberg, N. Madrane, and B. Merialdo, “Automatic face and gestural recognition for video indexing,” in *Proc. Int’l Workshop Automatic Face and Gesture Recognition*, 1995, pp. 110–115.

[104] A. C. Downton and H. Drouet, “Image analysis for model-based sign language coding,” in *Progress In Image Analysis and Processing II: Proc. Sixth Int’l Conf. Image Analysis and Processing*, 1991, pp. 637–644.

[105] M. Etoh, A. Tomono, and F. Kishino, “Stereo-based description by generalized cylinder complexes from occluding contours,” *Systems and Computers in Japan*, vol. 22, no. 12, pp. 79–89, 1991.

[106] M. Fukumoto, Y. Suenaga, and K. Mase, “Finger-Pointer: Pointing interface by image processing,” *Computers and Graphics*, vol. 18, no. 5, pp. 633–642, 1994.

[107] D. M. Gavrila and L. S. Davis, “Towards 3-D model-based tracking and recognition of human movement: A multi-view approach,” in *Proc. Int’l Workshop Automatic Face and Gesture Recognition*, 1995, pp. 272–277.

[108] J. J. Kuch, “Vision-based hand modeling and gesture recognition for human computer interaction,” M.S. thesis, University of Illinois at Urbana-Champaign, 1994.

[109] J. Lee and T. L. Kunii, “Model-based analysis of hand posture,” *IEEE Computer Graphics and Applications*, pp. 77–86, Sept. 1995.

[110] S. Ahmad, “A usable real-time 3D hand tracker,” in *Proc. Twenty-Eight Asilomar Conf. Signals, Systems, and Computer*, 1994, pp. 1257-1261.

[111] R. Kjeldsen and J. Kender, “Finding skin in color images,” in *Proc. Int’l Conf. Automatic Face and Gesture Recognition*, 1996, pp. 312–317.

[112] F. K. H. Quek, T. Mysliwiec, and M. Zhao, “Finger Mouse: A freehand pointing interface,” in *Proc. Int’l Workshop Automatic Face and Gesture Recognition*, 1995, pp. 372–377.

[113] R. Cipolla, Y. Okamoto, and Y. Kuno, “Robust structure from motion using motion parallax,” in *Proc. Fourth Int’l Conf. Computer Vision*, 1993, pp. 374–382.

[114] J. Davis and M. Shah, "Recognizing hand gestures," in *Proc. Third European Conf. Computer Vision*, 1994, pp. 331–340.

[115] Y. Kuno, M. Sakamoto, K. Sakata, and Y. Shirai, "Vision-based human computer interface with user centered frame," in *Proc. IROS*, 1994, pp. 2923–2029.

[116] C. Maggioni, "A novel gestural input device for virtual reality," in *IEEE Annual Virtual Reality Int'l Symp.*, 1993, pp. 118–124.

[117] A. D. Wilson and A. F. Bobick, "Recovering the temporal structure of natural gestures," in *Proc. Int'l Conf. Automatic Face and Gesture Recognition*, October 1996, pp. 66–71.

[118] U. Bröckl-Fox, "Real-time 3-D interaction with up to 16 degrees of freedom from monocular image flows," in *Proc. Int'l Workshop Automatic Face and Gesture Recognition*, 1995, pp. 172–178.

[119] Y. Cui and J. Weng, "Learning-based hand sign recognition," in *Proc. Int'l Workshop Automatic Face and Gesture Recognition*, 1995, pp. 201–206.

[120] D. L. Swets and J. Weng, "Using discriminant eigenfeatures for image retrieval," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 18, no. 8, pp. 831–836, 1996.

[121] L. W. Campbell, D. A. Becker, A. Azarbayejani, A. F. Bobick, and A. Pentland, "Invariant features for 3-D gesture recognition," in *Proc. Int'l Conf. Automatic Face and Gesture Recognition*, 1996, pp. 157–162.

[122] M. T. Vo and C. Wood, "Building an application framework for speech and pen input integration in multimodal learning interfaces," in *Proc. IEEE Int'l Conf. Acoustics, Speech, and Signal Processing*, 1996, pp. 3545–3548.

[123] P. R. Cohen, M. Johnston, D. McGee, S. Oviatt, and J. Pittman, "QuickSet: Multimodal interaction for distributed applications," in *Proc. ACM Multimedia*, 1997, pp. 31–40.

[124] P. A. Devijver and J. Kittler, *Pattern Recognition: A Statistical Approach.* London: Prentice Hall, 1982.

[125] S. D. Silvey, *Statistical Inference.* London: Chapman and Hall, 1987.

[126] E. Charniak, *Statistical Language Learning.* Cambridge, MA: MIT Press, 1996.

[127] D. McNeill, *Hand and Mind: What Gestures Reveal About Thought.* Chicago: University of Chicago Press, 1992.

[128] A. Adjoudani and C. Benoit, "Audio-visual speech recognition compared across two architectures," 1995, http://ophale.icp.grenet.fr/ali.html.

[129] M. Minsky, "A framework for representing knowledge," in *The Psychology of Computer Vision*, P. H. Winston, Ed. New York: McGraw-Hill, 1975.

[130] F. Lehman, *Semantic Networks in Artificial Intelligence.* Oxford: Pergamon Press, 1992.

[131] J. Guan and D. A. Bell, *Evidence Theory and Its Applications*, vol. 1, Amsterdam: North-Holland, 1991.

[132] R. A. Bolt, "Put that there: Voice and gesture at the graphics interface," *ACM Computer Graphics*, vol. 14, no. 3, pp. 262–270, 1980.

[133] L. Nigay and J. Coutaz, "A design space for multimodal systems: concurrent processing and data fusion," in *Proc. InterCHI*, 1993, pp. 280-281.

[134] B. Suhm, P. Geunter, T. Kemp, A. Lavie, L. Mayfield, A. McNair, I. Rogina, T. Schultz, T. Sloboda, W. Ward, M. Woszczyna, and A. Waibel, "JANUS: Towards multilingual spoken language translation," 1995, http://www.is.cs.cmu.edu/ISL.speech.janus.html.

[135] J. A. Pittman, I. Smith, P. Cohen, S. Oviatt, and T.-C. Yang, “QuickSet: A multimodal interface for military simulation,” in *Proc. Sixth Conf. Computer-Generated Forces and Behavioral Representation*, 1996, pp. 217–224.

[136] C. Codella, R. Jalili, L. Koved, et al., “Interactive simulation in a multi-person virtual world,” in *ACM Conf. Human Factors in Computing Systems - CHI’92*, 1992, pp. 329–334.

[137] P. Maes, T. Darrell, B. Blumberg, and A. Pentland, “The ALIVE system: Wireless, full-body interaction with autonomous agents,” *Proc. ACM Multimedia Systems*, 1996, pp. 570–576

[138] A. Pentland, “Smart rooms,” *Scientific American*, pp. 54–62, April 1996.

[139] M. Kakimoto, N. Tosa, J. Mori, and A. Sanada, “Tool of Neuro-Baby,” *Institute of Television Engineers of Japan Tech. Rep.*, vol. 16, pp. 7–12, June 1992.

[140] N. Tosa, “Neuro-Baby,” ATR, Japan, 1992, http://www.mic.atr.co.jp/tosa.

[141] R. Kober, U. Harz, and J. Schiffers, “Fusion of visual and acoustic signals for command-word recognition,” in *Proc. IEEE Int’l Conf. Acoustics, Speech, and Signal Processing*, 1997, pp. 1495–1497.

[142] U. Meier, W. Hürst, and P. Duchnowski, “Adaptive bimodal sensor fusion for automatic speechreading,” in *Proc. IEEE Int’l Conf. Acoustics, Speech, and Signal Processing*, 1996, pp. 833-836.

[143] E. T. Levy and D. McNeill, “Speech, gesture, and discourse,” *Discourse Processes*, no. 15, pp. 277–301, 1992.

VITA

Vladimir Ivan Pavlovic earned a Diploma in electrical engineering from the University of Novi Sad, Yugoslavia, in 1991 and a Master of Science degree in electrical engineering from the University of Illinois at Chicago in 1993. From 1991 to 1993 he was a teaching assistant at the Department of Electrical Engineering at the University of Illinois at Chicago. In 1994 he joined the Image Formation and Processing group at the Beckman Institute for Advanced Science and Technology, University of Illinois at Urbana-Champaign, as a research assistant. Vladimir Pavlovic is a recipient of the National Science Foundation of Serbia Honorable Scholarship from 1989 to 1991.