Division of Engineering, Science and Computing
Department of Computing

Techniques for the Discovery of Anomalous Human Behaviour in Intelligent Environments

Sebastian Lühr

This thesis is presented for the Degree of
Doctor of Philosophy
of
Curtin University of Technology

March 2006

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

Sebastian Lühr

Date

Abstract

Motivated by a desire to create smart homes that will enable the elderly to maintain their independence for as long as possible, this thesis presents techniques for detecting abnormality in human activity observed in both laboratory and real world smart environments.

The use of stochastic models as tools for learning models of normality, with which incoming observational data from a visual tracking system can be examined, is investigated. In particular, the Hierarchical Hidden Markov Model (HHMM) is applied to the training of multi-level models of behaviour to show that the hierarchical structure of the model allows for a more expressive representation of human behaviour than is possible using flat models. The usefulness of modelling duration in models of human activity is then investigated by comparing the classification and abnormality detection performance of the Hidden Markov Model (HMM) against that of the Explicit State Duration HMM (ESD-HMM). The data sets used differ primarily in the duration of activities rather than in the ordering of the events. An extension of the ESD-HMM where the state transition times are inferred from an observation signal that has been augmented with pressure mat sensor data is then introduced. Work into this area is then concluded with results from experimentation on real world data.

A data mining technique that employs Intertransaction Association Rule (IAR) mining to discover new and changing human behaviours is then presented. The Frequent Pattern Tree (FP-Tree) and the Frequent Pattern Growth (FP-Growth) algorithm are extended for IAR mining. The resulting data structure and mining algorithm, dubbed the Extended FP-Tree (EFP-Tree) and Extended FP-Growth (EFP-Growth) respectively, are benchmarked against the First Intra Then Inter (FITI) algorithm, the existing state of the art algorithm for IAR mining. Results demonstrating that the EFP-Growth algorithm is an order of magnitude computationally more efficient than FITI are presented and discussed. The viability of emergent IAR mining as a technique for identifying unexpected behaviours in a smart home environment is affirmed with a discussion of

observations made mining emergent behaviours from sensor event data recorded in the homes of two real world subjects.

Finally, a novel visual interface that enables emergent behaviours to be examined in the context of the original data is introduced. Mapping emergent IARs back into the original data space, the interface is demonstrated to allow greater insight to be gained in significantly less time than is possible by manual inspection of the sensor event log data.

Acknowledgments

This thesis is the result of the work and combined support of many individuals.

Foremost acknowledgment must go to Prof. Geoff West and Prof. Svetha Venkatesh for their supervision and guidance these past few years. Without their combined direction and their ceaseless patience I would still be freewheeling today.

To Michelle and to my parents; thank you for your love and for your support. Having a strong base to fall back on to is very reassuring when undertaking a project such as this. Thanks, Shell, for keeping me sane and for all your smiles.

Thanks also must go to Leon, Ezra and Patrick for their patience and help over the years. I wish you the best in your chosen career paths. Extra thanks to you, Leon, for not doing anything to my desk. I really appreciate it even though you were probably just too busy to get around to it.

Finally, respect to the thousands of independent musicians and groups whose brilliant work is generally ignored by the music cartels and the radio stations they control. Thank you for not conforming to the cookie cutter mold that the music industry has stagnant into. Your skill and spirit is an inspiration and has made the last few years more enjoyable. I may not have a musical bone in my body but I know what I like.

Oh, and thank *you* for reading this far. You can stop, now; there is no money hidden amidst these pages.

Published Work

Elements of this thesis have previously been published elsewhere. Ownership of these elements remains with the copyright holders of the relevant publications:

Section 3.1: Sebastian Lühr, Hung H. Bui, Svetha Venkatesh and Geoff West (2003). Recognition of Human Activity Through Hierarchical Stochastic Learning. In *IEEE International Conference on Pervasive Computing and Communications*, pages 416–423. © 2003 IEEE. Reprinted, with permission, from Proceedings of the First IEEE International Conference on Pervasive Computing and Communications.

Section 3.2: Sebastian Lühr, Svetha Venkatesh, Geoff West and Hung H. Bui (2004). Explicit State Duration HMM for Abnormality Detection in Sequences of Human Activity. In *Proceedings of the* 8^{th} *Pacific Rim International Conference on Artificial Intelligence*, volume 3157 of *Lecture Notes in Artificial Intelligence*, pages 983–984. © 2004 Springer. With *kind* permission of Springer Science and Business Media.

Section 4.3: Sebastian Lühr, Svetha Venkatesh and Geoff West (2005). Emergent Intertransaction Association Rules for Abnormality Detection. In *Proceedings of the International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, pages 343–347. © 2005 IEEE. Reprinted, with permission, from Proceedings of the International Conference on Intelligent Sensors, Sensor Networks and Information Processing.

Chapter 1 (introduction), Chapter 4 (introduction), Section 4.1, Section 4.2 and Chapter 5: Sebastan Lühr, Geoff West and Svetha Venkatesh (2007). Recognition of Emergent Human Behaviour in a Smart Home: A Data Mining Approach. In *Pervasive and Mobile Computing*, to appear. © 2007 Elsevier B. V. Reprinted with permission.

Table of Contents

Abstract **i**

Acknowledgments **iii**

Published Work **iv**

Table of Contents **vi**

List of Figures **xvii**

List of Tables **xxv**

List of Algorithms **xxvi**

1 Introduction **1**

1.1 Aims and Approach . . . 2

1.2 Significance and Contributions . . . 5

1.3 Structure of the Thesis . . . 6

2 Related Work **7**

2.1 Hidden Markov Model . . . 8

2.1.1 Inferencing . . . 9

2.1.2 Decoding . . . 10

2.1.3 Training . . . 11

2.1.4 Application to Activity Recognition . . . 12

2.2 Explicit State Duration HMM . . . 13

2.3 Hierarchical HMM . . . 15

2.4 Association Rule Mining . . . 19

2.4.1 Apriori . . . 20
2.4.2 Frequent Pattern Growth . . . 22
2.5 Intertransaction Association Rules . . . 26
2.5.1 E-Apriori . . . 28
2.5.2 First Intra Then Inter . . . 28
2.5.3 Frequent Pattern Growth . . . 35
2.6 Visual Association Rule Exploration . . . 35
2.7 Review Summary . . . 37

3 Detecting Human Behaviour Abnormality via Stochastic Models 38
3.1 Hierarchical HMM . . . 39
3.1.1 Experimentation Methodology . . . 41
3.1.2 Modelling Human Behaviour . . . 44
3.1.3 Classification Results . . . 52
3.1.4 Conclusion . . . 53
3.2 Explicit State Duration HMM . . . 54
3.2.1 Experimentation Methodology . . . 54
3.2.2 Sequence Classification . . . 56
3.2.3 Duration Abnormality . . . 56
3.2.4 Longer Term Duration Abnormality . . . 59
3.2.5 Conclusion . . . 60
3.3 Explicit State Duration HMM With Observed Time Indices . . . 61
3.3.1 Model Derivation . . . 61
3.3.2 Sequence Classification . . . 68
3.3.3 Duration Abnormality . . . 68
3.3.4 Conclusion . . . 69
3.4 Real World Application . . . 70
3.4.1 Experimentation Methodology . . . 70
3.4.2 Sequence Classification . . . 72
3.4.3 Abnormality Detection . . . 74
3.4.4 Longer Term Duration Abnormality . . . 79
3.4.5 Conclusion . . . 79
3.5 Conclusion . . . 81

4 Emergent Intertransaction Association Rule Mining: Discovering New and Changing Human Behaviours **83**

4.1 Intertransaction Association Rules 85

4.2 Extended Frequent Pattern Tree 86

4.2.1 Tree Construction 86

4.2.2 Data Mining 91

4.2.3 Benchmark Comparisons 102

4.3 Identifying Emergent Behaviours 111

4.3.1 Emergent Intertransaction Association Rules 111

4.3.2 Minimal Emergent Intertransaction Association Rules 112

4.3.3 Observations on the Working Professional Data Set 113

4.3.4 Observations on the Retiree Data Set 115

4.3.5 Automatic Selection of the Intertransaction Sliding Window Size 117

4.4 Conclusion 120

5 Visual Exploration and Analysis of Emergent Behaviours **122**

5.1 An Interface for the Visual Exploration of Emergent IARs 123

5.2 Design Rationale 126

5.3 Repeat Experimentation on Real World Data Sets 134

5.3.1 Observations on the Working Professional Data Set 135

5.3.2 Observations on the Retiree Data Set 138

5.3.3 Automatic Selection of the Intertransaction Sliding Window Size Revisited 146

5.4 Conclusion 156

6 Conclusion **162**

6.1 Future Directions 164

A Hierarchical HMM State Transition and Emission Likelihoods **166**

A.1 Kitchen Models 166

A.1.1 "Food Preparation First" 166

A.1.2 "Washing Dishes First" 170

A.1.3 "Washing Dishes and Preparing Food" 174

A.1.4 "Round Robin" 178

A.2 Lounge Room Models 182
A.2.1 "Watch Television" 182
A.2.2 "Read Book on Couch" 186
A.2.3 "Eat Dinner" 190
A.2.4 "Eat Dinner While Watching TV" 194
A.2.5 "There is Nothing Good on TV – Read a Book Instead" 198

Bibliography **202**

List of Figures

2.1 Finite state machine representation of a three state HMM showing the possible state transitions. 8

2.2 A HMM rolled out over time. At each discrete time interval t an observation o_t is emitted by state q_t to produce the observation sequence $O = \{o_1, o_2, \ldots o_t, \ldots o_T\}$. 9

2.3 A ESD-HMM rolled out over time. At each discrete time interval the t^{th} observation o_t is emitted by a state q_k to produce the observation sequence $O = [o_1, o_2, \ldots o_t, \ldots o_T]$. A state q_k is active for duration d_k prior to making a transition to another state. 13

2.4 Finite state machine representation of a HHMM showing the possible state transitions between the root (R) node, the internal (I) states and the production (P) states. 17

2.5 A HHMM rolled out over time. At each discrete time interval t the observation o_t is emitted by a production state q_i^D to produce the observation sequence $O = [o_1, o_2, \ldots o_t, \ldots o_T]$. Control returns to a state q_i^d when one of its substates q_j^{d+1} makes a transition to an end state. End states are marked with an E. 17

2.6 The FP-Tree for the example database in Table 2.1 with minimum support $\alpha = 2$. Colon delineated numbers depict the node frequency count. 24

2.7 The conditional FP-Tree $T_c|$B. Colon delineated numbers depict the node frequency count. 26

2.8 The FILT data structure of the (a) item links, (b) generator and extension links (c) subset links and (d) descendant links used in FITI. 33

3.1 Layout of the kitchen showing the approximate track and elapsed time t in seconds for the (a) "food preparation first," (b) "washing dishes first," (c) "washing dishes and preparing the food" and the (d) "round robin" meal preparation sequences. Figure (a) and Figure (b) © 2003 IEEE. Reprinted, with permission, from Proceedings of the First IEEE International Conference on Pervasive Computing and Communications. 42

3.2 Layout of the lounge room showing the approximate track and elapsed time t in seconds for the (a) "watch television," (b) "read a book on the couch," (c) "eat dinner," (d) "eat dinner while watching television" and the (e) "there is nothing good on TV, read a book instead" lounge room activity sequences. 43

3.3 The trained model for the "food preparation first" kitchen style. Darker arrows indicate stronger state dependence and a higher transitional likelihood among the production (P), internal (I) and end (E) states. Insignificant transitions have been omitted for clarity. 45

3.4 The trained model for the "washing dishes first" style. Darker arrows indicate stronger state dependence and a higher transitional likelihood among the production (P), internal (I) and end (E) states. Insignificant transitions have been omitted for clarity. 46

3.5 The trained model for the "washing dishes and preparing the food" kitchen style. Darker arrows indicate stronger state dependence and a higher transitional likelihood among the production (P), internal (I) and end (E) states. Insignificant transitions have been omitted for clarity. 47

3.6 The trained model for the "round robin" kitchen style. Darker arrows indicate stronger state dependence and a higher transitional likelihood among the production (P), internal (I) and end (E) states. Insignificant transitions have been omitted for clarity. 48

3.7 The trained model for the "watch television" lounge style. Darker arrows indicate stronger state dependence and a higher transitional likelihood among the production (P), internal (I) and end (E) states. Insignificant transitions have been omitted for clarity. 49

3.8 The trained model for the "read book on couch" lounge style. Darker arrows indicate stronger state dependence and a higher transitional likelihood among the production (P), internal (I) and end (E) states. Insignificant transitions have been omitted for clarity. 50

3.9 The trained model for the "eat dinner" lounge style. Darker arrows indicate stronger state dependence and a higher transitional likelihood among the production (P), internal (I) and end (E) states. Insignificant transitions have been omitted for clarity. 51

3.10 The trained model for the "eat dinner while watching TV" lounge style. Darker arrows indicate stronger state dependence and a higher transitional likelihood among the production (P), internal (I) and end (E) states. Insignificant transitions have been omitted for clarity. 51

3.11 The trained model for the "there is nothing good on TV, read a book instead" lounge style. Darker arrows indicate stronger state dependence and a higher transitional likelihood among the production (P), internal (I) and end (E) states. Insignificant transitions have been omitted for clarity. 52

3.12 ROC curves for (a) the twelve state HMM, (b) the two state left-right HMM, (c) the three state ESD-HMM and (d) the two state left-right ESD-HMM. 58

3.13 The normalised log-likelihood for each of the models as the primary activity in an observation sequence is varied from one second to five minutes. The normal duration for the primary activity is circa 40 seconds. Only the left-right ESD-HMM is able to detect abnormality in a timely manner. 60

3.14 The OTI ESD-HMM rolled out over time. The states $q_1 \dots q_K$ are seen to produce the observations $o_{\tau_1} \dots o_{\tau'_K}$ over k sequence segments. The starting and ending time of each state k is given as τ_k and τ'_k respectively. The duration of each state is ϕ_k such that $\tau'_k - \tau_k = \phi_k + 1$. 62

3.15 ROC curve for the five state ESD-HMM with observed time indices. . . 69

3.16 Approximate layout of the real world environment showing the location of the areas of interest, cameras and the virtual pressure mats. 71

3.17 The real world scene as viewed by cameras deployed in (a) the kitchen and (b) the lounge room. 71

3.18 ROC curves showing the performance of (a) the HMM, (b) the left-right HMM, (c) the ESD-HMM and (d) the left-right ESD-HMM for the detection of abnormality among the real world sequences. 76

3.19 The normalised log-likelihoods of the normal and abnormal sequences for (a) the HMM, (b) the left-right HMM, (c) the ESD-HMM and (d) the left-right ESD-HMM for the detection of abnormality among the 116 real world sequences. 78

3.20 The normalised log-likelihood of (a) the HMM, (b) the left-right HMM, (c) the OTI ESD-HMM and (d) the left-right OTI ESD-HMM as the time spent standing in the lounge hallway in a sequence from the `Walk from rear room to kitchen` class is artificially adjusted. 80

4.1 The Extended FP-Tree for the example database with $w = 5$ and $\alpha = 3$. Subscript numbers represent the dimensional offset, in this case the item time, relative to the intratransaction items while colon delineated numbers depict the node frequency. The header tables of the intertransaction item subtrees have been omitted for clarity. 89

4.2 The intraitem FP-Tree at various stages of construction after the intraitems from the extended transactions in Table 4.2 at (a) time 1 are added, (b) the tree after time 6, (c) after time 7 and, finally, (d) after time 11. 91

4.3 EFP-Tree construction as the frequent interitems from Table 4.2 at time 1 are inserted. The header tables of the intertransaction item subtrees have been omitted for clarity. 92

4.4 EFP-Tree construction as the frequent interitems from Table 4.2 at time 2 are inserted into the EFP-Tree in Figure 4.3. The header tables of the intertransaction item subtrees have been omitted for clarity. 93

4.5 EFP-Tree construction as the frequent interitems from Table 4.2 at time 3 are inserted into the EFP-Tree in Figure 4.4. The header tables of the intertransaction item subtrees have been omitted for clarity. 94

4.6 EFP-Tree construction as the frequent interitems from Table 4.2 at time 5 are inserted into the EFP-Tree in Figure 4.5. The header tables of the intertransaction item subtrees have been omitted for clarity. 95

4.7 EFP-Tree construction as the frequent interitems from Table 4.2 at time 6 are inserted into the EFP-Tree in Figure 4.6. The header tables of the intertransaction item subtrees have been omitted for clarity. 96

4.8 EFP-Tree construction as the frequent interitems from Table 4.2 at time 7 are inserted into the EFP-Tree in Figure 4.7. The header tables of the intertransaction item subtrees have been omitted for clarity. 97

4.9 EFP-Tree construction as the frequent interitems from Table 4.2 at time 9 are inserted into the EFP-Tree in Figure 4.8. The header tables of the intertransaction item subtrees have been omitted for clarity. 98

4.10 The conditional EFP-Tree tree T_c and conditional interitem FP-Tree T_e for (a) $T_c|\mathrm{A}_0$, (b) $T_e|\mathrm{A}_0$ and $T_e|\mathrm{B}_0\mathrm{A}_0$, (c) $T_c|\mathrm{C}_0\mathrm{A}_0$, (d) $T_c|\mathrm{C}_0$ and (e) $T_e|\mathrm{C}_0$ when the EFP-Tree in Figure 4.1 is mined with a minimum support threshold of $\alpha = 2$. 101

4.11 The execution time (a), memory use (c) and the number of rules found (e) for the synthetic sparse data set with the intertransaction window size fixed at $w = 4$ and the execution time (b), memory use (d) and the number of rules found (f) for the synthetic dense data set with $w = 6$ as the minimum support threshold is adjusted. 107

4.12 The execution time (a), memory use (c) and the number of rules found (e) for the working professional subject and the execution time (b), memory use (d) and the number of rules found (f) for the retiree subject as the minimum support threshold is adjusted. The intertransaction window size is fixed at $w = 6$ with an interval size of 300 seconds for both data sets. 108

4.13 The execution time (a), memory use (c) and the number of rules found (e) for the synthetic sparse data set and the execution time (b), memory use (d) and the number of rules found (f) for the synthetic dense data set as the intertransaction window size is increased. The minimum support level was fixed at 1% and 10% for the sparse and dense data sets respectively. 109

4.14 The execution time (a), memory use (c) and the number of rules found (e) for the working professional subject with a minimum support threshold $\alpha = 1\%$ and the execution time (b), memory use (d) and the number of rules found (f) for the retiree subject with $\alpha = 0.4\%$ as the intertransaction window size is increased. 110

4.15 The number of rules discovered in the (a) first and (b) second week of the working professional data set and the (a) first and (b) second week of the retiree data set as the sliding intertransaction window size is increased. Five minute transaction intervals and a raw minimum support threshold of $\alpha = 6$ was used. 118

4.16 The number of rules discovered in the (a) first and (b) second week of the working professional data set and the (a) first and (b) second week of the retiree data set as the sliding intertransaction window size is increased. Five minute transaction intervals and a raw minimum support threshold of $\alpha = 8$ was used. 119

5.1 The visual interface highlighting the presence of selected rules (top) on the event time line. The horizontal axis represents the date and time of the intervals while the vertical axis represents the sensor events. The date and time and the name of the sensor event that the mouse cursor is pointing at is being displayed. 124

5.2 Magnified view of (a) the compact time line showing the sensor events within a single room over a period of one day and (b) the expanded view showing approximately three hours from the same room and period. . . 125

5.3 The number of sensor events triggered per five minute interval over the first six days of data from the working professional subject data set. . . 127

5.4 The number of sensor events triggered per five minute interval over the days 7 to 8 of data from the working professional subject data set. . . . 128

5.5 The number of sensor events triggered per five minute interval over the last four days of data from the working professional subject data set. . . 129

5.6 The number of sensor events triggered per five minute interval over the first six days of data from the retiree subject data set. 130

5.7 The number of sensor events triggered per five minute interval over days 7 to 8 of data from the retiree subject data set. 131

5.8 The number of sensor events triggered per five minute interval over the last four days of data from the retiree subject data set. 132

5.9 Visualisation of the flurry of kitchen activity on the third day of the first week of the working professional data using a three minute transaction interval. The discovered rules describe temporal relationships between the kitchen drawer, cabinet and refrigerator not seen in the training data. Triggered events are coloured in blue while events matching the emergent IARs are coloured in red. 136

5.10 Visualisation of the abnormal kitchen door activity using a two minute transaction interval on the third day of the first week of the working professional data. The regularity, frequency and length of time over which the kitchen door is repeatedly opened and closed is unusual. Triggered events are coloured in blue while events matching the emergent IARs are coloured in red. 137

5.11 Visualisation of the opening and closing of the bathroom sink faucets one to two minutes after their previous use on (a) days three to five and on (b) days six to eight of the second week of the working professional. The rules are shown using a three minute transaction interval. Triggered events are coloured in blue while events matching the emergent IARs are coloured in red. 139

5.12 Visualisation of the abnormal bathroom activity using a two minute transaction interval on the first day of the first week of the retiree data. The subject is seen to repeatedly open and close the bathroom door over a forty-five minute interval. Activity in the lounge room and the kitchen suggests that frequent short visits to the bathroom are being made. Triggered events are coloured in blue while events matching the emergent IARs are coloured in red. 141

5.13 Visualisation of the innocuous emergent kitchen door behaviour on the second day of the first week of the retiree data set shown on a two minute transaction interval. The historical data sets suggests that the kitchen door would normally remain in its open or closed state for longer periods of time. Triggered events are coloured in blue while events matching the emergent IARs are coloured in red. 142

5.14 Visualisation of the kitchen related activity on (a) days one to three, (b) days four to six and (c) days seven and eight of the first week of the retiree data set. The presence of this emergent behaviour appears to be caused by a lack of historical support. The behaviour is shown using a two minute transaction interval. Triggered events are coloured in blue while events matching the emergent IARs are coloured in red. 143

5.15 Visualisation of the repetitive toaster usage on the first day of the first week of the retiree data set shown using a one minute transaction interval. Triggered events are coloured in blue while events matching the emergent IARs are coloured in red. 144

5.16 Visualisation of the malfunctioning shower faucet sensors on the first day of the second week of the retiree data set using a two minute transaction interval. Continuous toggling of the state of these sensors and their notable absence in the remainder of the data set reinforces the suspicion that a significant fault has been detected here. Triggered events are coloured in blue while events matching the emergent IARs are coloured in red. 145

5.17 Visualisation of the unusual access of the butler's pantry on the seventh day of the second week of the retiree data set using a two minute transaction interval. Triggered events are coloured in blue while events matching the emergent IARs are coloured in red. 147

5.18 The number of rules discovered in the first week of the working professional data set using (a) one, (b) two, (c) three, (d) four and (e) five minute transaction intervals as the intertransaction sliding window size is increased. A raw minimum support threshold of $\alpha = 6$ was used. . . 149

5.19 The number of rules discovered in the first week of the working professional data set using (a) one, (b) two, (c) three, (d) four and (e) five minute transaction intervals as the intertransaction sliding window size is increased. A raw minimum support threshold of $\alpha = 8$ was used. . . 150

5.20 The number of rules discovered in the second week of the working professional data set using (a) one, (b) two, (c) three, (d) four and (e) five minute transaction intervals as the intertransaction sliding window size is increased. A raw minimum support threshold of $\alpha = 6$ was used. . . 151

5.21 The number of rules discovered in the second week of the working professional data set using (a) one, (b) two, (c) three, (d) four and (e) five minute transaction intervals as the intertransaction sliding window size is increased. A raw minimum support threshold of $\alpha = 8$ was used. . . 153

5.22 The number of rules discovered in the first week of the retiree data set using (a) one, (b) two, (c) three, (d) four and (e) five minute transaction intervals as the intertransaction sliding window size is increased. A raw minimum support threshold of $\alpha = 6$ was used. 154

5.23 The number of rules discovered in the first week of the retiree data set using (a) one, (b) two, (c) three, (d) four and (e) five minute transaction intervals as the intertransaction sliding window size is increased. A raw minimum support threshold of $\alpha = 8$ was used. 155

5.24 The number of rules discovered in the second week of the retiree data set using (a) one, (b) two, (c) three, (d) four and (e) five minute transaction intervals as the intertransaction sliding window size is increased. A raw minimum support threshold of $\alpha = 6$ was used. 157

5.25 The number of rules discovered in the second week of the retiree data set with the malfunctioning shower faucet sensor events removed using (a) one, (b) two, (c) three, (d) four and (e) five minute transaction intervals as the intertransaction sliding window size is increased. A raw minimum support threshold of $\alpha = 6$ was used. 158

5.26 The number of rules discovered in the second week of the retiree data set using (a) one, (b) two, (c) three, (d) four and (e) five minute transaction intervals as the intertransaction sliding window size is increased. A raw minimum support threshold of $\alpha = 8$ was used. 159

5.27 The number of rules discovered in the second week of the retiree data set with the malfunctioning shower sensor removed using (a) one, (b) two, (c) three, (d) four and (e) five minute transaction intervals as the intertransaction sliding window size is increased. A raw minimum support threshold of $\alpha = 8$ was used. 160

List of Tables

2.1 An example database showing the raw and frequency ordered items in each transaction. 24

2.2 The example database from Table 2.1 with introduced transaction times. The extended transaction items shown are gathered when the transaction time is used as an intertransaction dimensional offset and a sliding intertransaction window of size $w = 4$ is passed over the database. . . . 27

2.3 FIT representation of the example database in Table 2.2 created with a minimum support threshold of $\alpha = 3$ and encoded using the lookup link structure in Figure 2.8(a). 32

3.1 Confusion matrix for the cooking test sequences. 52

3.2 Confusion matrix for the lounge room test sequences. 53

3.3 The optimal number of states and classification accuracy for both the training and the test sequences for each of the four model types. 57

3.4 Test sequence confusion matrix for the twelve state standard HMM. . . 57

3.5 Test sequence confusion matrix for both the three state ESD-HMM and the two state left-right ESD-HMM. 57

3.6 Test sequence confusion matrix for the five state ESD-HMM with observed time indices. 68

3.7 The number of states used to model the four real world sequence classes for the four models. 72

3.8 Confusion matrix for the fully connected HMM. 73

3.9 Confusion matrix for the left-right constrained HMM. 73

3.10 Confusion matrix for the observed time indices ESD-HMM. 73

3.11 Confusion matrix for the observed time indices ESD-HMM with left-right constraints. 73

3.12 Summary of the real world abnormality sequences. 75

4.1 Example database with the unsorted and descending frequency ordered items. The time at which each transaction occurs is shown. 88

4.2 Extended transactions retrieved from the example database in Table 4.1 using a sliding intertransaction window of size $w = 5$. 88

4.3 Parameters used in the generation of the synthetic sparse and synthetic dense data sets. 104

A.1 Initial state likelihoods stored at the root node in the "food preparation first" model. 166

A.2 State transition likelihoods governing the second layer of the "food preparation first" model stored at the root node. 167

A.3 Initial state likelihoods for the "cooking" submodel in the "food preparation first" model stored at state q_1^2. 167

A.4 State transition likelihoods governing the production state children of the "cooking" submodel in the "food preparation first" model stored at state q_1^2. 167

A.5 Emission likelihoods of the production state children of the "cooking" submodel in the "food preparation first" model stored at the children states of q_1^2. 167

A.6 Initial state likelihoods for the "fridge & food prep" submodel in the "food preparation first" model stored at state q_2^2. 168

A.7 State transition likelihoods governing the production state children of the "fridge & food prep" submodel in the "food preparation first" model stored at state q_2^2. 168

A.8 Emission likelihoods of the production state children of the "fridge & food prep" submodel in the "food preparation first" model stored at the children states of q_2^2. 168

A.9 Initial state likelihoods for the "enter/exit" submodel in the "food preparation first" model stored at state q_3^2. 168

A.10 State transition likelihoods governing the production state children of the "enter/exit" submodel in the "food preparation first" model stored at state q_3^2. 169

A.11 Emission likelihoods of the production state children of the "enter/exit" submodel in the "food preparation first" model stored at the children states of q_3^2. 169

A.12 Initial state likelihoods stored at the root node in the "washing dishes first" model. 170

A.13 State transition likelihoods governing the second layer of the "washing dishes first" model stored at the root node. 170

A.14 Initial state likelihoods for the "cooking" submodel in the "washing dishes first" model stored at the children states of q_1^2. 170

A.15 State transition likelihoods governing the production state children of the "cooking" submodel in the "washing dishes first" model stored at state q_1^2. 171

A.16 Emission likelihoods of the production state children of the "cooking" submodel in the "washing dishes first" model stored at the children states of q_1^2. 171

A.17 Initial state likelihoods for the "wash dishes" submodel in the "washing dishes first" model stored at state q_2^2. 171

A.18 State transition likelihoods governing the production state children of the "wash dishes" submodel in the "washing dishes first" model stored at state q_2^2. 172

A.19 Emission likelihoods of the production state children of the "wash dishes" submodel in the "washing dishes first" model stored at the children states of q_2^2. 172

A.20 Initial state likelihoods for the "enter/exit" submodel in the "washing dishes first" model stored at state q_3^2. 172

A.21 State transition likelihoods governing the production state children of the "enter/exit" submodel in the "washing dishes first" model stored at state q_3^2. 173

A.22 Emission likelihoods of the production state children of the "enter/exit" submodel in the "washing dishes first" model stored at the children states of q_3^2. 173

A.23 Initial state likelihoods stored at the root node in the "washing dishes and preparing food" model. 174

A.24 State transition likelihoods governing the second layer of the "washing dishes and preparing food" model stored at the root node. 174

A.25 Initial state likelihoods for the "enter/exit and wash dishes" submodel in the "washing dishes and preparing food" model stored at state q_1^2. 174

A.26 State transition likelihoods governing the production state children of the "enter/exit and wash dishes" submodel in the "washing dishes and preparing food" model stored at state q_1^2. 175

A.27 Emission likelihoods of the production state children of the "enter/exit and wash dishes" submodel in the "washing dishes and preparing food" model stored at the children states of q_1^2. 175

A.28 Initial state likelihoods for the "food prep" submodel in the "washing dishes and preparing food" model stored at state q_2^2. 175

A.29 State transition likelihoods governing the production state children of the "food prep" submodel in the "washing dishes and preparing food" model stored at state q_2^2. 176

A.30 Emission likelihoods of the production state children of the "food prep" submodel in the "washing dishes and preparing food" model stored at the children states of q_2^2. 176

A.31 Initial state likelihoods for the "cooking" submodel in the "washing dishes and preparing food" model stored at state q_3^2. 176

A.32 State transition likelihoods governing the production state children of the "cooking" submodel in the "washing dishes and preparing food" model stored at state q_3^2. 177

A.33 Emission likelihoods of the production state children of the "cooking" submodel in the "washing dishes and preparing food" model stored at the children states of q_3^2. 177

A.34 Initial state likelihoods stored at the root node in the "round robin" model. 178

A.35 State transition likelihoods governing the second layer of the "round robin" model stored at the root node. 178

A.36 Initial state likelihoods for the "cooking" submodel in the "round robin" model stored at state q_1^2. 178

A.37 State transition likelihoods governing the production state children of the "cooking" submodel in the "round robin" model stored at state q_1^2. 179

A.38 Emission likelihoods of the production state children of the "cooking" submodel in the "round robin" model stored at the children states of q_1^2. 179

A.39 Initial state likelihoods for the "food prep" submodel in the "round robin" model stored at state q_2^2. 179

A.40 State transition likelihoods governing the production state children of the "food prep" submodel in the "round robin" model stored at state q_2^2. 180

A.41 Emission likelihoods of the production state children of the "food prep" submodel in the "round robin" model stored at the children states of q_2^2. 180

A.42 Initial state likelihoods for the "enter/exit via stove" submodel in the "round robin" model stored at state q_3^2. 180

A.43 State transition likelihoods governing the production state children of the "enter/exit via stove" submodel in the "round robin" model stored at state q_3^2. 181

A.44 Emission likelihoods of the production state children of the "enter/exit via stove" submodel in the "round robin" model stored at the children states of q_3^2. 181

A.45 Initial state likelihoods stored at the root node in the "watch television" model. 182

A.46 State transition likelihoods governing the second layer of the "watch television" model stored at the root node. 182

A.47 Initial state likelihoods for the "walk across room / exit" submodel in the "watch television" model stored at state q_1^2. 182

A.48 State transition likelihoods governing the production state children of the "walk across room / exit" submodel in the "watch television" model stored at state q_1^2. 183

A.49 Emission likelihoods of the production state children of the "walk across room / exit" submodel in the "watch television" model stored at the children states of q_1^2. 183

A.50 Initial state likelihoods for the "enter" submodel in the "watch television" model stored at state q_2^2. 183

A.51 State transition likelihoods governing the production state children of the "enter" submodel in the "watch television" model stored at state q_2^2. 184

A.52 Emission likelihoods of the production state children of the "enter" submodel in the "watch television" model stored at the children states of q_2^2. 184

A.53 Initial state likelihoods for the "watch TV" submodel in the "watch television" model stored at state q_3^2. 184

A.54 State transition likelihoods governing the production state children of the "watch TV" submodel in the "watch television" model stored at state q_3^2. 185

A.55 Emission likelihoods of the production state children of the "watch TV" submodel in the "watch television" model stored at the children states of q_3^2. 185

A.56 Initial state likelihoods stored at the root node in the "read book on couch" model. 186

A.57 State transition likelihoods governing the second layer of the "read book on couch" model stored at the root node. 186

A.58 Initial state likelihoods for the "read on couch" submodel in the "read book on couch" model stored at state q_1^2. 186

A.59 State transition likelihoods governing the production state children of the "read on couch" submodel in the "read book on couch" model stored at state q_1^2. 187

A.60 Emission likelihoods of the production state children of the "read on couch" submodel in the "read book on couch" model stored at the children states of q_1^2. 187

A.61 Initial state likelihoods for the "enter via bookcase" submodel in the "read book on couch" model stored at state q_2^2. 187

A.62 State transition likelihoods governing the production state children of the "enter via bookcase" submodel in the "read book on couch" model stored at state q_2^2. 188

A.63 Emission likelihoods of the production state children of the "enter via bookcase" submodel in the "read book on couch" model stored at the children states of q_2^2. 188

A.64 Initial state likelihoods for the "exit via bookcase" submodel in the "read book on couch" model stored at state q_3^2. 188

A.65 State transition likelihoods governing the production state children of the "exit via bookcase" submodel in the "read book on couch" model stored at state q_3^2. 189

A.66 Emission likelihoods of the production state children of the "exit via bookcase" submodel in the "read book on couch" model stored at the children states of q_3^2. 189

A.67 Initial state likelihoods stored at the root node in the "eat dinner" model. 190

A.68 State transition likelihoods governing the second layer of the "eat dinner" model stored at the root node. 190

A.69 Initial state likelihoods for the "enter/exit" submodel in the "eat dinner" model stored at state q_1^2. 190

A.70 State transition likelihoods governing the production state children of the "enter/exit" submodel in the "eat dinner" model stored at state q_1^2. 191

A.71 Emission likelihoods of the production state children of the "enter/exit" submodel in the "eat dinner" model stored at the children states of q_1^2. 191

A.72 Initial state likelihoods for the first "eat dinner" submodel in the "eat dinner" model stored at state q_2^2. 191

A.73 State transition likelihoods governing the production state children of the first "eat dinner" submodel in the "eat dinner" model stored at state q_2^2. 192

A.74 Emission likelihoods of the production state children of the first "eat dinner" submodel in the "eat dinner" model stored at the children states of q_2^2. 192

A.75 Initial state likelihoods for the second "eat dinner" submodel in the "eat dinner" model stored at state q_3^2. 192

A.76 State transition likelihoods governing the production state children of the second "eat dinner" submodel in the "eat dinner" model stored at state q_3^2. 193

A.77 Emission likelihoods of the production state children of the second "eat dinner" submodel in the "eat dinner" model stored at the children states of q_3^2. 193

A.78 Initial state likelihoods stored at the root node in the "eat dinner while watching TV" model. 194

A.79 State transition likelihoods governing the second layer of the "eat dinner while watching TV" model stored at the root node. 194

A.80 Initial state likelihoods for the "eat dinner" submodel in the "eat dinner while watching TV" model stored at state q_1^2. 194

A.81 State transition likelihoods governing the production state children of the "eat dinner" submodel in the "eat dinner while watching TV" model stored at state q_1^2. 195

A.82 Emission likelihoods of the production state children of the "eat dinner" submodel in the "eat dinner while watching TV" model stored at the children states of q_1^2. 195

A.83 Initial state likelihoods for the "enter/exit" submodel in the "eat dinner while watching TV" model stored at state q_2^2. 195

A.84 State transition likelihoods governing the production state children of the "enter/exit" submodel in the "eat dinner while watching TV" model stored at state q_2^2. 196

A.85 Emission likelihoods of the production state children of the "enter/exit" submodel in the "eat dinner while watching TV" model stored at the children states of q_2^2. 196

A.86 Initial state likelihoods for the "walk by TV" submodel in the "eat dinner while watching TV" model stored at state q_3^2. 196

A.87 State transition likelihoods governing the production state children of the "walk by TV" submodel in the "eat dinner while watching TV" model stored at state q_3^2. 197

A.88 Emission likelihoods of the production state children of the "walk by TV" submodel in the "eat dinner while watching TV" model stored at the children states of q_3^2. 197

A.89 Initial state likelihoods stored at the root node in the "there is nothing good on TV" model. 198

A.90 State transition likelihoods governing the second layer of the "there is nothing good on TV" model stored at the root node. 198

A.91 Initial state likelihoods for the "enter via TV" submodel in the "there is nothing good on TV" model stored at state q_1^2. 198

A.92 State transition likelihoods governing the production state children of the "enter via TV" submodel in the "there is nothing good on TV" model stored at state q_1^2. 199

A.93 Emission likelihoods of the production state children of the "enter via TV" submodel in the "there is nothing good on TV" model stored at the children states of q_1^2. 199

A.94 Initial state likelihoods for the "exit via bookcase" submodel in the "there is nothing good on TV" model stored at state q_2^2. 199

A.95 State transition likelihoods governing the production state children of the "exit via bookcase" submodel in the "there is nothing good on TV" model stored at state q_2^2. 200

A.96 Emission likelihoods of the production state children of the "exit via bookcase" submodel in the "there is nothing good on TV" model stored at the children states of q_2^2. 200

A.97 Initial state likelihoods for the "find book / use couch" submodel in the "there is nothing good on TV" model stored at state q_3^2. 200

A.98 State transition likelihoods governing the production state children of the "find book / use couch" submodel in the "there is nothing good on TV" model stored at state q_3^2. 201

A.99 Emission likelihoods of the production state children of the "find book / use couch" submodel in the "there is nothing good on TV" model stored at the children states of q_3^2. 201

List of Algorithms

2.1 Apriori 21

2.2 FP-Tree Construction 23

2.3 FP-Growth 25

2.4 E-Apriori 29

2.5 FITI 31

2.6 FITI-Gen 32

4.1 EFP-Tree Construction 87

4.2 EFP-Growth 100

Chapter 1

Introduction

Current trends suggest that the global population will consist predominately of older people, those aged sixty and over, in as little as fifty years (United Nations Population Division, 2002). This demographic shift is expected to lead to an increase in demand for care of the elderly and, as a result, in the need for smart homes; intelligent environments that are able to assist its occupants in maintaining independent lifestyles for as long as possible (Mynatt et al., 2000). Several such projects are currently underway. The MavHome (Cook et al., 2003) project and the Adaptive House (Mozer, 2004) both seek to develop home automation that does not require users to manually program the components in their homes. The guiding philosophy behind these projects is that a true automated home is an environment that is capable of learning, recognising and predicting its occupants' behaviour, able to reason about the state of the home and to adapt itself over time to the changing needs of its occupants. Researchers at the Aware Home (Kidd et al., 1999) are exploring issues such as context aware computing, human-computer-interaction and occupant tracking and identification with an application to caring for the elderly. Recently, the opening of the PlaceLab (Intille et al., 2005) was announced. This project provides a live in laboratory facility for multidisciplinary research into areas such as human behaviour and the use of technology to simplify home control and promote healthy living.

The work detailed in this thesis is motivated by a desire to create homes that are able to detect the presence of new, possibly abnormal, behaviour in their occupants and to take action accordingly. Appropriate action may be to query the occupant on the new behaviour, to jog their memory on a task that they were carrying out or even to alert an occupant's relative that their assistance is required. The central issue faced here is the identification of abnormality present in behaviours observed through visual and event driven sensory data obtained from pressure mat sensors and reed switches. Existing approaches to accomplish this vary from statistical analysis of the time and frequency with which behaviours or events are seen (Hauptmann et al., 2004) to the deployment of stand-alone devices that recognise specific anomalies such as falls (Sixsmith and Johnson, 2004).

This thesis details two different approaches to abnormality detection that build representations of normality with which incoming data from a large spatio-temporal search space can be examined for behavioural anomalies. The first method employs the use of stochastic models to encode sequences of observations returned by a visual tracking system monitoring a subject in their home. In the second, data mining techniques are employed to distill the associative relationship of sensor events recorded in a smart home into a set of core rules that are representative of new and changing behaviours. An accompanying visual data mining interface with which discovered emergent behaviours can be interpreted is also discussed.

1.1 Aims and Approach

This thesis examines the problem of detecting abnormality in human behaviour in a smart home environment. The objectives are to:

- Investigate stochastic models representative of normal human behaviours from observations gathered by a visual tracking system and to apply these models to the task of recognising normal behaviours and hence as classifiers of abnormality.
- Explore data mining techniques for making sense of sensor event data captured from an array of sensors deployed within a smart home environment.

The first part of this thesis tackles the issue of modelling behaviours using stochastic models. Here, application of the Hidden Markov Model (HMM) is an obvious choice given the spatio-temporal nature of the problem domain. The large state space required to train HMMs on large scale daily activities is undesirable, however, given that the hierarchical nature of human behaviour suggests that the problem domain may be decomposed into increasingly fine grained activities down to a level of observable atomic events (Zacks and Tversky, 2001). Work into the application of stochastic models therefore begins with an examination of the suitability of the Hierarchical Hidden Markov Model (HHMM) to learning multi-level representations of sequences of human activity gathered using a visual tracker. Of interest here is the ability of the HHMM to learn the higher level relationships between atomic observations such that the resulting models may be reused in the construction of increasingly complex hierarchical representations of large scale human activity. The HHMM offers the ability to learn behavioural patterns such that their structural meaning can be found and used to aid interpretation of the trained models.

Activity duration is then considered. Duration is an important feature in the recognition of human behaviour and in the identification of anomalies. It is not, however, explicitly modelled by the standard HMM and HHMM. Although these models are able to accommodate different durations, the lack of an explicit duration model suggests that the HMM and the HHMM do not map well to the abnormality detection domain when the abnormality is caused by unusual duration in activity. The importance of incorporating duration in a model of human behaviour is therefore investigated using activity sequences where the ordering of events is similar yet the duration of the activities differs. The standard HMM and the Explicit State Duration HMM (ESD-HMM), also known as the Hidden Semi-Markov Model (HSMM), are investigated both in terms of their ability to recognise known normal behaviours and as classifiers of abnormality caused by unusual activity durations.

An extension of the ESD-HMM in which the state durations are given by an observation signal augmented with the state transition times is then examined. The resulting Observed Time Indices ESD-HMM (OTI ESD-HMM) offers the ability to incorporate duration into the HMM using the state transition times obtained from strategically deployed pressure mat sensors. Of interest here is whether the OTI ESD-HMM is able to offer any advantages over the standard ESD-HMM when used to model observed human behaviour and in the detection of abnormality.

The specific aims of this first part of the thesis are to:

- Investigate the construction of hierarchical representations of simple activities that are suitable for reuse as the atomic elements of higher level long term behaviours.
- Explore how the lack of an explicit model of duration reduces the practical application of the HMM and, by extension, the HHMM for modelling human behaviour.
- Explore the extension of the ESD-HMM to exploit observable activity durations.
- Investigate the application of the OTI ESD-HMM to real world data and compare its performance in behaviour recognition and abnormality detection against that

of the HMM.

The second part of the thesis concentrates on the application of Intertransaction Association Rules (IARs) (Lu et al., 1998) to mine the frequently occurring associative relationships between sensor events that are triggered in a home as a person goes about their daily routines. IARs are implication rules that capture the non-sequential associations between events occurring within a single transaction, or time interval. IARs retain some of the higher level temporal context in which the events occur in the form of the intertransaction relationships between frequent intratransaction itemsets.

Work in IAR mining begins with the introduction of a new algorithm for IAR mining that is more computationally efficient than previous methods. The Extended FP-Growth (EFP-Growth) algorithm and its accompanying Extended FP-Tree (EFP-Tree) data structure are proposed as extensions of the Frequent Pattern Tree (FP-Tree) and the Frequent Pattern Growth (FP-Growth) algorithm (Han et al., 2000, 2004). The EFP-Tree allows the pattern growth property, a divide and conquer technique that avoids candidate rule generation, to be applied to the discovery of intertransaction associations. Emergent IAR mining is then introduced as a technique for finding those rules whose frequent presence in a set of new sensor data is unexpected given a historical data set. The purpose here is to identify those associations whose presence signals a new or unusually frequent behaviour.

A visual data mining interface that enables users to interpret discovered rules without requiring the sensor event logs to be manually inspected is then proposed. The interface aims to reduce the burden of an otherwise cumbersome task by permitting users to visualise emergent rules in the context of the original sensor data.

The data mining specific aims are to:

- Investigate a new algorithm for IAR mining that eliminates the need for the computationally expensive candidate generation and testing procedure employed by current levelwise IAR mining algorithms.
- Identify the core set of rules whose presence may indicate abnormality.
- Present these rules in a format that allows for easy interpretation while eliminating the need for manual inspection of the rules by trawling through the original sensor event logs.

1.2 Significance and Contributions

The contributions of the stochastic models work in this thesis are:

- An investigation into the use of stochastic models for abnormality detection in human behaviours. It is shown that the HHMM is suitable for construction of multi-level models of human activity sequences such that behavioural subpatterns are represented by HHMM submodels.
- Incorporation of duration in models of human behaviour as a measure of abnormality. To this end, an extension of the Explicit State Duration HMM that allows known state durations to be used in model training and inferencing is provided. The Observed Time Indices ESD-HMM is introduced and shown to be superior to the standard ESD-HMM due to its ability to more accurately represent sequences of activity upon which it is trained.
- The performance of the models are compared using real world data gathered by a visual tracker deployed in a volunteer subject's home.

The data mining part of the thesis makes the following contributions:

- A new algorithm for Intertransaction Association Rule (IAR) mining is presented. The proposed Extended FP-Tree and Extended FP-Growth algorithms avoid the stepwise candidate generation approach employed in current IAR mining algorithms. Experimental results on both synthetic and real world data demonstrates an order of magnitude improvement in the computational complexity of the new algorithm over existing techniques.
- A novel application of IAR mining for abnormality detection is introduced via the mining of emergent behaviours by identifying rules that are representative of activity which has not been seen previously or whose frequency is unusually high given known past activities. Real world data is used to demonstrate the practical application of emergent behaviour mining by identifying both sensor aberrations and new and changing behaviours.
- A novel visual data mining technique is proposed to aid the interpretation of emergent behaviours. The visual interface provides a significantly simpler means of interpreting emergent behaviours by mapping these behaviours back onto the space of the original sensor data. Rule interpretations made using the visual

interface are compared with those interpretations gathered via manual inspection of the rules using the sensor event logs. Observations made demonstrate the benefits of the visual data mining technique over the manual inspection process.

1.3 Structure of the Thesis

The structure of this thesis is as follows. A review of related work is presented in Chapter 2 beginning with an overview of the HMM, HHMM and ESD-HMM graphical models. Current data mining algorithms and visual data mining techniques pertinent to the thesis are then examined.

Chapter 3 investigates the suitability of the HHMM to the problem of multi-level modelling of human behaviours and examines the importance of incorporating duration in models of human activity. An extension to the ESD-HMM in which the state transition times are provided by an observation signal augmented with pressure mat sensor information is then proposed. The feasibility of using these models in the real world is tested with a deployment of the tracking and inferencing system in the home of a volunteer subject.

Contributions made in the field of data mining are presented in Chapter 4. An extension of the Frequent Pattern Tree (FP-Tree) allowing IARs to be mined using the pattern growth property is proposed. An application of IAR mining to the discovery of emergent human behaviours is then presented using sensor event log data recorded in the homes of two volunteer subjects.

A visual data mining interface for exploring emergent behaviours is introduced along with a discussion of the design rationale in Chapter 5. The real world sensor event logs are analysed again in a repeat of the experimentation in Chapter 4. Results and observations made using the visual data mining tool are contrasted to the observations made in Chapter 4 via manual inspection of the emergent IARs in the context of the event logs.

Finally, a concluding summary of the work of this thesis is presented in Chapter 6 with ideas for possible future work.

Chapter 2

Related Work

The related works upon which this thesis is founded originate from two areas within the field of computer science: stochastic models and data mining. Although differing, both stochastic models and data mining offer techniques and approaches that can be applied to the problem of discovering abnormality in human behaviours. The works that are considered to be most pertinent to this thesis from both of these research areas will be reviewed in this chapter.

The stochastic models component of the review begins with an introduction to the Hidden Markov Model (HMM) in Section 2.1. A brief overview of the classic problems of inferencing, decoding and parameter learning is given. An introduction to the Explicit State Duration HMM, known also as the Hidden Semi-Markov Model, and the Hierarchical HMM is given in Section 2.2 and Section 2.3 respectively.

Review of the data mining literature begins in Section 2.4 with a look at the problem of association rule mining. Two seminal algorithms for association rule mining, Apriori (Agrawal et al., 1993) and Frequent Pattern Growth (Han et al., 2000), are respectively detailed in Section 2.4.1 and Section 2.4.2. Intertransaction association rule mining, an extension of traditional association rule mining that considers relationships among items or events between transactions, will be considered in Section 2.5 with a detailed examination of the E-Apriori algorithm in Section 2.5.1 and the First Intra Then Inter (FITI) algorithm in Section 2.5.2. Finally, an overview of related work in visualisation and exploration of discovered association rules is provided in Section 2.6. A summary of this review is then presented in Section 2.7.

2.1 Hidden Markov Model

The Hidden Markov Model (HMM) (Rabiner, 1989) is a first order Markov process whose state layer is not directly observable. Rather, the state of the model and its parameters are probabilistically deduced from an observable signal. HMMs have, since their inception in the 1960s, been applied to a diverse range of fields from speech recognition, activity recognition, handwriting recognition and signal processing (Cappé, 2001).

Following the notation of Rabiner (1989), a HMM of N hidden states is said to generate a sequence of observations $O = [o_1, o_2, \ldots o_t, \ldots o_T]$ over time t from a codebook $V = \{v_1, v_2, \ldots v_k, \ldots V_M\}$ of M possible discrete features using the parameter set $\lambda = (\Pi, A, B)$. The parameter $\Pi = \{\pi_i\}$ is the initial state distribution; π_i being the probability that the i^{th} state in the model will be activated at time $t = 1$. The state transition probabilities are defined by $A = \{a_{ij}\}$ such that the probability of transitioning to a state j from state i is a_{ij}. The observation emission probabilities are given as $B = \{b_i(k)\}$ where $b_i(k)$ is the likelihood that the i^{th} state will generate an observation V_k. Finally, the states activated at each discrete time period t are given as $Q = [q_1, q_2, \ldots q_t, \ldots q_T]$. A Finite State Machine (FSM) representation of a HMM is given in Figure 2.1. A graphical representation of a HMM rolled out over time is given in Figure 2.2.

Three well-known issues are associated with HMMs. The first, inferencing, in Section 2.1.1 is concerned with computing the probability that a model will generate a given observation sequence. The decoding problem in Section 2.1.2 seeks to find an optimal state sequence that best "explains" an observation sequence. The final HMM problem, training, is concerned with the learning of the model parameters to optimise the probability $\Pr(O|\lambda)$. A well-known means by which to do so is presented in Section 2.1.3.

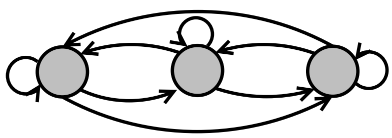


Figure 2.1: Finite state machine representation of a three state HMM showing the possible state transitions.

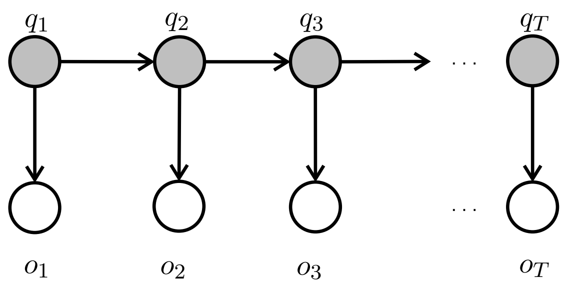


Figure 2.2: A HMM rolled out over time. At each discrete time interval t an observation o_t is emitted by state q_t to produce the observation sequence $O = \{o_1, o_2, \ldots o_t, \ldots o_T\}$.

2.1.1 Inferencing

Inferencing is an important task in the HMM. Its solution enables the use of the model in classification; sequence classification becoming a problem of finding the HMM most likely to generate a given sequence from a set of HMMs representative of the available classes. It is also a first step towards resolution of the decoding and training problems.

Finding the probability of $\Pr(o_1 \ldots o_T|\lambda)$ is achieved via the recursive *forwards* procedure

$$\alpha_t(i) = \Pr(o_1 \ldots o_t, q_t = i|\lambda) \tag{2.1}$$

which is calculated as:

$$\alpha_1(i) = \pi_i b_i(o_1) \tag{2.2}$$

$$\alpha_t(i) = \left[\sum_{j=1}^{N} \alpha_{t-1}(j)\, a_{ji}\right] b_i(o_t). \tag{2.3}$$

The probability of a sequence O being generated is found by summing over the possible N states the model may be in at time T:

$$\Pr(O|\lambda) = \sum_{i=1}^{N} \alpha_T(i). \tag{2.4}$$

2.1.2 Decoding

Decoding seeks to find the state sequence that most likely explains an observation sequence. It requires the probability $\Pr(q_t = i|o_t, \lambda)$, the likelihood of the model being in a given state at time t, to be computed. An additional *backwards* smoothing component β is required to find

$$\beta_t(i) = \Pr(o_{t+1} \dots O_T | q_t = i, \lambda) \tag{2.5}$$

which can be recursively computed by:

$$\beta_T(i) = 1 \tag{2.6}$$

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(o_{t+1}) \beta_{t+1}(j). \tag{2.7}$$

The probability $\Pr(q_t = i|o_t, \lambda)$ can then be computed using the Forwards-Backwards procedure also known as the Baum-Welch algorithm (Baum et al., 1970):

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{\sum_{i=1}^{N} \alpha_t(i)\beta_t(i)}. \tag{2.8}$$

The optimal state at time t is found by finding the state q_t that returns the highest likelihood:

$$q_t = \underset{1\le i\le N}{\operatorname{argmax}} \Pr(q_t = i|o_t, \lambda) \tag{2.9}$$

$$= \underset{1\le i\le N}{\operatorname{argmax}} [\gamma_t(i)]. \tag{2.10}$$

A procedure for finding the optimal state sequence will need to consider the optimal state q_t with respect to the optimal path leading to q_{t-1}. This can be handled via the

Viterbi algorithm (Forney, 1973):

$$\begin{aligned}
\delta_1(i) &= \pi_i b_i(o_1) && (2.11)\\
\psi_1(i) &= 0 && (2.12)\\
\delta_t &= \max_{1\le j\le N}\left[\delta_{t-1}(j)\, a_{ji}\right] b_i(o_t) && (2.13)\\
\psi_t(i) &= \underset{1\le j\le N}{argmax}\left[\delta_{t-1}(j)\, a_{ji}\right] && (2.14)\\
p^* &= \max_{1\le i\le N}\left[\delta_T(i)\right] && (2.15)\\
q_T^* &= \underset{1\le i\le N}{argmax}\left[\delta_T(i)\right] && (2.16)
\end{aligned}$$

where $\delta_t(i)$ is the highest likelihood for a path at time t that explains the observations $o_1 \dots o_T$ and where $\psi_t(i)$ is the state at time $t-1$ that maximised the likelihood of reaching state i at time t. The complete optimal state sequence can then be found by backtracking from p^* and q^*.

2.1.3 Training

The HMM model parameters are typically optimised for a given observation sequence, or set of sequences, using Expectation Maximisation (EM) via the Baum-Welch algorithm. Model parameters are randomly initialised and then adapted towards the training data using the EM two-step iterative process; the first step calculates the expected sufficient statistics (ESS) for each parameter in the model; the second step maximises the ESS by normalising the parameter probability.

The parameter estimation formulae for the HMM are:

$$\begin{aligned}
\hat{\pi}_i &= \frac{\pi_i b_i(o_1)\,\beta_1(i)}{\Pr(O|\lambda)} && (2.17)\\
\hat{a}_{ij} &= \frac{\sum_{t=1}^{T-1}\alpha_t(i)\, a_{ij} b_j(o_{t+1})\,\beta_{t+1}(j)}{\sum_{j=1}^{N}\sum_{t=1}^{T-1}\alpha_t(i)\, a_{ij} b_j(o_{t+1})\,\beta_{t+1}(j)} && (2.18)\\
\hat{b}_i(k) &= \frac{\sum_{\substack{t=1\\ \text{s.t. } o_t=v_k}}^{T-1}\alpha_t(i)\,\beta_t(i)}{\sum_{t=1}^{T-1}\alpha_t(i)\,\beta_t(i)} && (2.19)
\end{aligned}$$

2.1.4 Application to Activity Recognition

Hidden Markov Models have been employed in human activity recognition. Notable examples include the work of Yamato et al. (1992) who recognised tennis strokes from foreground segmented video footage. The video was sampled to provide only a few images per sequence for use in training and recognition. A grid pattern was used to divide up the images such that the ratio of foreground to background pixels in each grid could be used as features in the HMM.

Coupled Hidden Markov Models (CHMMs), extensions of the HMM for modelling independent yet interacting processes, were used in Oliver et al. (2000) to model the interactions of people in an outdoor scene observed by a visual tracker. Models were first trained on data from synthetic agents and then updated using real world data. Results demonstrated that the models were able to detect and classify human interactions with good results. CHMMs were earlier used to model T'ai Chi gestures by Brand et al. (1997).

The use of layered HMMs to monitor the behaviour of people working in an office environment was introduced in Oliver et al. (2002). Here, the outputs from lower level HMMs were used as input into higher layer HMMs to distill the relationships of the lower layers into increasingly granular representations of the activities being modelled.

Ivanov and Bobick (2000) employed Stochastic Context Free Grammars (SCFGs) to recognise activities. HMMs were used to detect the occurrence of low-level events representing the primitives used in a grammar parser and applied to both gesture recognition and outdoor surveillance.

The work of Brand and Kettnaker (2000) demonstrated the application of entropy minimisation in HMM parameter learning with trained models featuring clearly recognisable structure that mimicked the observation signal more closely than conventional EM. The entropically estimated models were applied alongside the HMM to the problem of event recognition and abnormality detection in video footage from both an office environment and an outdoor traffic scene. In the former, the descriptive features of an ellipse drawn around the foreground segmented blob of a single office worker was used to train the models and to classify between a variety of office activities. Abnormal behaviours in the form of an activity being acted out in reverse and jittery behaviour due to excessive consumption of coffee by the test subject were able to be more reliably detected by the entropically trained HMM than by the standard model. The former model was also shown to have sound application in the detection of abnormality using flow vectors to learn the normal behaviour of vehicles and pedestrians in a traffic surveillance scenario.

A review of earlier works on activity recognition involving HMMs with a focus on motion analysis and gesture recognition can be found as part of a review of motion analysis by Aggarwal and Cai (1999).

2.2 Explicit State Duration HMM

Explicit state duration modelling (Ferguson, 1980; Russell and Moore, 1985) was introduced into the HMM to more accurately capture the temporal structure of speech than is possible with the standard HMM. Duration is incorporated into the model via the variable $p_i\left(d\right)$ such that $1 \leq d \leq D$ where D constrains the maximum duration. In its non-parametric form, p_i is a vector of discrete duration probabilities such that $\sum_{d=1}^{D} p_i\left(d\right) = 1$. The self transition probabilities are set so that $a_{ii} = 0$. A graphical representation of the Explicit State Duration HMM (ESD-HMM), alternatively known as the Hidden Semi-Markov Model (HSMM), in which the model is shown rolled out over time is given in Figure 2.3. The FSM representation of the model remains the same as for the HMM depicted in Figure 2.1. The generative process is similar to that of the HMM with the addition that the number of discrete-time steps a state i will remain active prior to making a transition to another state j $(i \neq j)$ is selected from the state duration distribution $P = \{P_i\left(d\right)\}$ upon activation. For clarity, the notation $\lambda = (\Pi, A, B, P)$ is used to represent the ESD-HMM parameters.

The Forwards-Backwards variables differ slightly to those of the HMM to accommodate

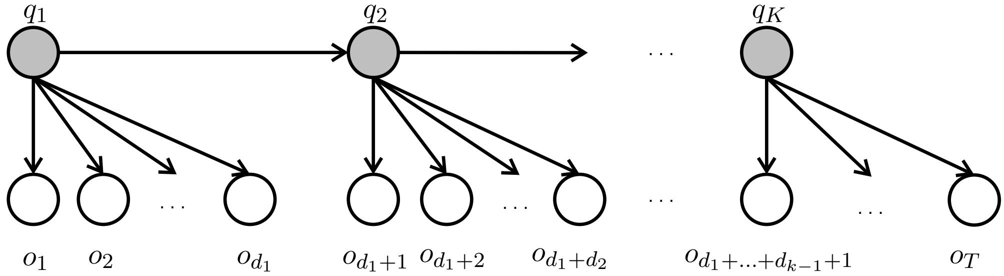


Figure 2.3: A ESD-HMM rolled out over time. At each discrete time interval the t^{th} observation o_t is emitted by a state q_k to produce the observation sequence $O = [o_1, o_2, \ldots o_t, \ldots o_T]$. A state q_k is active for duration d_k prior to making a transition to another state.

the starting and ending times of the hidden states (Rabiner, 1989):

$$\alpha_t(i) = \Pr(o_1 \ldots o_t, q_i \text{ ends at } t|\lambda) \tag{2.20}$$

$$\alpha_t^*(i) = \Pr(o_1 \ldots o_t, q_i \text{ begins at } t+1|\lambda) \tag{2.21}$$

$$\beta_t(i) = \Pr(o_{t+1} \ldots o_T|q_i \text{ ends at } t, \lambda) \tag{2.22}$$

$$\beta_t^*(i) = \Pr(o_{t+1} \ldots o_T|q_i \text{ begins at } t+1, \lambda). \tag{2.23}$$

The model re-estimation formulas for the Explicit State Duration HMM (ESD-HMM) are similarly presented:

$$\hat{\pi}_i = \frac{\pi_i \beta_0^*(i)}{P(O|\lambda)} \tag{2.24}$$

$$\hat{a}_{ij} = \frac{\sum_{t=1}^{T} \alpha_t(i) a_{ij} \beta_t^*(j)}{\sum_{j=1}^{N} \sum_{t=1}^{T} \alpha_t(i) a_{ij} \beta_t^*(j)} \tag{2.25}$$

$$\hat{b}_i(k) = \frac{\sum_{\substack{t=1 \\ \text{s.t. } o_t = k}}^{T} \left[\left(\pi_i \beta_0^*(i) + \sum_{\tau=1}^{t-1} \alpha_\tau^*(i) \beta_\tau^* \right) - \left(\sum_{\tau=1}^{t-1} \alpha_\tau(i) \beta_\tau(i) \right) \right]}{\sum_{k=1}^{M} \sum_{\substack{t=1 \\ \text{s.t. } o_t = v_k}}^{T} \left[\left(\pi_i \beta_0^*(i) + \sum_{\tau=1}^{t-1} \alpha_\tau^*(i) \beta_\tau^* \right) - \left(\sum_{\tau=1}^{t-1} \alpha_\tau(i) \beta_\tau(i) \right) \right]} \tag{2.26}$$

$$\hat{p}_i(d) = \frac{\pi_i p_i(d) \beta_d(i) \prod_{s=1}^{d} b_i(o_s) + \sum_{t=1}^{T-d} \alpha_t^*(i) p_i(d) \beta_{t+d}(i) \prod_{s=t+1}^{t+d} b_i(o_s)}{\sum_{d=1}^{D} \left[\pi_i p_i(d) \beta_d(i) \prod_{s=1}^{d} b_i(o_s) + \sum_{t=1}^{T-d} \alpha_t^*(i) p_i(d) \beta_{t+d}(i) \prod_{s=t+1}^{t+d} b_i(o_s) \right]}. \tag{2.27}$$

Unfortunately, the discrete state duration probabilities defined by P requires that a significant amount of data be available to adequately train the model. Identifying this limitation, Levinson (1986) proposed a model using the continuous gamma distribution while Mitchell and Jamieson (1993) suggested that the exponential family be used. Recently, the application of the Coxian distribution was demonstrated as a means of modelling the state durations in the switching HSMM (Duong et al., 2005), a special case of the two layer Hierarchical HMM.

Variations of the ESD-HMM in which the duration of state occupancy conditions the state transition likelihoods (Sin and Kim, 1995; Vaseghi, 1995) and the observation emission likelihoods (Park et al., 1996) have also been introduced.

2.3 Hierarchical HMM

The HHMM was first proposed in Fine et al. (1998) and is a special case of a SCFG. The authors applied the model to handwriting detection, demonstrating that a HHMM trained on a single handwritten word is able to learn the hierarchical nature of the training data by populating the lowest states in the model topology with observations mapping to atomic strokes, compound strokes representing letters and combinations of letters at the middle layers and finally the word itself at the top level.

The HHMM as used in this thesis will be formally introduced following the notation presented in Fine et al. (1998). An observation sequence $\overline{O} = [o_1, o_2, \ldots, o_T]$ is defined to be a finite length string from all possible strings $\sum^*$ from the finite alphabet $\sum$. States within the HHMM are represented by q_i^d where $d \in [1, 2, \ldots, D-1, D]$ denotes the hierarchy level and i the state index relative to the parent. The state index may be omitted if it is clear which state is being referred to. States are one of three types: internal, end or production. Internal states are themselves HHMMs and may have an arbitrary number of children states, the number of non-end sub-states of state q_i^d being denoted by $|q_i^d|$. Production and end states do not have children. Vertical and horizontal transition probabilities are defined for each internal state as the vector Π^{q^d} and the matrix A^{q^d} respectively. An internal state must always perform a vertical transition down to one of its children before a horizontal transition may be made with control of the transitions returning to the calling state only when a lower state has made a horizontal transition to an end state. The end state is a special token state that exists only to signal when an upwards vertical transition is to be made. The probability of state q^{d-1} vertically transitioning to sub-state q_i^d is specified as $\pi^{q^{d-1}}\left(q_i^d\right)$ while the probability of state q_i^d making a horizontal transition to state q_j^d is written as $a_{ij}^{q^d}$. Production nodes are the only states within the HHMM that emit observations and are much like the states of a HMM. The discrete probability density function (PDF) of the production nodes is represented as the vector B^{q^D} which defines the probability of state q^D producing observation v_k as $b^{q^D}(k)$. The model parameters are denoted in the compact form $\lambda = (\Pi, A, B)$. Figure 2.4 shows a FSM representation of a HHMM while Figure 2.5 depicts a HHMM that has been unrolled over time.

To compliment the forward α and backward β path variables of the HMM, the HHMM introduces the variables χ and ξ corresponding to the downward and upward transition probabilities respectively. The notation and meaning of the path variables is given by:

$$\alpha\left(t, t+k, q_i^d, q^{d-1}\right) = \Pr\left(o_t \ldots o_{t+k}, q_i^d \text{ completed at } t+k | q^{d-1} \text{ started at } t\right) \tag{2.28}$$
$$\beta\left(t, t+k, q_i^d, q^{d-1}\right) = \Pr\left(o_t \ldots o_{t+k}, | q_i^d \text{ started at } t, q^{d-1} \text{ completed at } t+k\right) \tag{2.29}$$
$$\xi\left(t, q_i^d, q_j^d, q^{d-1}\right) = \Pr\left(o_1 \ldots o_t, q_i^d \text{ transitions to } q_j^d, o_{t+1} \ldots o_T | \lambda\right) \tag{2.30}$$
$$\chi\left(t, q_i^d, q^{d-1}\right) = \Pr\left(o_1 \ldots o_{t-1}, q^{d-1} \text{ transitions to } q_i^d, o_t \ldots o_T | \lambda\right). \tag{2.31}$$

The auxiliary variables γ_{in} and γ_{out} are defined as follows:

$$\gamma_{\text{in}}\left(t, q_i^d, q^{d-1}\right) = \sum_{k=1}^{\left|q^{d-1}\right|} \xi\left(t-1, q_k^d, q_i^d, q^{d-1}\right) \tag{2.32}$$
$$\gamma_{\text{out}}\left(t, q_i^d, q^{d-1}\right) = \sum_{k=1}^{\left|q^{d-1}\right|} \xi\left(t-1, q_i^d, q_k^d, q^{d-1}\right). \tag{2.33}$$

Readers are invited to refer to Fine et al. (1998) for the complete definitions and derivations of the path variables and their auxiliary variables.

The HHMM re-estimation formulas required for training are:

$$\hat{\pi}^{q^1}\left(q_i^2\right) = \frac{\chi\left(1, q_2^2, q^1\right)}{\sum_{i=1}^{|q_i^1|} \chi\left(1, q_2^2, q^1\right)} \tag{2.34}$$

$$\hat{\pi}^{q^{d-1}}\left(q_i^d\right) = \frac{\sum_{t=1}^{T} \chi\left(t, q_i^d, q^{d-1}\right)}{\sum_{i=1}^{|q^{d-1}|} \sum_{t=1}^{T} \chi\left(t, q_i^d, q^{d-1}\right)} \tag{2.35}$$

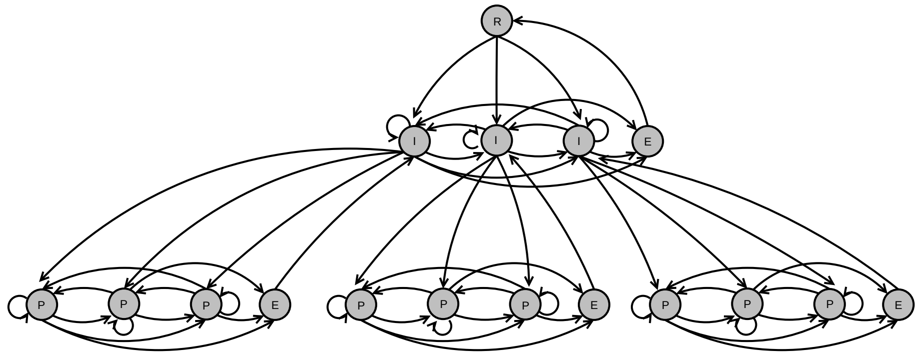


Figure 2.4: Finite state machine representation of a HHMM showing the possible state transitions between the root (R) node, the internal (I) states and the production (P) states.

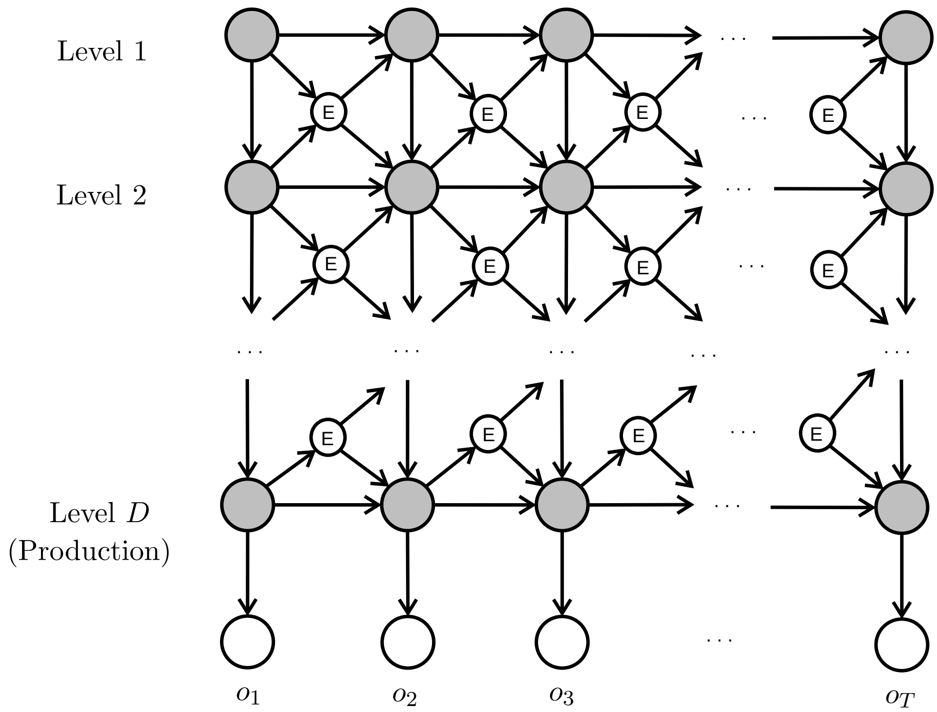


Figure 2.5: A HHMM rolled out over time. At each discrete time interval t the observation o_t is emitted by a production state q_i^D to produce the observation sequence $O = [o_1, o_2, \ldots o_t, \ldots o_T]$. Control returns to a state q_i^d when one of its substates q_j^{d+1} makes a transition to an end state. End states are marked with an E.

$$\hat{a}_{ij}^{q^{d-1}} = \frac{\displaystyle\sum_{t=1}^{T} \xi\left(t, q_i^d, q^{d-1}\right)}{\displaystyle\sum_{t=1}^{T} \gamma_{\text{out}}\left(t, q_i^d, q^{d-1}\right)} \tag{2.36}$$

$$\hat{b}_{q_i^D}^{q^{D-1}}\left(v_k\right) = \frac{\displaystyle\sum_{\substack{t=1 \\ \text{s.t. } o_t=v_k}}^{T} \chi\left(q_i^D, q^{D-1}\right) + \sum_{t>1, o_t=v_k} \gamma_{\text{in}}\left(t, q_i^D, q^{D-1}\right)}{\displaystyle\sum_{t=1}^{T} \chi\left(q_i^D, q^{D-1}\right) + \sum_{t>1, o_t=v_k} \gamma_{\text{in}}\left(t, q_i^D, q^{D-1}\right)}. \tag{2.37}$$

The concept of hierarchical HMMs has since been applied in the extension of Partially Observable Markov Decision Processes (POMDPs) to Hierarchical POMDPs (HPOMDPs) and used in indoor environment learning for robot navigation (Theocharous, 2002). Murphy and Paskin (2001) have shown how a Dynamic Bayesian Network (DBN) representation of the HHMM can be utilised to reduce the inference complexity of the finite state machine (FSM) HHMM from $O\left(T^3\right)$ down to $O\left(T\right)$ for use in situations where exponential complexity with regards to the model depth is tolerable. The HHMM was recently extended in Phung et al. (2004) to allow for the sharing of substructures to reduce the computational requirements for inferencing and training with improved parameter learning accuracy while requiring less training data than the standard HHMM.

Increased accuracy in information extraction from technical documents was the motivating goal in Skounakis et al. (2003) who presented an extension to the HHMM termed Context HHMM (CHHMM) for capturing the grammatical structure in biomedical texts. CHHMM modeled the relationships between domain specific and neighbouring words which could then be used to extract meaning from phrases with a higher precision and recall than previously proposed methods. A Monte Carlo approach to unsupervised learning of the hierarchical HMM topography with an application to learning the statistical structure of soccer videos has been discussed in Xie et al. (2003).

Similar to the HHMM, the Abstract Hidden Markov model (AHMM) (Bui et al., 2001, 2002) is a hierarchical stochastic model for representing a hierarchical abstraction of an agent's state and goals at varying levels of detail. The AHMM extends the HHMM by allowing the refinement of a layer into sequences at the lower level to be dependent on the current "environment" state. This allows for simple types of context-sensitive behaviours to be modelled. Inference scalability in the AHMM is attained by limiting interaction between chains to only those directly above or below it. The AHMM has recently been applied to tracking behaviour and recognising activities being carried out by two people in a computer vision laboratory (Nguyen et al., 2002). In this experiment the authors were able to successfully distinguish between a person interacting with an object and only passing by the object, predicting the subjects' intentions only through observations of their location.

2.4 Association Rule Mining

The class of pattern mining most relevant to this thesis is the classic data mining problem of association rule mining, or "market basket" analysis, first introduced by Agrawal et al. (1993). Association mining is concerned with the discovery of sets of items that frequently occur together within the records of a transactional database. Association rule mining is, essentially, a counting exercise; the problem being tackled is how to efficiently find, possibly large, subsets of frequently occurring associations within a combinatorial search space.

The problem can be defined as follows. Let there be a database $DB = \langle T_1 T_2 \ldots T_N \rangle$ of transactions T_i $(1 \leq i \leq N)$ such that $T_i\,(x) \in I$ $\forall$ items x in T_i where I is the set of all items $I = \left\{a^1 a^2 \ldots a^i \ldots a^M\right\}$ within DB. We wish to find associations, or itemsets, among the transactions T_i in DB that are implication rules of the form $X \Rightarrow Y$ with the properties $X \subseteq I$, $Y \subseteq I$ and $X \cap Y = \emptyset$.

A rule r is considered to be of interest, or significant, when its support and confidence measures meet some arbitrary thresholds. The support and confidence measures are calculated as $\frac{|T_{xy}|}{N}$ and $\frac{|T_{xy}|}{|T_x|}$ respectively where $|T_{xy}|$ is the number of transactions in DB containing the items $X \cup Y$, $|T_x|$ is the number of transactions in DB containing all items X and N is the number of transactions in DB. The support of a rule is a simple measure of the rule frequency; the most frequent rules being likely to reflect common knowledge about a domain while rules with lower support may highlight insights that are little known and may even be unexpected. The confidence measure, in contrast, is a statistical measure of the accuracy of the rule's implication. That is, how confident the implication that a transaction containing the items in X will also contain the items in Y.

The definition of significance used to mine a database is dependent on the application domain; alternative measures of interestingness may be chosen in place of, or in conjunction with, the traditional support and confidence thresholds (Tan et al., 2002). Constraints may also be placed on the mining such that only a subset of rules are returned dependent on inclusions or exclusions of certain items in the rule antecedent or consequent.

2.4.1 Apriori

The most well known method for the mining of association rules is the Apriori algorithm (Agrawal and Srikant, 1994)[1], an extension of the original algorithm of Agrawal et al. (1993) that exploits the support monotonicity property of association rules. The monotonicity property guarantees that the support of k-length itemsets will be equal to or less than the support of the $(k-1)$-length subsets of the rule; it is used in Apriori to restrict the search space at each pass k over the database to those itemsets that contain known frequent subsets from the previous pass.

Apriori is presented algorithmically as Algorithm 2.1 and works as follows. An initial pass is made over the database DB to collect the frequency counts of each item. Those items that meet a minimum support threshold α are placed into the set of frequent items $L_1 \in L$ where $L = \{L_1, L_2, \ldots, L_k, \ldots, L_K\}$ is the set of frequent itemsets discovered at each pass k over DB; all other items being discarded. The stepwise phase of the algorithm now begins, the algorithm using the known frequent items in the set L_{k-1} from the previous pass to generate the candidate itemsets for the k^{th} pass over DB. Two itemsets $p \in L_{k-1}$ and $q \in L_{k-1}$ $(p \neq q)$ may be joined to produce a new candidate itemset when $p_1 = q_1, p_2 = q_2, \ldots, p_{k-2} = q_{k-2}$ and $p_{k-1} < q_{k-1}$, it being assumed that items within DB are presented to Apriori in their lexicographic order and that this order is maintained in all generated itemsets. Each newly generated candidate is tested to ensure that all its $(k-1)$-length subsets are known to be supported, else the candidate is discarded. The frequency of all remaining candidates is then tested by a traversal over the database, the entire procedure ending when no more candidate items are generated.

Agrawal and Srikant (1994) also introduced the AprioriTid algorithm, a modification that transposes the original database into a structure of lookup tables to improve the efficiency of candidate generation and itemset counting, and to avoid the need for multiple database passes to be made. As the size of the data structures may be larger than available memory at the earlier stages of mining due to a potentially large number of candidate itemsets, the authors proposed AprioriHybrid as a hybrid mining method

[1]See also Mannila et al. (1994) for similar work conducted independently.

Algorithm 2.1: Apriori

Input: Transaction Database DB, minimum support threshold α
Output: Complete set of frequent associations $L = \langle L_1, L_2, \ldots, L_k, \ldots, L_K \rangle$

```
Method:
    C_1 ← frequency of all single items in DB;
    L_1 ← {c ∈ C_1 | support(c) ≥ α};
    k = 2;
    While L_{k-1} ≠ ∅ do
        // Generate the candidate itemsets
        For i ← 1 to |L_{k-1}| - 1 do
            p ← L_{k-1}(i);
            For j ← i + 1 to |L_{k-1}| do
                q ← L_{k-1}(j);
                If p_x = q_x (1 ≤ x ≤ k - 2) and p_{k-1} < q_{k-1} then
                    C_k ← C_k ∪ (p ∪ q);
                end
            end
        end
        // Prune unsupported candidates
        For each c ∈ C_k do
            prune c from C_k unless s ∈ L_{k-1} ∀ (k - 1)-length subsets s of C_k;
        end
        // Find the frequency of the candidate itemsets
        For each transaction T_i ∈ DB do
            For each candidate c ∈ C_k do
                If c ∈ T_i then
                    c.count++;
                end
            end
        end
        L_k ← {c ∈ C_k | support(c) ≥ α};
        k++;
    end
```

that employs the Apriori algorithm for the earlier stages of mining before switching to AprioriTid once the number of candidate itemsets is low enough to fit into memory.

Improving the efficiency of Apriori and finding better suited alternatives to the algorithm has been the focus of much research. For example, Savasere et al. (1995) proposed partitioning to reduce the number of passes required to be made over a database. In this work, a database is divided into smaller partitions which are independently mined to find the frequent itemsets they contain. The localized frequent itemsets are then combined to generate a list of potentially frequent itemsets over the entire database. Their support is then verified to produce the final complete set of frequent associations. Toivonen (1996) introduced a method of improving the efficiency of candidate generation by sampling the records of a large database. This probabilistic approach greatly reduces the computational and I/O overhead of mining such databases by requiring that only a single full pass over the data to accurately measure and test the frequency of the candidate associations.

Hashing was used by Park et al. (1997) for counting and candidate generation. In this work the count of all possible $(k+1)$-length candidate itemsets was stored within the bucket of a hash table whilst the frequency of the k-length itemsets was being tested. The $(k+1)$-length candidates could then be pruned by considering the count of the bucket into which they were hashed prior to making a pass over the database. The advantage in their method was the early pruning of generated candidate itemsets, especially at the earlier stages of mining where Apriori is known to generate a large number of itemsets. Holt and Chung (2002) extended this principle by assigning each item within a database a hash table of transaction identifiers (TIDs). Each occurrence of an item in the database is noted by hashing the TID of the transaction in which it occurs and incrementing the count of its bucket. These hash tables can then be employed to restrict the search space of the mining algorithm.

The algorithms mentioned here are but a cursory examination of this area of association rule mining; interested readers are invited to refer to Goethals (2003) for a more detailed review of these and related algorithms.

2.4.2 Frequent Pattern Growth

The Apriori and Apriori-inspired algorithms in Section 2.4.1 are breadth-first algorithms that use stepwise mining to find the complete set of frequent associations of length k prior to the mining of length $k+1$ associations. In contrast, the FP-Growth

Algorithm 2.2: FP-Tree Construction

Input: Transaction Database DB, support threshold α
Output: FP-Tree *root*

Method:

```
// First Pass
```

Count the frequency of single items in DB to build the set of frequent items F such that support $(F_j) \geq \alpha \ \ (1 \leq j \leq |F|)$;

```
// Second Pass
```

Create the root node *root*;
For each transaction $T_i \in DB$ **do**
  $A \leftarrow$ transaction items such that $A_j \in T_i, A_j \in F \ \ (1 \leq j \leq |A|)$ ordered by descending frequency;
  recursively insert the nodes A onto *root*;
end
return *root*

algorithm proposed by Han et al. (2000, 2004) employs depth-first search to mine Frequent Pattern Trees (FP-Trees), a tree structure representation of transactional data sets. Each node within the tree carries a codebook entry mapping to an item descriptor, a frequency count, links to both its parent node and any children nodes, and a link to the next node of the same codebook ID in the tree. A header table of descending frequency ordered items points to the first occurrence of each item node in the tree. Transaction items are placed into the tree in descending frequency order to produce a compressed representation of the original database transactions. The frequent items in an arbitrary transaction can be retrieved by traversing the tree from the root down to a leaf node.

Two passes over a database are required to construct the tree, the method being given in Algorithm 2.2. The first pass returns the frequency counts of all items found in DB. The frequent items are sorted by descending frequency and are used to create the tree header table. Items that do not meet the minimum support threshold α are discarded. A second pass over DB sees the items in each transaction being sorted into descending frequency and recursively inserted into the tree. An example FP-Tree, constructed from the database in Table 2.1, is provided in Figure 2.6.

The resulting tree structure is mined using the FP-Growth algorithm, a recursive divide and conquer approach that avoids candidate generation. The algorithm used, presented as Algorithm 2.3, works as follows. At each recursive step, FP-Growth will build a conditional FP-Tree T_c for each item a_i in the item header table given the conditional prefix paths of a_i in the current tree T. The conditional prefix paths are those nodes in T that must be traversed from the root to reach each instance of a_i in the tree. Having found the conditional prefix paths, the newly created tree T_c is populated using the

Table 2.1: An example database showing the raw and frequency ordered items in each transaction.

Trans. ID	Raw Items	Ordered Items
100	A C E	A C E
200	A B C	A C B
300	B D F	D B F
400	A C D	A C D
500	A C D F	A C D F

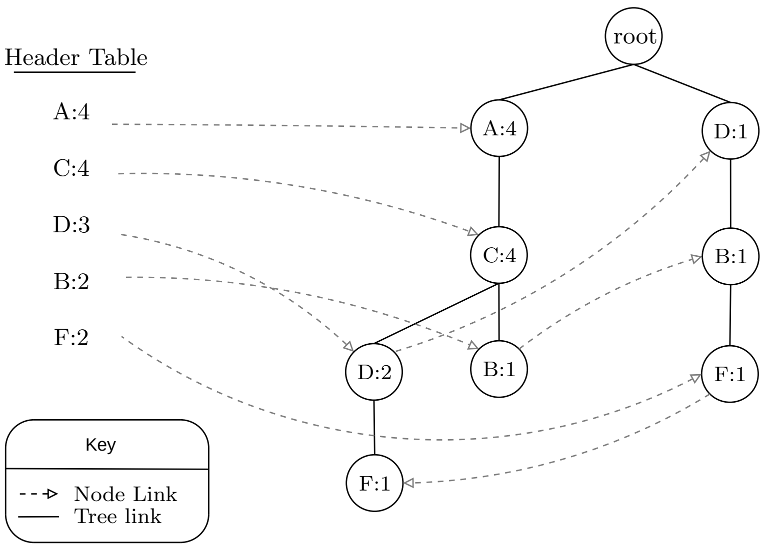


Figure 2.6: The FP-Tree for the example database in Table 2.1 with minimum support $\alpha = 2$. Colon delineated numbers depict the node frequency count.

Algorithm 2.3: FP-Growth

Input: FP-Tree T, mining support threshold α
Output: Set of frequent rules
Method:
  minedRules $\leftarrow \emptyset$;
  For each item a_i in header table of T from least to most frequent such that support $(a_i) \geq \alpha$ **do**
    Find the conditional prefix path for a_i and build the conditional FP-Tree T_c;
    If T_c contains a single path P **then**
      $T_c \leftarrow T_c$ with P removed;
      singlePathRules $\leftarrow$ all combinations of intratransaction nodes in P;
    end
    returnedRules $\leftarrow$ call FP-Growth (T_c, α);
    ruleSet $\leftarrow$ *returnedRules* $\cup$ *singlePathRules*;
    For each rule $R \in$ *ruleSet* **do**
      add a_i to R with support $(R) = \min(\text{support}(R), \text{support}(a_i))$;
    end
    add a_i to *ruleSet* with support = frequency of a_i in T;
    minedRules $\leftarrow$ *minedRules* $\cup$ *ruleSet*;
  end
  Return *minedRules*

tree construction method in Algorithm 2.2. For example, the prefix nodes of item B in the FP-Tree in Figure 2.6 are $\langle$A C:1$\rangle$ and $\langle$D:1$\rangle$ and produce the conditional tree $T_c|$B seen in Figure 2.7, the colon notation indicating the frequency count of each prefix path. Itemsets are generated by taking the dot product of the current item a_i and any rules returned by a recursive call to FP-Growth on the conditional tree $T_c|a_i$. Recursion ends when no more frequent items exist in the header table of T_c.

FP-Growth has been shown to be computationally more efficient than the Apriori and TreeProjection (Agarwal et al., 2001) algorithms, especially on dense data sets where the sharing of item nodes allows for a highly compact representation of the data. The bushy nature of the tree when applied to sparse data sets, however, offers opportunity for improvement. This was demonstrated by Pei et al. (2001) in their proposed H-struct hyper-structure, a two dimensional array of interlinked items, and H-mine, an algorithm for mining the structure. The H-mine algorithm uses the H-struct for mining until dense data is encountered; at this stage it will dynamically switch to FP-Growth.

Liu et al. (2003, 2004) investigated the performance of the Ascending Frequency Ordered Prefix-Tree (AFOPT), a variation of the FP-Tree in which items are ordered by their ascending frequency. It was shown that reversing the item ordering allowed the cost of tree traversal and conditional subtree construction to be greatly reduced; the order of nodes within the AFOPT ensuring that the next conditional subtree to be mined is prepared with minimal overhead.

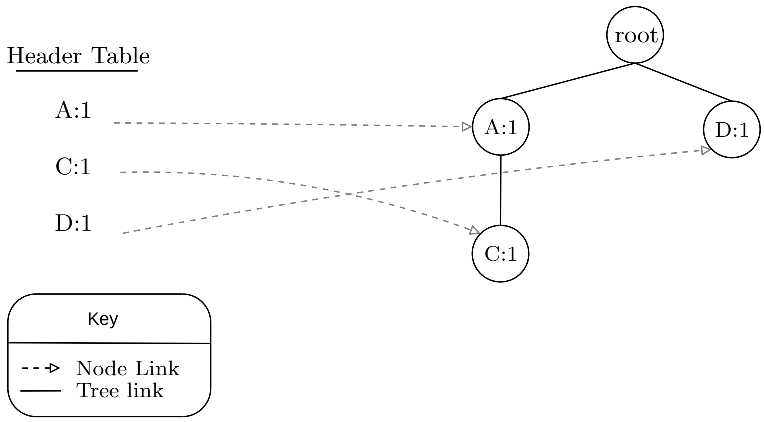


Figure 2.7: The conditional FP-Tree $T_c|\text{B}$. Colon delineated numbers depict the node frequency count.

Other notable works relevant to this area is Eclat by Zaki et al. (1997) and Zaki (2000), the first depth-first association rule mining algorithm, and OpportuneProject by Liu et al. (2002), a pattern growth algorithm that dynamically selects between tree and array representations of the transactional data to optimise mining efficiency.

2.5 Intertransaction Association Rules

*Inter*transaction Association Rule (IAR) mining (Lu et al., 1998, 2000) extends the discovery of association rules discussed in Section 2.4 to include relationships that span transactions in one or more domain specific dimensions. The dimensional attribute may be, for example, temporal in the prediction of stock market movements or spatial in a GIS application.

As for the *intra*transaction association rule case, the set of all items $I = \left\{a^1 a^2 \ldots a^i \ldots a^M\right\}$ are said to occur in a database $DB = \langle T_1 T_2 \ldots T_N \rangle$ of transactions T_i $(1 \leq i \leq N)$ such that $T_i(x) \in I$ $\forall$ items x in T_i. At any transaction T_i the items form the set $S_{T_i} = \left\{a^i_{T_i} \ldots a^k_{T_i}\right\}$. For the case of a single intertransaction dimension attribute, an intertransaction sliding window of size w transactions is passed over the transactions in DB to extract the extended transaction items such that the extended transaction at T_i is $E_{T_i} = \{S_{T_i}, S_{T_i+1} \ldots S_{T_i+w}\}$ and the set of all possible extended transaction items is $E = \left\{a^1_0 a^2_0 \ldots a^i_d \ldots a^M_w\right\}$. The term *intraitems* is used to refer to the set of items $\left\{a^1_0 \ldots a^M_0\right\}$. The superscript notation is dropped when the value of an item is known. IAR mining seeks to find rules of the form $X \Rightarrow Y$ with the properties:

$$X \subseteq E, Y \subseteq E \tag{2.38}$$

$$\exists a_i^0 \in X \tag{2.39}$$

$$\exists a_i^d \in Y, d > 0 \tag{2.40}$$

$$X \cap Y = \emptyset \tag{2.41}$$

The support measure for an association rule r continues to be calculated as $\frac{|T_{xy}|}{N}$ where $|T_{xy}|$ is the number of extended transactions containing all items in $X \cup Y$ and N is the number of transactions in DB. The confidence of a rule continues to be calculated as $\frac{|T_{xy}|}{|T_x|}$ where $|T_x|$ is the number of extended transactions containing all items in X.

Table 2.2 depicts the example database from Table 2.1 with an introduced temporal dimension. The extended transactions found when a sliding intertransaction window of size $w = 4$ is applied are shown.

IAR mining is similar to frequent episode and sequential pattern mining, two different types of intertransaction mining. Frequent episode mining (Mannila et al., 1995; Mannila and Toivonen, 1996; Mannila et al., 1997) seeks to find frequent partial ordering of items within a sliding window but does not consider the time interval relationships among those items. Sequential patterns (Agrawal and Srikant, 1995; Srikant and Agrawal, 1996) is concerned with finding frequent sequences where the precise ordering of items or events is important yet the temporal relationships are not considered. These two areas contrast to IAR mining where, assuming that the intertransaction dimensional attribute is temporal, the associative relationship among items is considered between time intervals but the ordering of items within the intervals is

Table 2.2: The example database from Table 2.1 with introduced transaction times. The extended transaction items shown are gathered when the transaction time is used as an intertransaction dimensional offset and a sliding intertransaction window of size $w = 4$ is passed over the database.

Trans. ID	Time	Items	Extended Items
100	1	A C E	A_0 C_0 E_0 A_2 B_2 C_2 B_3 D_3 F_3 A_4 C_4 D_4
200	3	A B C	A_0 B_0 C_0 B_1 D_1 F_1 A_2 C_2 D_2 A_4 C_4 D_4 F_4
300	4	B D F	B_0 D_0 F_0 A_1 C_1 D_1 A_3 C_3 D_3 F_3
400	5	A C D	A_0 C_0 D_0 A_2 C_2 D_2 F_2
500	7	A C D F	A_0 C_0 D_0 F_0

unimportant. IAR mining was selected for this work due to the interleaved nature of human behaviour as it allows us to capture the non-sequential relationships between observed activities while retaining some of the temporal aspect of such relationships. Opportunity exists for sequential pattern mining to be applied when the precise ordering of activities is desired and when an adequate supply of training data is available.

2.5.1 E-Apriori

Intertransaction association mining was first proposed in Lu et al. (1998, 2000) with the Apriori inspired E-Apriori and EH-Apriori algorithms. E-Apriori, like Apriori, is a stepwise algorithm that makes numerous passes over a data set to test, in each pass, the frequency of the k-length candidate itemsets generated from frequent $(k-1)$-length associations discovered in the previous pass.

The E-Apriori algorithm is detailed in Algorithm 2.4. The count of all items in the extended item set E is gathered in an initial pass over the data set and items not meeting the minimum support threshold α are discarded. The set of all possible 2-length itemsets is then generated and their support verified with a second pass over the database. The generation-then-test phase of the algorithm nows changes, all future passes over the data set making use of hash-trees in the generation and counting of itemsets. The search space for candidate generation is reduced in E-Apriori by grouping previously discovered associations into bins based on the number of intraitems contained in the itemsets. E-Apriori assumes that items remain sorted both by their lexicographic order and by ascending intertransaction offset. Two itemsets $p \in L_{k-1}$ and $q \in L_{k-1}$ may be joined to create a new itemset when $p_x = q_x\,(1 \leq x \leq k-2)$ and $p_{k-1} < q_{k-1}$, the comparison of two items $a_d^i < a_t^j$ being true iff $(d < t) \vee \big((d = t) \wedge \big(a^i < a^j\big)\big)$.

The EH-Apriori algorithm introduced alongside E-Apriori makes additional use of hashing to reduce the overhead of counting itemsets on the second pass.

2.5.2 First Intra Then Inter

The First Intra Then Inter (FITI) algorithm was introduced by Tung et al. (1999, 2003) shortly after the work of Lu et al. (1998). FITI is an improved candidate generation algorithm that first finds the complete set of intratransaction associations in order to transform the data set into lookup structures that aid the subsequent mining of intertransaction itemsets. The advantage of FITI lies in its ability to discard unnecessary data early on in the mining process and to use the discovered intratransactions to more

Algorithm 2.4: E-Apriori

Input: Extended transaction Database DB, minimum support threshold α, sliding intertransaction window size w

Output: Complete set of frequent intertransaction associations $L = \{L_1, L_2, \ldots, L_k, \ldots, L_K\}$

Method:

```
// Initial pass over DB
C_1 ← frequency of all single extended items in DB;
L_1 ← {c ∈ C_1 | support(c) ≥ α};
// k = 2
C_2 ← {a_0^1, a_x^2} | (a_0^1 ∈ L_1) ∧ (a_x^2 ∈ L_1) ∧ ((x ≠ 0) ∨ (x = 0 ∧ a_0^1 < a_0^2));
For each extended transaction T_i ∈ DB do
    For each candidate c ∈ C_2 do
        If c ∈ T_i then
            c.count++;
        end
    end
end
L_2 ← {c ∈ C_2 | support(c) ≥ α};
// All k ≥ 3 passes
k = 3;
While L_{k-1} ≠ ∅ do
    // Generate the candidate itemsets
    C_k ← ∅;
    For n ← 1 to k − 1 do
        G_n ← itemsets in L_{k-1} with n intraitems;
        For i ← 1 to |G_n| − 1 do
            p ← G_n(i);
            For j ← i + 1 to |G_n| do
                q ← G_n(j);
                If p_x = q_x (1 ≤ x ≤ k − 2) and
                [(p_{k-1}.offset < q_{k-1}.offset) or ((p_{k-1}.offset = q_{k-1}.offset) and (p_{k-1}.item < q_{k-1}.item))]
                then
                    C_k ← C_k ∪ (p ∪ q);
                end
            end
        end
    end
    // Prune unsupported candidates
    For each c ∈ C_k do
        prune c from C_k unless s ∈ L_{k-1} ∀ (k − 1)-length subsets s of C_k;
    end
    // Find the frequency of the candidate itemsets
    For n ← 1 to k do
        G_n ← itemsets in C_k with n intraitems;
        For each extended transaction T_i ∈ DB do
            // Query hash-tree
            G'_n ← {c ∈ G_n | intraitems(c) ∈ T_i};
            For each c ∈ G'_n do
                If c ∈ T_i then
                    c.count++;
                end
            end
        end
    end
    L_k ← {c ∈ C_k | support(c) ≥ α};
    k++;
end
```

efficiently guide the generation of intertransaction candidates.

FITI is algorithmically detailed as Algorithm 2.5 and works as follows. First, the complete set of intratransaction associations are mined using any available intratransaction itemset mining algorithm. The frequent itemsets are then used to construct the lookup links of the Frequent Itemset Linked Table (FILT) used in FITI. The lookup link structure, depicted in Figure 2.8(a), is a lookup table of unique itemset IDs that point to nodes representing discovered intratransaction itemsets. Generator and extension links between the itemset nodes are constructed next; the generator links mapping the relationships between $(k-1)$-length itemsets and the k-length itemsets that result from their joining; the extension links showing which two $(k-1)$-length itemsets can be joined to generate any given k-length itemset. Example generator and extension links for the sample link lookup structure in Figure 2.8(a) are shown in Figure 2.8(b). Two more sets of links showing itemset subset and descendant relationships are stored in the FILT. The subset link structure, depicted in Figure 2.8(c), links k-length itemset nodes to their $(k-1)$-length subsets while descendant links such at those in Figure 2.8(d) show which $(k+1)$-length itemsets nodes contain a given k-length itemset as a prefix.

Next, the FILT is used to transform the database into Frequent Itemset Tables (FITs) to aid mining. Itemset lookup link IDs are used to encode the frequent intratransaction itemsets in the original database into a set of FITs $F = \{F_1, F_2, \ldots F_k, \ldots F_{maxK}\}$ such that a table F_k stores the occurrence of all k-length intratransaction itemsets in DB and the number of required tables $maxK$ is the maximum length of the discovered intratransaction itemsets. Table 2.3 demonstrates how the database in Table 2.2 can be encoded as three FITs using the lookup link structure in Figure 2.8(a) when a minimum support threshold of $\alpha = 3$ is applied.

Intertransaction itemsets are encoded as a sequence of intratransaction sub-windows $I = [I_0, I_1, \ldots I_p, \ldots I_w]$ such that each sub-window I_p contains a single intratransaction itemset ID from the FILT lookup links or $I_p = 0$. The encoded itemset $I = [1, 0, 4, 0, 7]$, for example, maps to the itemset A_0 A_2 C_2 A_4 C_4 D_4 using the lookup links in Figure 2.8(a).

The stepwise mining phase of FITI now begins with the generation of all possible 2-length intertransaction itemset candidates, the frequency of which are stored in a hash table to reduce memory usage. For all $(k \geq 3)$-length intertransaction itemsets, FITI adopts the familiar generate-prune-count process used in algorithms such as E-Apriori. Known frequent itemsets of length $k-1$ are inserted into a hash table during the candidate generation procedure such that two itemsets I and J are only tested for a

Algorithm 2.5: FITI

Input: Extended transaction Database DB, minimum support threshold α, sliding intertransaction window size w, set of frequent intratransaction itemsets L'

Output: Complete set of frequent intertransaction associations $L = \{L_1, L_2, \ldots, L_k, \ldots, L_K\}$

Method:

```
L_1 ← L'_1;
F' ← generate the FILT structure from L';
F ← database transformation using F';
// Itemsets are stored in FIT representation from here on
// Generate and count candidates for k = 2.  Counts for items in C_2 are stored
   in a hash table
C_2 ← {a^1_0, a^2_x} | (a^1_0 ∈ L'_1) ∧ (a^2_x ∈ L'_1) ∧ ((x ≠ 0) ∨ (x = 0 ∧ a^1_0 < a^2_0));
L_2 ← {c ∈ C_2| support (c) ≥ α};
// Levelwise mining for k ≥ 3
k = 3;
While L_{k−1} ≠ ∅ do
    // Candidate itemset generation
    C_k ← call FITI-Gen(F', L_{k−1}, k, w) in Algorithm 2.6;
    // Prune unsupported candidates
    For each c ∈ C_k do
        prune c from C_k unless s ∈ L_{k−1} ∀ (k − 1)-length subsets s of C_k;
    end
    // Count
    T ← create counting tree;
    For each c ∈ C_k do
        insert c into T;
    end
    For each transaction f ∈ F do
        increment count of itemsets in T;
    end
    L_k ← {c ∈ C_k| support (c) ≥ α};
end
```

Algorithm 2.6: FITI-Gen

Input: FILT structure F', set of frequent candidates L_{k-1}, step k, intertransaction sliding window size w

Output: Set of candidates C_k

Method:

$C_k \leftarrow \emptyset$;
$H \leftarrow$ create new hash table;
For each $I \in L_{k-1}$ **do**
  For $d \in 0$ to w **do**
    If $I(d) \neq 0$ **then**
      $I' \leftarrow I$;
      $I'(d) \leftarrow 0$;
      $indx \leftarrow$ hash of I';
      If $H(indx) \neq \emptyset$ **then**
        For each $J \in H(indx)$ **do**
          `// Intra-window join`
          If $(\exists p, I_p$ and J_p are generators of a $K_p) \wedge (\forall q, I_q = J_q\,(q \neq p)) \wedge$
          $[((\text{length}(I_p) > 1) \wedge (\text{length}(J_p) > 1) \wedge (\forall r, I_r = 0 \wedge J_r = 0\,(r > p))) \vee$
          $((\text{length}(I_p) = 1) \wedge (\text{length}(J_p) = 1) \wedge (\forall r, \text{length}(I_r) \leq 1 \wedge \text{length}(J_r) \leq 1\,(r > p)))]$
          then
            $C_k \leftarrow C_k \cup (I_1, \ldots I_{p-1}, K_p, I_{p+1}, \ldots I_k)$;
          end
          `// Cross-window join`
          If $k > w$ **then**
            If $(\exists p, I_p \neq 0 \wedge J_p = 0) \wedge (\exists q, I_q = 0 \wedge J_q \neq 0\,(p \neq q)) \wedge$
            $(\forall r, I_r = J_r\,(r \neq p, r \neq q)) \wedge (\forall r, I_r = 0\,(r > p)) \wedge$
            $(\forall r, J_r = 0\,(r > q)) \wedge (\forall r, \text{length}(I_r) \leq 1 \wedge \text{length}(J_r) \leq 1)$ **then**
              $C_k \leftarrow C_k \cup (I_1, \ldots I_p, \ldots, J_p, \ldots I_k)$;
            end
          end
        end
      end
      $H(indx) \leftarrow H(indx) \cup I$;
    end
  end
end

Table 2.3: FIT representation of the example database in Table 2.2 created with a minimum support threshold of $\alpha = 3$ and encoded using the lookup link structure in Figure 2.8(a).

Trans. ID	Time	Items	Items $\vert\alpha$	F_1	F_2	F_3
100	1	A C E	A C	1, 2	4	
200	3	A B C	A C	1, 2	4	
300	4	B D F	D	3		
400	5	A C D	A C D	1, 2, 3	4, 5, 6	7
500	7	A C D F	A C D	1, 2, 3	4, 5, 6	7

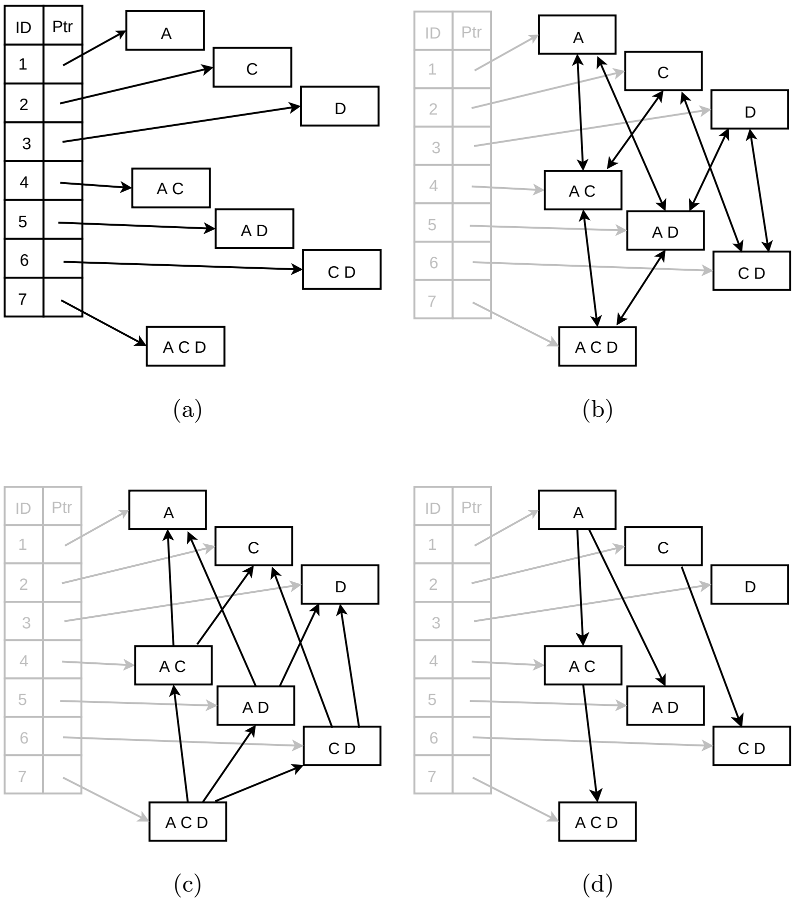


Figure 2.8: The FILT data structure of the (a) item links, (b) generator and extension links (c) subset links and (d) descendant links used in FITI.

possible join if they are assigned to the same bin. Two types of itemset joins may be made in the generation of a new candidate itemset. The first, an intra-window join between two itemsets I and J is possible and will produce a new candidate of the form $\{I_1, \ldots I_{p-1}, K_p, I_{p+1}, \ldots I_k\}$ when the following conditions are true:

$$\exists p, I_p \text{ and } J_p \text{ are generators of } K_p \tag{2.42}$$

$$\forall q, I_q = J_q \, (q \neq p) \tag{2.43}$$

$$\begin{gathered}[(\text{length}\,(I_p) > 1) \wedge (\text{length}\,(J_p) > 1) \wedge (\forall r, I_r = 0 \wedge J_r = 0\,(r > p))] \vee \\ [(\text{length}\,(I_p) = 1) \wedge (\text{length}\,(J_p) = 1) \wedge (\forall r, \text{length}\,(I_r) \leq 1 \wedge \text{length}\,(J_r) \leq 1\,(r > p))]\,.\end{gathered} \tag{2.44}$$

A cross-window join between itemsets I and J is possible and will produce a new candidate of the form $[I_1, \ldots I_p, \ldots, J_p, \ldots I_k]$ when the following conditions are true:

$$\exists p, I_p \neq 0 \wedge J_p = 0 \tag{2.45}$$

$$\exists q, I_q = 0 \wedge J_q \neq 0\,(p \neq q) \tag{2.46}$$

$$\forall r, I_r = J_r\,(r \neq p, r \neq q) \tag{2.47}$$

$$\forall r, I_r = 0\,(r > p) \tag{2.48}$$

$$\forall r, J_r = 0\,(r > q) \tag{2.49}$$

$$\forall r, \text{length}\,(I_r) \leq 1 \wedge \text{length}\,(J_r) \leq 1. \tag{2.50}$$

The frequency of the candidate itemsets is counted using a tree of depth w, each level l representing the encoded itemsets of the candidate itemsets in the l^{th} sub-window. The count of itemsets within the tree are incremented whenever a leaf node is reached via a traversal down from the root using the itemsets present in the FITs. Itemsets found to be unsupported are pruned from the tree.

Prior to counting FITI will query the tree of frequent $(k-1)$-length itemsets from the previous pass on the frequency of the $(k-1)$-length subsets of each candidate itemset. Candidates found to contain unsupported subsets will be discarded.

2.5.3 Frequent Pattern Growth

The FP-Tree has previously been applied to a constrained case of intertransaction association mining in Berberidis et al. (2004) for the prediction of rare events. Here, a transactional database containing occurrences of specific rare events was mined to find frequent intertransaction associations that could be used to predict and avoid these events in the future. Items in the database were encoded using a codebook whose alphabet mapped to all possible combinations of events and intertransaction offsets in the database. The intertransaction relationships were restored using the codebook following mining. This technique requires an FP-Tree be conditioned on each rare event that is to be mined on and can not be efficiently applied to the general IAR mining problem.

2.6 Visual Association Rule Exploration

Visual data mining tools for association rule exploration aim to provide users with a convenient way by which to navigate through large sets of discovered associations and the relationships they entail. Ideally, such tools will allow users to both obtain summary information about the results being visualised and retain the ability to drill down to specific rules of interest. This section will focus on previous work in visual data mining concentrating on the problem of understanding association rules; interested readers may wish to refer to Ferreira de Oliveira and Levkowitz (2003) for a wider survey of research into the area of visual data mining.

Two dimensional matrices showing the one-to-one relationship of rules was one of the first techniques used to visualise associations. The antecedent and consequent of the rules were used to label the rows and the columns of the matrix while the support and confidence of the rules were visually represented through icons on the grid. Such displays are limited in the complexity of the associations they are able to represent and are prone to occlusion when visualised in three dimensions. Commercial tools such as MineSet (Brunk et al., 1997) and Quest (Agrawal et al., 1996) were the first to make use of this method.

A more versatile matrix representation that shows the relationships between items and itemsets was introduced in Wong et al. (1999). The matrix columns were used to represent associations with rows representing the individual items. Item membership in the rules was depicted by binary coloured bricks placed onto the grid, the brick colour distinguishing between the rules antecedent and consequent items. The grid was displayed in three dimensions with the support and confidence of the associations charted alongside the x-axis. The two dimensional matrix theme was again used in Ong et al. (2002). Here, associations of the same items were grouped into cells and placed onto the grid in order of descending support and descending confidence. Interesting changes in the rules were marked on the grid and on an accompanying hierarchical view of the associations.

The use of mosaic plots for association rule analysis was proposed by Hofmann et al. (2000). Mosaic plots are graphical representations of contingency tables that show the frequency of combinations of items in a data set. The frequency of associations is graphically depicted such that the relative size of each table cell, or tile, in the mosaic plot reflects the relative frequency of the items within the association. The use of double decker plots, a variation of mosaic plots that show the conditional probabilities of an association rule's consequent given its antecedent items, was also introduced. The appeal of double decker plots lies in their ability to visually highlight those combinations of antecedent items that provide the strongest indication that a particular rule's consequent will apply.

A richer display system was discussed in Blanchard et al. (2003). Here associations are sorted into subsets of similar rules and displayed in a virtual landscape that users navigate. The landscape shows a field that is overlooked at either end by blocks of grandstand seating. Avatar representations of the rules, constructed from cones upon which spheres are placed, are seated according to their similarity to the current rule subset, the dissimilar rules being delegated to "cheaper" seats at the rear. The height of an avatar's cone represents the confidence of the association it represents while the size of the sphere visually indicates its support. Selecting a rule moves the visualisation to another field, or rule subset. One block of seating is populated with specialisations of the currently selected rule while the other is populated with generalisations, giving users a means of drilling through the discovered rules.

Bruzzese and Davino (2003) proposed a visualisation technique using parallel coordinates on a two dimensional plane. Antecedent items of rules sharing a common consequent are evenly distributed on the horizontal axis and a measure of the rules' efficacy with or without the presence of each antecedent item is displayed on the vertical axis as a value in the range $[1, -1]$. Redundant and non-redundant itemsets alike are plotted, the utility measure of an item not present in a redundant rule being set to -1.

Parallel coordinates were recently used by Yang (2005) to visualise the many-to-many relationships of association rules. In this work, items are placed on a vertical axis which is then duplicated multiple times and evenly distributed along the horizontal axis. Rules and itemsets are plotted by drawing Bézier curves through the items on the vertical axes that make up a given rule, the maximum rule length defining the number of vertical axes that are required. To avoid clutter, only the non-redundant rules are displayed and item taxonomies can be defined for a generalised representation of the associations. A taxonomic node can be selected to expand or collapse it, resulting in an interactive hierarchical representation of the associations.

2.7 Review Summary

This chapter has reviewed work pertinent to this thesis from the graphical model and data mining literature. The Hidden Markov Model (HMM) and its Explicit State Duration HMM and Hierarchical HMM variants were introduced with a survey of related works on activity and gesture recognition. The classic data mining problem of association rule mining was then introduced with an examination of the Apriori and FP-Growth algorithms and their related works. The relatively modern problem of intertransaction association mining was then described along with a detailed overview of the two current intertransaction mining algorithms: E-Apriori and First Intra Then Inter. Finally, the chapter concluded with a description of several techniques concerned with the visual display and exploration of discovered association rules.

Chapter 3

Detecting Human Behaviour Abnormality via Stochastic Models

This chapter investigates the use of the Hidden Markov Model (HMM) and its Hierarchical HMM (HHMM) and Explicit State Duration HMM (ESD-HMM) variants for the recognition of abnormality in human behaviours observed through a visual tracking system. Stochastic models are of interest in such an application due to their ability to cope with noise produced both by the underlying tracking system and from variations in how a subject may carry out their daily activities. The use of such models is also interesting from a philosophical perspective; the hidden state layers of the models can be said to function as a representation of a person's decision making process, the probabilistic structure of the hidden states being inferred from actions observed as a result of the decision making process (Pentland and Liu, 1995). Although the context of behaviour recognition in this thesis is primarily concerned with identifying deviations from expected behaviour, the use of the models as classifiers of normal behaviour is not overlooked.

An examination into how the Hierarchical HMM (HHMM) may be applied to learning and recognising human activity sequences is given in Section 3.1. The focus of this first work is to investigate the HHMM's ability to capture and represent simple activity sequences that are representative of a person's expected behaviour. The HHMM was chosen because its structure is able to more naturally capture the hierarchy present in human activity than similar flat models whilst its statistical nature offers the ability to deal with noisy and missing data. Furthermore, the hierarchical nature of the HHMM allows for model reuse and faster learning through independent submodel generation (Theocharous, 2002).

The importance of incorporating duration in a model of human behaviour will be investigated in Section 3.2. Here, the Hidden Semi-Markov Model (HSMM), or Explicit State Duration HMM (ESD-HMM), is applied to the task of sequence classification and to the problem of detecting abnormality due to a subject spending either an unusually long or unusually short period of time at an otherwise normal activity. A modification to the standard ESD-HMM, that assumes that the activity sequence has been augmented with additional information from which the state transition times can be derived, will be examined in Section 3.3.

Finally, the ability of the Observed Time Indices (OTI) ESD-HMM to classify unseen sequences and to detect abnormality from video footage gathered from a real world smart home will be investigated in Section 3.4.

3.1 Hierarchical HMM

The hierarchical hidden Markov model (HHMM) (Fine et al., 1998) is a multi-level stochastic finite state machine that offers greater descriptive powers than similar flat models. The HHMM is recursive, hidden states are themselves HHMMs down to a final HMM (Rabiner, 1989) level that is responsible for the generation of observations. The recursive and autonomous nature of the HHMM allows for the construction of complex high level models that can describe the signal under observation in a way that is more natural than traditional models while affording flexible parameter estimation through the reuse of separately trained models. Rapid training of increasingly complex models can thus be accomplished, especially in multi-processor systems which are able to parallelize the training of lower level models for reuse.

In this section, the HHMM is applied to the task of modelling and inferencing on sequences of human activity as observed by a visual tracking system. Section 3.1.1 defines the experimentation methodology that is used in Section 3.1.2 to explore the application of the HHMM to the task of modelling human behaviours. Classification results demonstrating the application of the model to the recognition of previously unseen sequences of normal behaviour are presented in Section 3.1.3.

The original HHMM re-estimation formula in Section 2.3 was modified to accommodate multiple observation sequences. The set of N observation sequences is defined as $O = [O^1, O^2, \ldots, O^N]$ and a single observation O^n is referred to as o_t^n. As the sequence strings are independent, the likelihood of all observation sequences being produced by the model is

$$\Pr(O|\lambda) = \prod_{n=1}^{N} \Pr(O^n|\lambda). \tag{3.1}$$

The short hand notation $P_n = \Pr(O^n|\lambda)$ will be used to reduce the notation complexity.

The new vertical and horizontal transition probability estimates are defined as a normalised sum of the individually weighted observation sequences as follows:

$$\hat{\pi}^{q^1}(q_i^2) = \frac{\sum_{n=1}^{N} \frac{1}{P_n} \chi\left(O^n, 1, q_2^2, q^1\right)}{\sum_{n=1}^{N} \frac{1}{P_n} \sum_{i=1}^{|q_i^1|} \chi\left(O^n, 1, q_2^2, q^1\right)} \tag{3.2}$$

$$\hat{\pi}^{q^{d-1}}(q_i^d) = \frac{\sum_{n=1}^{N} \frac{1}{P_n} \sum_{t=1}^{T} \chi\left(O^n, t, q_i^d, q^{d-1}\right)}{\sum_{n=1}^{N} \frac{1}{P_n} \sum_{i=1}^{|q^{d-1}|} \sum_{t=1}^{T} \chi\left(O^n, t, q_i^d, q^{d-1}\right)} \tag{3.3}$$

$$\hat{a}_{ij}^{q^{d-1}} = \frac{\sum_{n=1}^{N} \frac{1}{P_n} \sum_{t=1}^{T} \xi\left(O^n, t, q_i^d, q^{d-1}\right)}{\sum_{n=1}^{N} \frac{1}{P_n} \sum_{t=1}^{T} \gamma_{\text{out}}\left(O^n, t, q_i^d, q^{d-1}\right)} \tag{3.4}$$

$$\hat{b}_{q_i^D}^{q^{D-1}}(v_k) = \frac{\displaystyle\sum_{n=1}^{N} \frac{1}{P_n} \left[\sum_{\substack{t=1 \\ \text{s.t. } o_t^n = v_k}}^{T} \chi\left(O^n, q_i^D, q^{D-1}\right) + \sum_{t>1, o_t^n = v_k} \gamma_{\text{in}}\left(O^n, t, q_i^D, q^{D-1}\right) \right]}{\displaystyle\sum_{n=1}^{N} \frac{1}{P_n} \left[\sum_{t=1}^{T} \chi\left(O^n, q_i^D, q^{D-1}\right) + \sum_{t=2}^{T} \gamma_{\text{in}}\left(O^n, t, q_i^D, q^{D-1}\right) \right]}. \tag{3.5}$$

3.1.1 Experimentation Methodology

A visual tracker was implemented to extract observation sequences from video recordings of a person performing various kitchen and a lounge room related activities in a laboratory environment. The tracker employs a Gaussian background model for foreground segmentation as described in Stauffer and Grimson (2000) and a Kalman filter to track objects across frames (Kalman, 1960). The position of the subject's feet in the scene, as approximated by taking the centre of the bottom of the tracker supplied bounding box, was used to calculate the proximity of the person to defined areas of interest in the room: the door, fridge, food preparation area, sink and the stove in the kitchen environment and the door, dinning table, television, bookcase and couch in the lounge room environment. Discrete observations mapping to these areas were recorded if the subject was in close proximity else an observation mapping to "undefined" was logged.

Four simplistic styles of dinner preparation and five typical lounge room activities were recorded at 25fps, each over a period of 60 to 70 seconds. The tracker extracted an observation sequence for each of the recordings by sampling the person's location every 25 frames.

The first cooking style, shown in Figure 3.1(a), involves spending some time preparing the food and rummaging through the fridge before the meal is cooked. The second cooking style, depicted in Figure 3.1(b), involves washing dishes prior to cooking on the stove. The third cooking style is shown in Figure 3.1(c). Here the person walks to the sink to wash the dishes, spends some time at the food preparation area and at the fridge before finally cooking the meal. The last cooking style in Figure 3.1(d) sees the subject transitioning between each area of interest in a round robin fashion, starting and ending at the stove, before leaving the room.

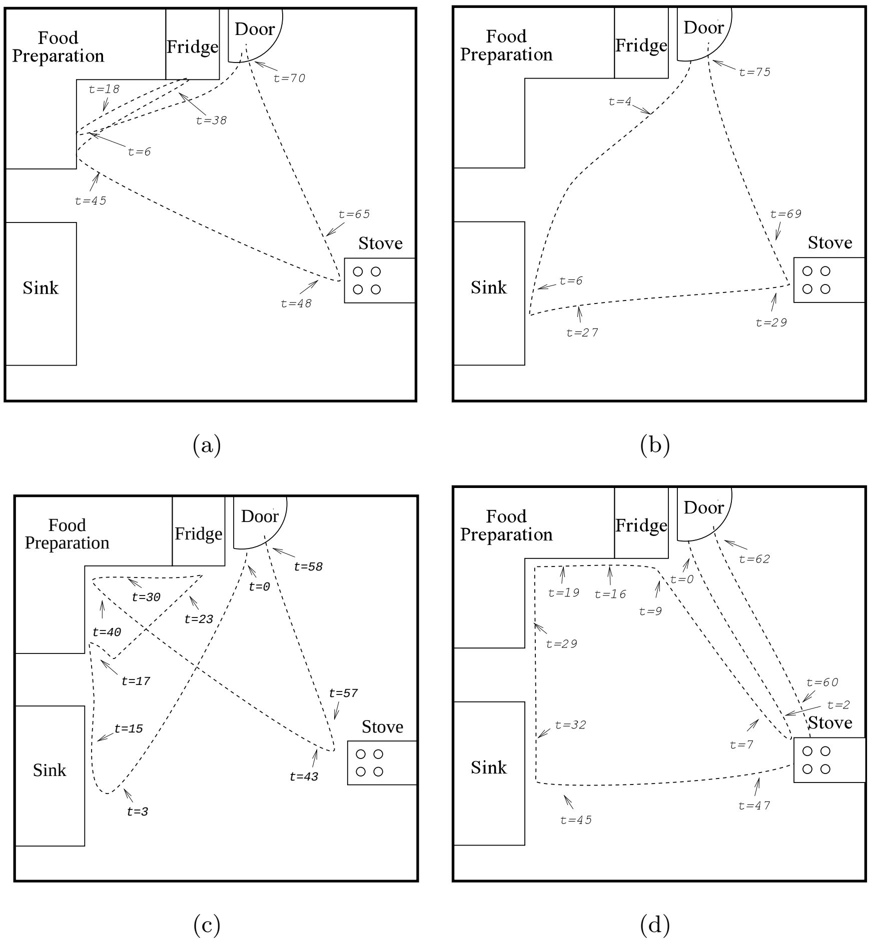


Figure 3.1: Layout of the kitchen showing the approximate track and elapsed time t in seconds for the (a) "food preparation first," (b) "washing dishes first," (c) "washing dishes and preparing the food" and the (d) "round robin" meal preparation sequences. Figure (a) and Figure (b) © 2003 IEEE. Reprinted, with permission, from Proceedings of the First IEEE International Conference on Pervasive Computing and Communications.

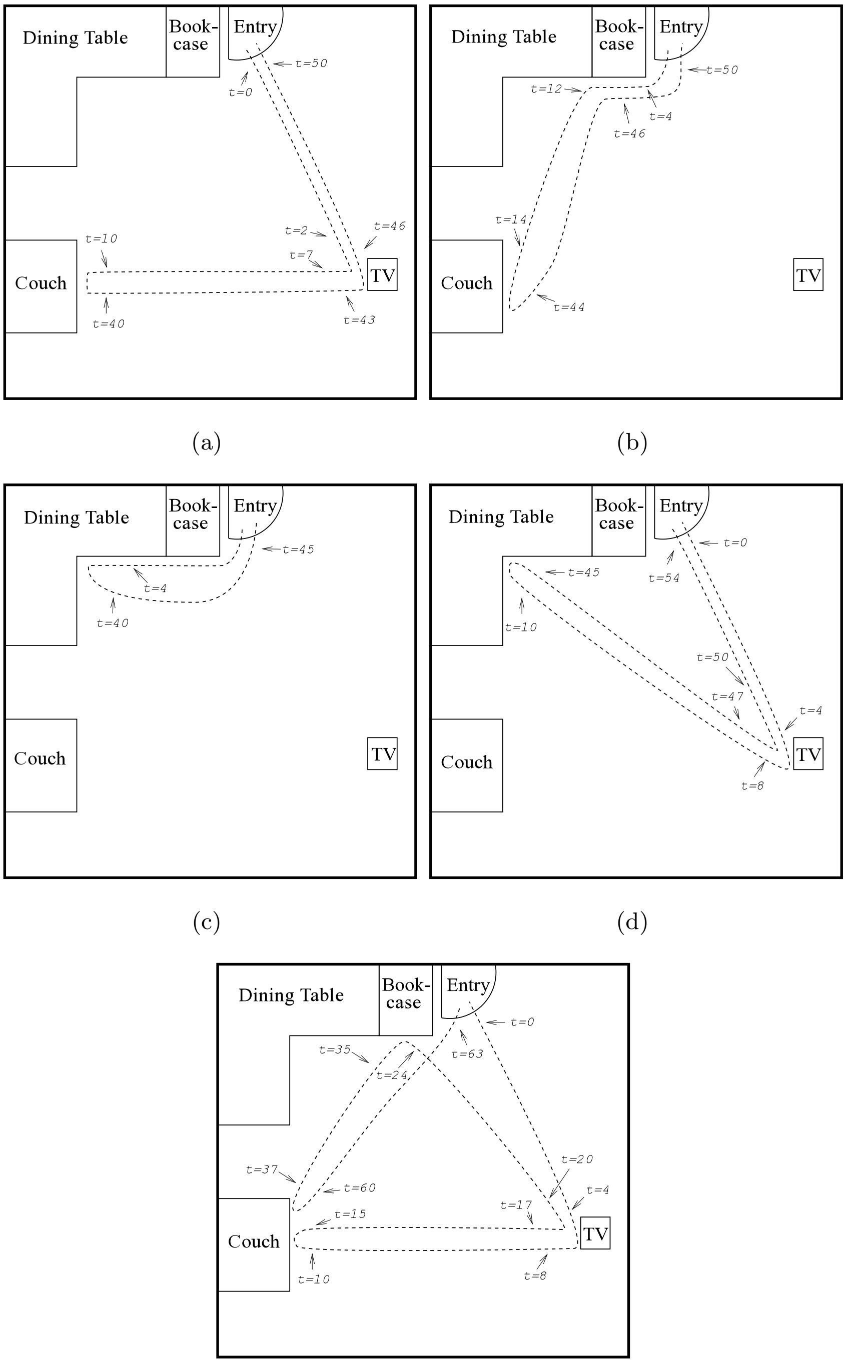


Figure 3.2: Layout of the lounge room showing the approximate track and elapsed time t in seconds for the (a) "watch television," (b) "read a book on the couch," (c) "eat dinner," (d) "eat dinner while watching television" and the (e) "there is nothing good on TV, read a book instead" lounge room activity sequences.

The lounge room sequences similarly feature transitions between areas of interest that aim to be representative of typical lounge room activities. The first, shown in Figure 3.2(a), sees the subject enter the room, turn on the television and sit on the couch for half a minute before leaving the room after visiting the television. Figure 3.2(b) depicts an activity of reading on the couch. Here the subject sits on the couch for half a minute after spending a few seconds selecting a book from the bookcase. The subject returns the book to the bookcase before exiting the room. In the third sequence, Figure 3.2(c), the subject enters the room and heads immediately towards the dinning room table. This sequence aims to mimic a basic eating activity with the subject leaving the room immediately afterward. A similar sequence is acted out in Figure 3.2(d); the subject eats dinner after first walking to the television set. Finally, Figure 3.2(e) depicts a hybrid lounge room sequence in which the subject turns on the television and sits down on the couch. Finding nothing of interest to watch, the subject returns to the television to switch it off before walking to the bookcase to select a book. The subject spends a moment reading on the couch before returning the book to the bookcase and vacating the room.

3.1.2 Modelling Human Behaviour

In this first experiment, a separate model for each of the cooking and lounge room activities is trained using a three layer topology. A total of four cooking and five lounge room models were built from the dataset of 126 training sequences, 14 per model, consisting of 56 cooking sequences and 70 lounge room sequences. The topology features all the production states at the lowest layer. Three internal states on the second level govern the vertical transitions to, and horizontal transitions among, the production nodes. Finally, the root node on the top level regulates the activation of, and horizontal transitions among, the second level internal states. The states are fully connected at all levels of the hierarchy.

The trained models are depicted in Figures 3.3–3.11. The thickness and darkness of the arrows and lines visually depict the relative transition likelihoods of the states such that darker and thicker lines indicate a higher probability. Insignificant state transition probabilities, typically those less than 2%, are not shown in order to improve clarity. The complete state transition and emission likelihoods of the trained models from the figures were derived are provided in Appendix A.

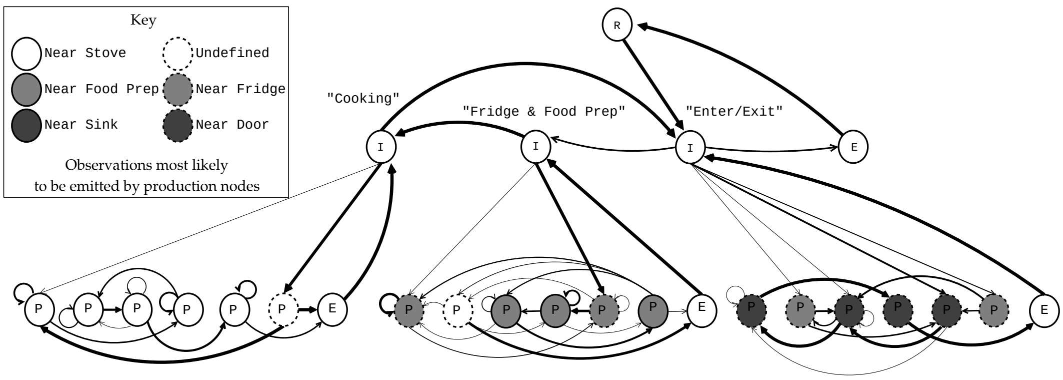


Figure 3.3: The trained model for the "food preparation first" kitchen style. Darker arrows indicate stronger state dependence and a higher transitional likelihood among the production (P), internal (I) and end (E) states. Insignificant transitions have been omitted for clarity.

Kitchen Activity

Examining the model in Figure 3.3, we notice the production states closely associate their calling parent states with clearly definable activity regions on the kitchen activity sequence maps. For example, the production nodes on the leftmost subtree only emit the `Near Stove` and `Undefined` labels. Given the context, we can associate the `Undefined` label with a person going to and from the stove area, as a person acting out the first style needs to first traverse the middle of the room and in doing so will not be in close proximity to the predefined areas. We hence label the internal parent state in Figure 3.3 as "`Cooking`."

The production state children of the second internal state primarily emit `Near Fridge` and `Near Food Prep` observations. The transition structure among the production nodes suggests a strong dependence between these two observations which closely corresponds to the "fridge then food preparation then fridge then food preparation" loop present in the training sequences. A single node responsible for the emission of `Undefined` observations exists to accommodate occasions when the tracker may momentarily lose track of the subject. This submodel has hence been labeled "`Fridge & Food Prep`." Analysis of the third internal state's children reveals that the internal node is responsible for generating only `Near Door` and `Near Fridge` observations. As this state is the only one vertically transitioned to by the root node and also the last internal node to be activated prior to termination of the entire model, we assign this internal state the label "`Enter/exit`."

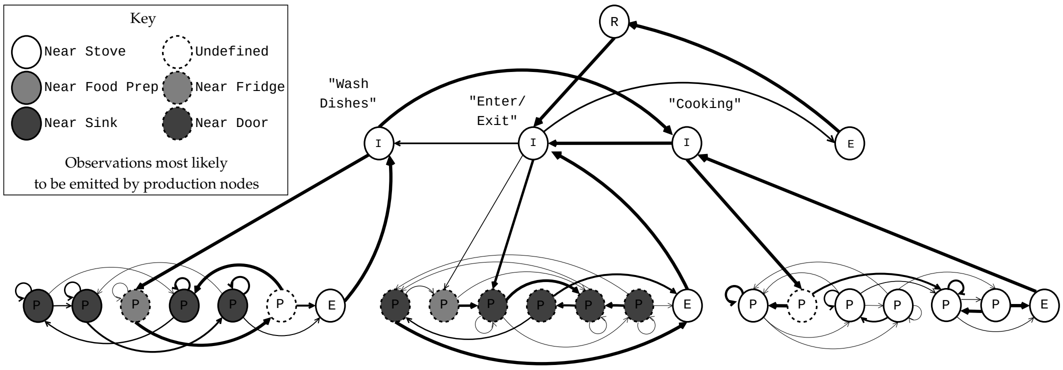


Figure 3.4: The trained model for the "washing dishes first" style. Darker arrows indicate stronger state dependence and a higher transitional likelihood among the production (P), internal (I) and end (E) states. Insignificant transitions have been omitted for clarity.

Tracing and labeling the state transitions governed by the root node we find that the entire dinner preparation sequence is abstracted as "`Enter/exit` → `Fridge & Food Prep` → `Cooking` → `Enter/Exit`" and corresponds almost completely with the atomic observations of the training data.

Similar results emerged through visual analysis of the second model in Figure 3.4. The three internal states on the second level were assigned the labels "`Wash Dishes`," "`Enter/exit`" and "`Cooking`" for the observations emitted by their respective production sub-states. Again, the transitional structure of the internal states strongly portrays the activity in the training data. The leftmost "`Wash Dishes`" submodel is seen to emit "`Near Fridge`" and "`Undefined`" observations as the subject passes in close proximity to the fridge on the way to the sink. "`Near Sink`" observations are then produced prior to the termination of this submodel. A transition to the "`Cooking`" internal state responsible for the production of "`Near Stove`" observations is then made. "`Undefined`" observations are also emitted here to accommodate the subject crossing the room in order to reach the stove. The entire model begins and ends via a transition to the second submodel labeled "`Enter/Exit`."

The state transition probabilities of the "washing dishes and preparing the food" model in Figure 3.5 also show a clearly defined structure. The leftmost internal state is the first state to be activated by a vertical transition down from the root node. The observation probabilities and the horizontal transition structure of this state's production node children suggest that they are responsible for emission of the "`Near Door`," "`Near Fridge`" and "`Undefined`" observations that are seen when the person enters the room

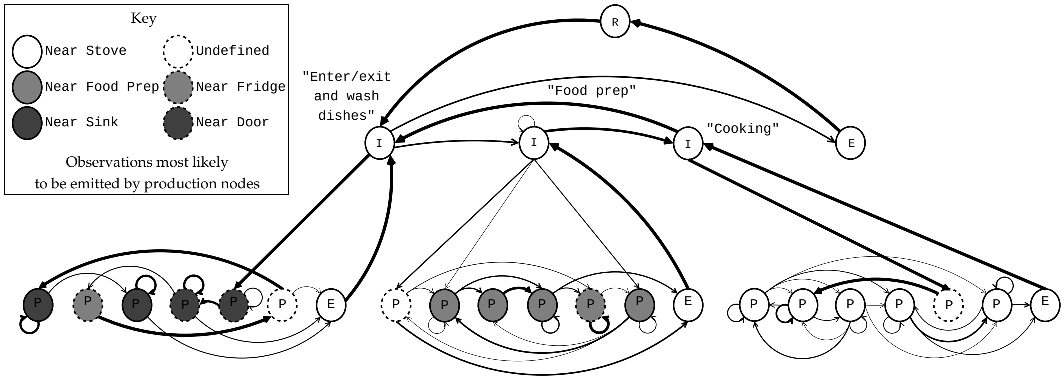


Figure 3.5: The trained model for the "washing dishes and preparing the food" kitchen style. Darker arrows indicate stronger state dependence and a higher transitional likelihood among the production (P), internal (I) and end (E) states. Insignificant transitions have been omitted for clarity.

and walks towards the kitchen sink. The "`Near Fridge`" observations are emitted as the subject is at times considered to be in close proximity to the fridge as they pass it by when entering the room. The production nodes are now likely to produce a series of "`Near Sink`" observations prior to making a horizontal transition to the end state. Alternatively, the production nodes may make an early transition to the end state following emission of the "`Undefined`" observations. The first internal state can hence be assigned the label "`Enter/exit and washes dishes`" given that this is the only internal state that is capable of terminating the execution of the model. The remaining two internal nodes and their production node children are seen to map to the "`Food prep`" and "`Cooking`" segments of the activity sequence.

The trained HHMM for the final kitchen style is shown in Figure 3.6. Here the model is seen to make a vertical transition down to the rightmost internal state. This state will, in turn, transition down to its production node children whose emission probabilities and transition structure maps to the "`Near Door`," "`Undefined`" and "`Near Stove`" observations that are seen when the subject enters and leaves the room via the stove. The third internal state can hence be labeled "`Enter/exit via stove`" given that this is the only internal state likely to make a horizontal transition to the final end state.

The second internal state is the next node that is likely to be visited by the model. This state has been labeled "`Food prep`" in Figure 3.6 as the observations produced by this state's production node children follow the "`Near Fridge`," "`Near Food Prep`" and "`Near Sink`" activity sequence as the subject sequentially moves between these areas in the room.

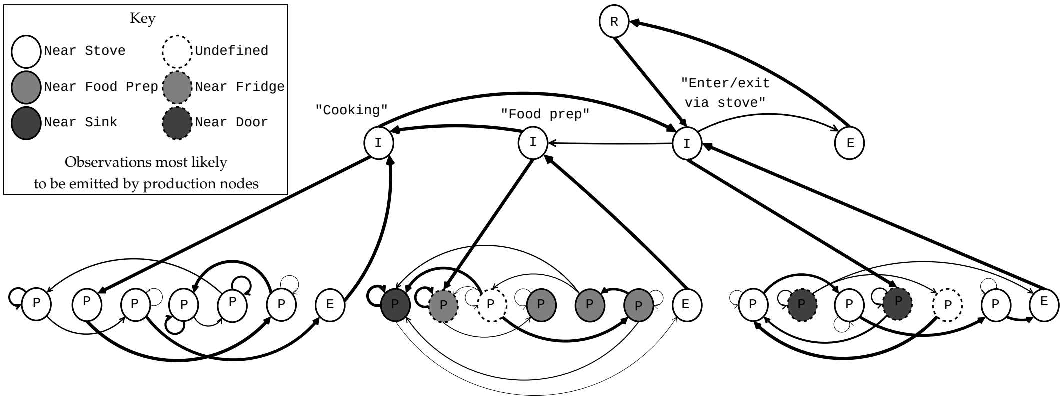


Figure 3.6: The trained model for the "round robin" kitchen style. Darker arrows indicate stronger state dependence and a higher transitional likelihood among the production (P), internal (I) and end (E) states. Insignificant transitions have been omitted for clarity.

The "`Near Stove`" observations mapping to the act of cooking in the sequence are emitted by the production node children of the leftmost internal state. Termination of this internal state will result in a horizontal transition back to the "`Enter/exit via stove`" internal state as the subject leaves the room.

Lounge Room Activity

The trained model for the "watch television" lounge style is shown in Figure 3.7. Here the second internal state has been assigned the label "`Enter`" given that it is the first internal state to be activated, its production node children are seen to be responsible solely for the generation of "`Near Door`" observations and the state is unable to make a horizontal transition to the end state. A horizontal transition will, instead, activate the leftmost internal state whose children nodes are seen to emit both "`Undefined`" and "`Near Door`" observations. These emission probabilities, combined with the transitional structure of the nodes, suggests that the leftmost internal state governs the production of sequence segments corresponding to both walking across and exiting the room. This node has therefore been labeled "`Walk across room/exit`" in the figure. The rightmost internal node can be said to correspond to "`Watch TV`." A horizontal transition from the "`Walk across room/exit`" to this node is seen to occur prior to a horizontal transition returning to "`Walk across room/exit`" and the termination of the model.

The structure of the trained model for the "read book on couch" activity sequence

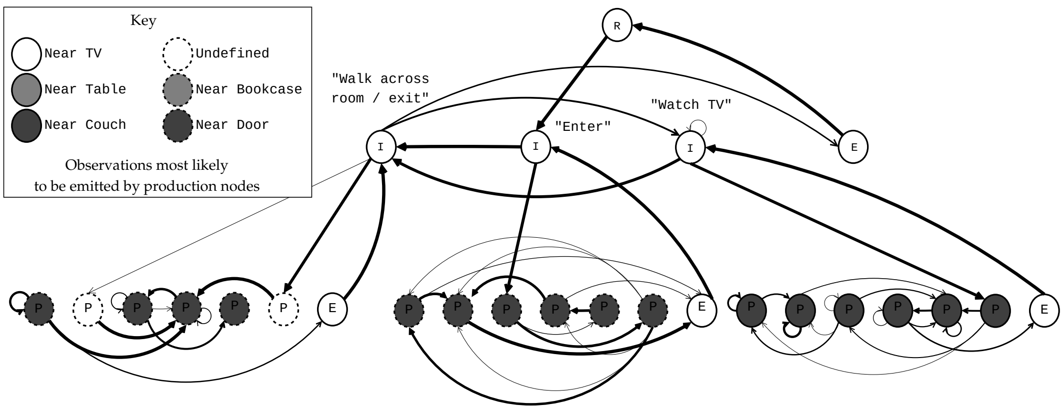


Figure 3.7: The trained model for the "watch television" lounge style. Darker arrows indicate stronger state dependence and a higher transitional likelihood among the production (P), internal (I) and end (E) states. Insignificant transitions have been omitted for clarity.

is shown in Figure 3.8. Well-defined state transitions exist among the internal states, offering only a single possible state activation path at this level. The first state activated by a vertical transition down from the root node appears to govern the production nodes responsible for observations corresponding to the subject entering the room and walking to the bookcase. This state has been labeled "`Enter via bookcase`".

A horizontal transition to the leftmost internal state is seen to result in a vertical transition down to a production node responsible for the emission of "`Undefined`" observations that correspond to the subject not being in proximity to any area of interest as they walk towards the couch. The remainder of the production nodes in this subtree appear to be responsible for encoding the duration that the person spends at the couch. This subtree has therefore been labeled "`Read on couch`." Termination of the "`Read on couch`" subtree now results in a horizontal transition to the rightmost internal state. Here the production nodes will emit "`Undefined`," "`Near Bookcase`" and "`Near Door`" observations corresponding to the subject leaving the room.

The trained model for the third lounge room style, "eat dinner," is given in Figure 3.9. This model is seen to feature a slightly disarranged transitional structure not seen in previous examples with the children nodes of the second and third internal state sharing production of the "`Near Bookcase`" and "`Near Table`" observations. These observations correspond to the subject spending time at the dinner table and coming into close proximity to the bookcase due to noisy observations from the tracker. The two subtrees also share a weak cyclic transitional relationship and have therefore been

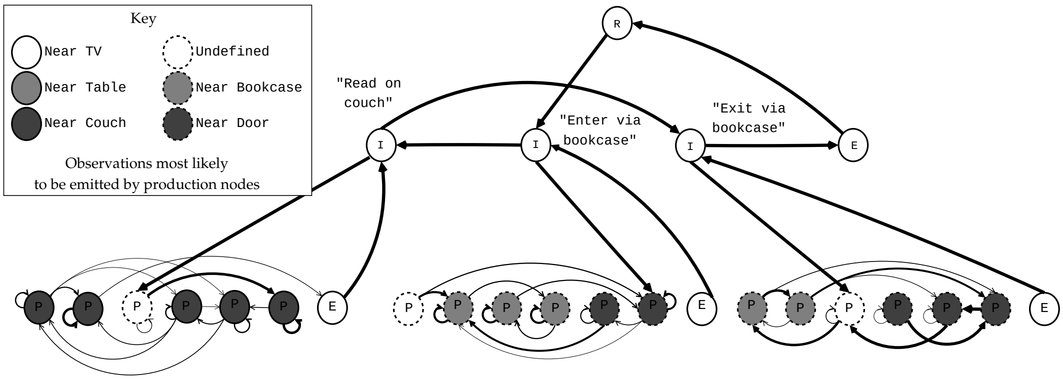


Figure 3.8: The trained model for the “read book on couch” lounge style. Darker arrows indicate stronger state dependence and a higher transitional likelihood among the production (P), internal (I) and end (E) states. Insignificant transitions have been omitted for clarity.

labeled “`Eat dinner`.”

The leftmost internal state continues to exhibit the “`Enter/exit`” structure seen previously, the subtree having learned to generate the “`Near Door`” and “`Near Bookcase`” sequence segments corresponding to the subject walking past the bookcase when entering and leaving the room.

Figure 3.10 depicts the HHMM for the “eat dinner while watching TV” lounge room style. The subtrees in this model have once again attained a well-defined structure in which the first, second and third internal states correspond to the “`Eat dinner`,” “`Enter/exit`” and “`Walk by TV`” segments of the sequences being modelled.

The final HHMM, trained on the “there is nothing good on TV, read a book instead” activity sequence style, is depicted in Figure 3.11. The transition structure of this model is slightly more complicated than other trained models with the rightmost internal state and its children production nodes corresponding to the sequence segments of finding a book and both watching television and reading on the couch. This node has been assigned the label “`Find book/use couch`” in Figure 3.11. The first and second internal states appear to correspond to the production of observations corresponding to the subject entering the room via the television and exiting the room via the bookcase. These two nodes have hence been labeled “`Enter via TV`” and “`Exit via bookcase`” respectively.

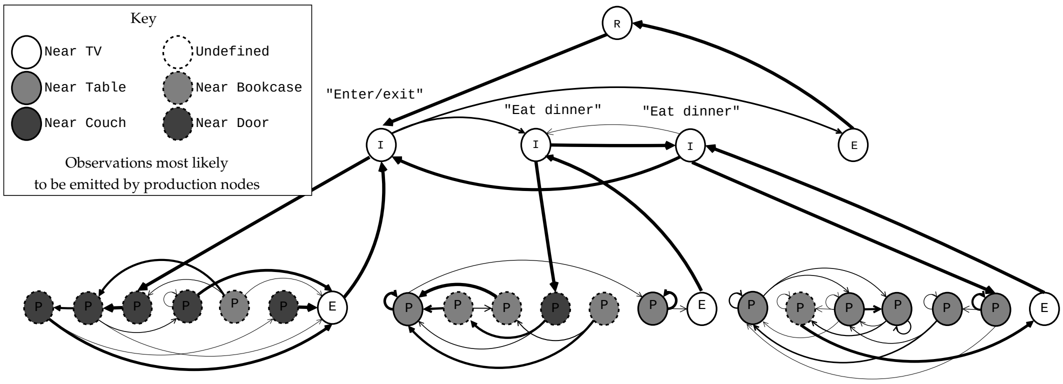


Figure 3.9: The trained model for the "eat dinner" lounge style. Darker arrows indicate stronger state dependence and a higher transitional likelihood among the production (P), internal (I) and end (E) states. Insignificant transitions have been omitted for clarity.

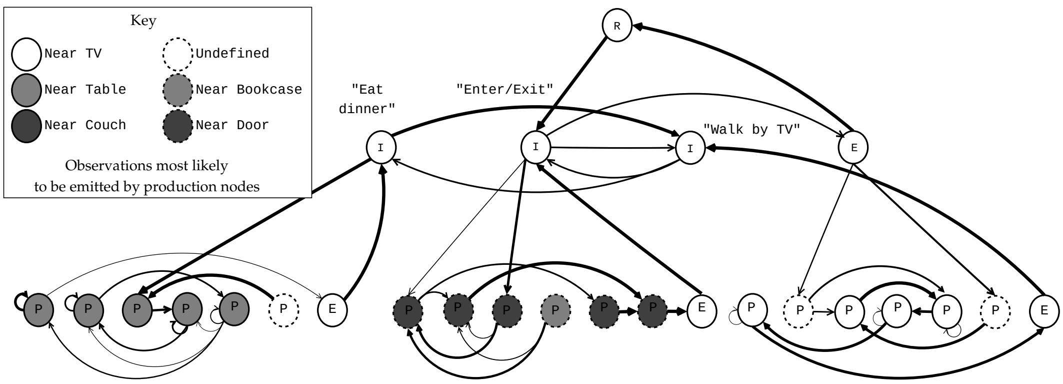


Figure 3.10: The trained model for the "eat dinner while watching TV" lounge style. Darker arrows indicate stronger state dependence and a higher transitional likelihood among the production (P), internal (I) and end (E) states. Insignificant transitions have been omitted for clarity.

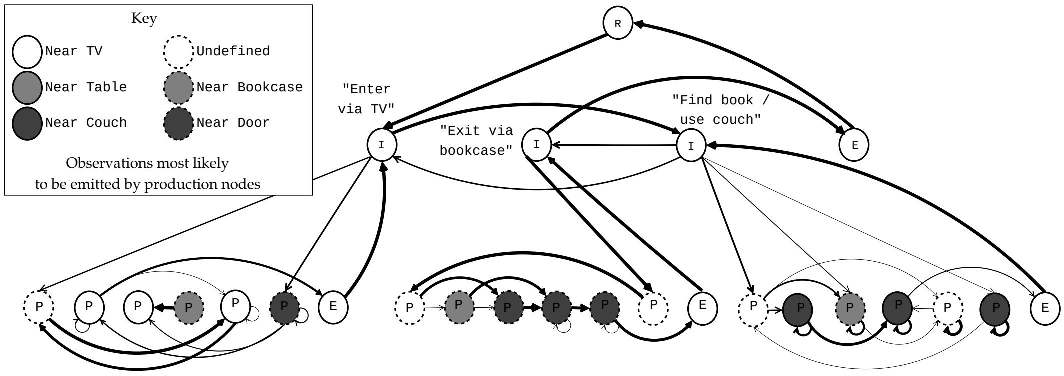


Figure 3.11: The trained model for the "there is nothing good on TV, read a book instead" lounge style. Darker arrows indicate stronger state dependence and a higher transitional likelihood among the production (P), internal (I) and end (E) states. Insignificant transitions have been omitted for clarity.

Table 3.1: Confusion matrix for the cooking test sequences.

	Prepare, cook	Dishes, cook	Wash, prepare, cook	Round robin
Prepare, cook	8	0	1	0
Dishes, cook	0	8	1	0
Wash, prepare, cook	0	0	9	0
Round robin	0	0	0	9

3.1.3 Classification Results

Eighty one unseen sequences were used to test the ability of the HHMM to classify activity sequences. The cooking and lounge room models were tested only on the cooking and lounge room sequences respectively as the person's locality indicates, in this particular case, the models of interest. The test sequences were designed to include major variations of duration and, in some cases, variations in transition order between the areas of interest.

The classification results of the cooking and the lounge room activity test sequences, presented in the confusion matrices in Table 3.1 and Table 3.2 respectively, show reasonable classification accuracy and demonstrate that variations in the order of activities are accommodated by the model.

Table 3.2: Confusion matrix for the lounge room test sequences.

	Watch TV	Read (couch)	Dinner	TV dinner	TV then read
Watch TV	8	0	0	0	1
Read on couch	0	8	0	0	1
Dinner	0	0	9	0	0
TV dinner	0	0	0	9	0
TV then read	0	0	0	0	9

3.1.4 Conclusion

This section has shown that the HHMM can be an effective tool for modelling and classifying human activity observed through a visual tracking system. Examination of the trained models revealed that the state transition structure and the emission likelihoods are easily interpreted. This affirms that the model is capable of learning a richer semantic structure than single layered "flat" models such as the HMM. The ability of the HHMM to learn the higher level relationships between the submodels responsible for production of observations is also promising as these lower level models can be reused in the training of increasingly complex higher level models to encompass a subject's longer term actions and activities.

Several limiting factors were observed, however. First, the cubic complexity of the HHMM with regards to the sequence length limits the practical application of the model towards longer term observation sequences. Numerical underflow also remains a problem due to a lack of scaling in implementations of the HHMM[1]. Finally, it is hypothesized that duration is an important feature in models of human activity. Incorporating duration into such models is necessary to prevent confusion between observation sequences were the ordering of activities is similar yet differs significantly in the duration of those activities. It is also expected to be a vital component in the detection of anomalous behaviour due to unusual durations in activity. The lack of explicit activity duration modelling in any stochastic model is therefore predicted to render models such as the HMM and the HHMM inadequate in such situations. The importance of duration shall therefore be investigated next.

[1]This issue has recently been resolved in Phung (2005), some time after this work was undertaken.

3.2 Explicit State Duration HMM

State duration in the standard HMM is implied as a function of a state's self transition probability. Given a state i and its self transition probability a_{ii}, we can show that the likelihood of remaining in the state for d consecutive time steps is exponential:

$$(a_{ii})^{d-1} \cdot (1 - a_{ii}) \,. \tag{3.6}$$

The implied duration model of the HMM and HHMM is expected to create situations in which highly abnormal deviations in expected behaviour as either less than or greater than usually observed activity duration are accommodated by the self transitions and can hence fail to be detected. In this section, the notion that the incorporation of duration into a model of human behaviour is necessary in order to properly recognise and detect duration abnormality will be investigated.

3.2.1 Experimentation Methodology

A further 150 video sequences of normal activity in a kitchen scenario were recorded using a single camera. Each sequence belongs to one of five normal classes of activity sequences one might observe in a kitchen: preparing cereal, making toast for breakfast, preparing or reheating dinner and cooking a bacon and eggs breakfast. The defined areas of interest in the room were: the stove, the kitchen bench, the sink, a fridge and the door. The discrete observations `stove`, `bench`, `sink`, `fridge` and `door` were returned by the visual tracker if the subject was in close proximity else an `undefined` observation was logged. The duration of the new video sequences ranged from 30 to 300 seconds with an average length of circa 90 seconds. Although a richer set of features from multiple camera angles would be beneficial in a real world deployment, these observations were found to be adequate for demonstrating the ideas behind this work.

Normal Activity Sequences

The five classes of normal behaviour were designed to highlight the importance of modelling duration given the limitations of the tracking system where classes may have the same sequence of observations but can differ in the duration spent in a location.

The first two classes, preparing cereal and making toast for breakfast, are identical in the order that the areas of interest in the room are visited by the person under observation. Hence, given that the order of observations returned by the tracker are `door`, `fridge`, `bench`, `sink`, `bench`, `fridge`, `door` it is only possible to distinguish between the two classes by observing the time spent at the kitchen bench, the act of making toast taking considerably longer than the preparation of a bowl of cereal.

Similarly, the dinner preparation and reheating classes consist of the activities `door`, `fridge`, `bench`, `stove`, `door`, the classes differing only in the duration spent standing by the stove. The fifth class is made up of the activities `door`, `fridge`, `bench`, `sink`, `bench`, `stove`, `fridge`, `door`. It was included because it differs to the other classes in both the activity duration and the order in which the activities are performed. The five activity sequences were evenly distributed among the recorded video sequences.

Abnormal Activity Sequences

A further 24 sequences of abnormal behaviour were recorded. The abnormal sequences differ from the normal only in terms of activity duration, either shorter or longer than the durations seen in the normal classes, not in the order or type of activities seen.

Model Selection

Each normal class was modelled using a standard fully connected HMM, a left-right HMM, a fully connected ESD-HMM and a left-right ESD-HMM. The left-right models were chosen to investigate how constraining the state transitions would affect classification and abnormality detection by preventing the models from treating duration as a cyclic activity. The HMM was selected as a baseline for comparison. The models were trained on a random sample of 60% of the normal activity sequences and tested on the remainder. To keep the comparison fair, an optimal number of states for each model was empirically selected based on classification accuracy.

A single Gaussian distribution was used to estimate the duration probabilities in the ESD-HMM case, the model otherwise requiring an unrealistic amount of training data to accurately estimate the state duration probabilities.

A comparison of the classification results for the various models will be presented next in Section 3.2.2. Duration abnormality detection is then discussed in Section 3.2.3. Finally, the models' ability to function under varying degrees of duration abnormality will be discussed in Section 3.2.4.

3.2.2 Sequence Classification

Classification accuracy and the optimal number of states for each of the four models are presented in Table 3.3. The HMM was found to be the weakest model for classification. Its low score is attributed to dynamic time warping, a property which renders it unsuitable for use as a classifier given the type of observation sequences used in this experiment. This is also evident in its relatively poor classification of the original training data.

Forcing the HMM to be a left-right model, that is $A_{ij} = 0$ for all $j < i$, improves classification accuracy with near perfect results. Although the two state left-right HMM performed well empirically, the limited number of parameters is inadequate to properly encode the sequences and hence discriminate between classes. Confusion between the similar activity classes is shown in Table 3.4.

The ESD-HMM, in contrast, appears to perform well given no state transition restrictions, providing no room for further improvement when left-right constraints are imposed. Explicit duration allows the model to clearly differentiate between all classes. The confusion matrix for the explicit state duration models is presented in Table 3.5.

3.2.3 Duration Abnormality

In this experiment, unseen observation sequences are classified as either normal or abnormal by querying each of the trained models on the likelihood of generating a given sequence and then thresholding on the highest log-likelihood returned. The log-likelihoods are normalised by the total length of the observation sequence so that a global threshold may be applied regardless of sequence length. Receiver operator characteristic (ROC) curves were used to investigate the suitability of each of the models as a detector of abnormality. The observation sequences used in this experiment consisted of the set of unseen normal sequences and the abnormal sequences described in Section 3.2.1.

The ROC curves in Figure 3.12(a) and Figure 3.12(b) for the HMM and the left-right HMM respectively suggest that neither model is able to reliably differentiate between the normal and abnormal sequences using the thresholding approach.

The ROC curve shown in Figure 3.12(c) for the ESD-HMM displays better results with the true positive rate increasing more rapidly than the false positive rate. The use of explicit state duration has increased the reliability of the HMM in the detection

Table 3.3: The optimal number of states and classification accuracy for both the training and the test sequences for each of the four model types.

Model	# States	Training	Testing
HMM	12	93.75%	81.43%
Left-right HMM	2	98.75%	97%
ESD-HMM	3	100%	100%
Left-right ESD-HMM	2	98.75%	100%

Table 3.4: Test sequence confusion matrix for the twelve state standard HMM.

	Cereal	Toast	Cook Dinner	Reheat Dinner	Bacon & Eggs
Cereal	11	3	0	0	0
Toast	0	14	0	0	0
Cook Dinner	0	0	10	4	0
Reheat Dinner	0	0	6	7	0
Bacon & Eggs	0	0	0	0	14

Table 3.5: Test sequence confusion matrix for both the three state ESD-HMM and the two state left-right ESD-HMM.

	Cereal	Toast	Cook Dinner	Reheat Dinner	Bacon & Eggs
Cereal	14	0	0	0	0
Toast	0	14	0	0	0
Cook Dinner	0	0	14	0	0
Reheat Dinner	0	0	0	14	0
Bacon & Eggs	0	0	0	0	14

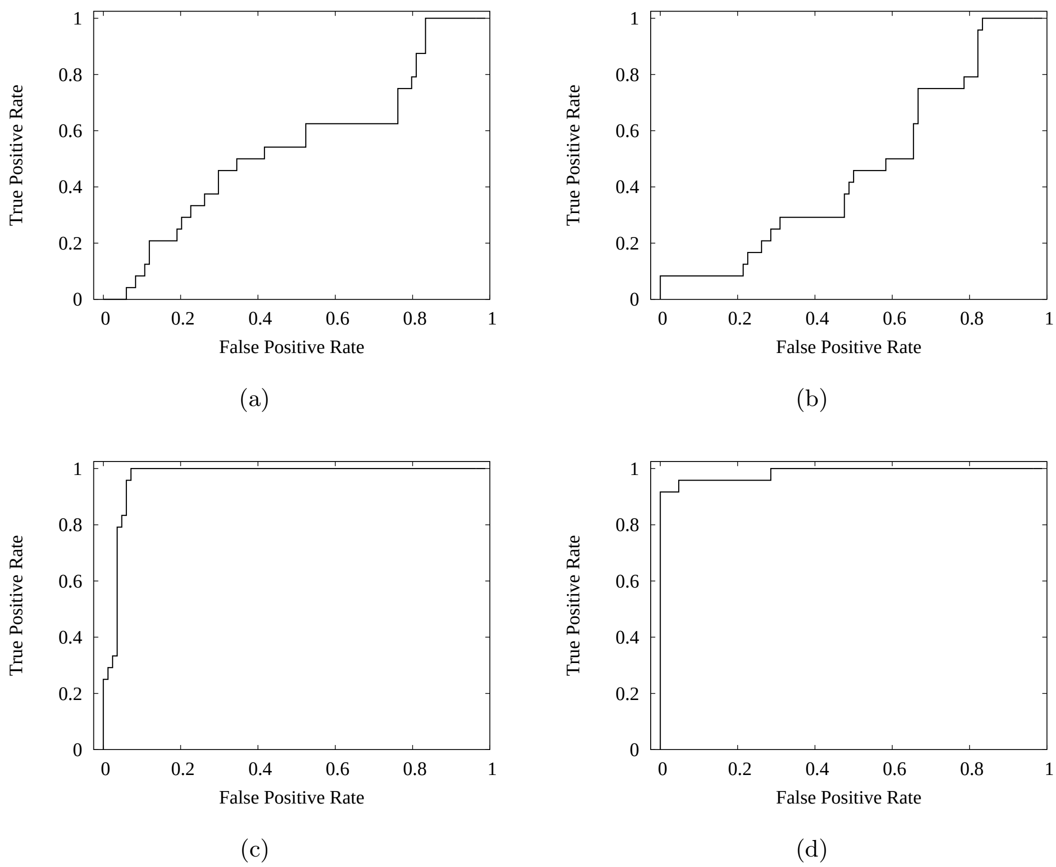


Figure 3.12: ROC curves for (a) the twelve state HMM, (b) the two state left-right HMM, (c) the three state ESD-HMM and (d) the two state left-right ESD-HMM.

of abnormality due to the presence of unusual activity duration. The main cause of the remaining misclassification was found to be due to the model freely transitioning between states. The model appears to temporarily enter a state with sub-optimal emission probabilities prior to returning to the original state so as to maximise the state duration likelihoods over the entire length of a given sequence.

Further improvement is seen when the transition constraints of the two state left-right ESD-HMM are imposed as evidenced by the steep ascent of the true positive rate in Figure 3.12(d).

Analysis of the errors showed that two of the normal activity sequences had been misclassified by the explicit state duration models because they contained a noisy observation, uncommon and not present in the training data, in the middle of a typically long activity. The models were forced to make a transition to another state in order to emit the rogue observation, leading to very low state duration probabilities.

3.2.4 Longer Term Duration Abnormality

This experiment aims to investigate the ability of the models to detect longer term abnormal duration. The time spent at a primary activity, standing near the kitchen bench, in a randomly selected test sequence from the first activity class was artificially varied from one second to five minutes. The usual time for a subject to remain at the kitchen bench is circa forty seconds.

The log-likelihood, normalised by the length of the observation sequence, of the modified activity sequence being generated by each of the standard HMM, left-right HMM, ESD-HMM and left-right ESD-HMM was plotted over the duration period and is presented in Figure 3.13. The figure shows the normalised log-likelihood returned by the HMM and the left-right HMM increasing with the time spent at the primary activity due to dynamic time warping. The HMM and left-right HMM are therefore not suitable for the detection of highly abnormal activity duration.

The ESD-HMM exhibits a similar trend. The lack of transition constraints allows the model to temporarily enter a state with a sub-optimal emission probability in order to maximise the state duration likelihoods.

The left-right ESD-HMM behaves correctly given the intention of the system; the model is unable to explain away highly abnormal duration as a cyclic activity and thus identify them as abnormal. The curve is seen to drop rapidly as the time spent at the kitchen

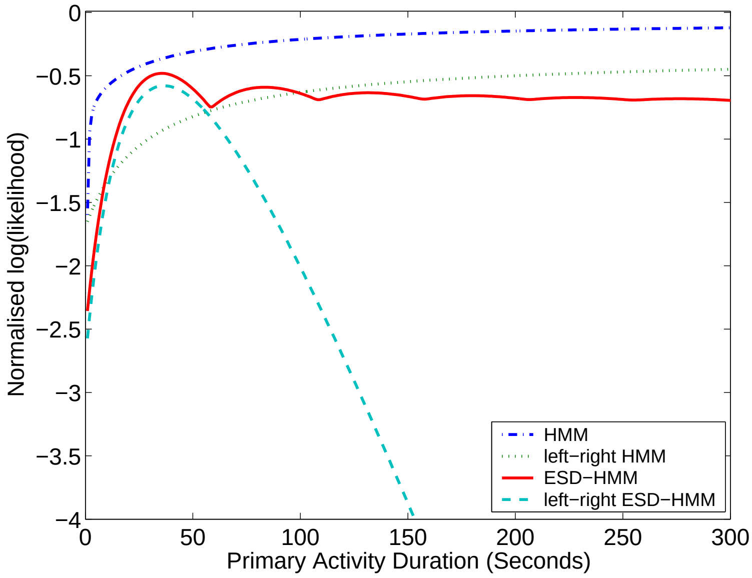


Figure 3.13: The normalised log-likelihood for each of the models as the primary activity in an observation sequence is varied from one second to five minutes. The normal duration for the primary activity is circa 40 seconds. Only the left-right ESD-HMM is able to detect abnormality in a timely manner.

bench increases.

3.2.5 Conclusion

This section has shown that the incorporation of duration in models of human activity is necessary. The application of ESD-HMM to the task of detecting both abnormally long and abnormally short activity durations in sequences of human activity was investigated. It was shown that the use of left-right state transition constraints are necessary for the detection of higher order duration anomalies.

The optimal number of states for the left-right ESD-HMM was found to be two, making the model unsuitable to properly encode the observation sequences. A generative process applied to this model is hence unlikely to produce a faithful reproduction of the original signal. The implicit nature of the state durations in the model appears to be of issue here. A possible solution to alleviate this problem is tackled in Section 3.3 with the introduction of explicitly known state durations that are derived from an augmented observation signal.

3.3 Explicit State Duration HMM With Observed Time Indices

Experimentation with the ESD-HMM in Sections 3.2.2–3.2.4 showed that explicit state duration modelling allowed abnormality in the form of unusually short or long activity durations to be more accurately detected than with the standard HMM. The low number of optimal states for the ESD-HMM, however, demonstrated that these models are unlikely to faithfully reproduce the activity sequences upon which they are trained due to the inherent smoothing performed by the models. In this section, observed state transition time indices are introduced into the ESD-HMM to alleviate this issue by augmenting the activity sequence signal with time indices that correspond to a subject stepping onto pressure mats deployed around the home. Strategic placement of the pressure mat sensors can, for example, ensure that ESD-HMM state transition times align with the subject entering clearly defined regions or rooms in the home.

Proof of the correctness of the Observed Time Indices (OTI) ESD-HMM model is presented via its derivation in Section 3.3.1. Results from repeat experimentation on the data set from Section 3.2.1 are provided and discussed in Section 3.3.2 and in Section 3.3.3.

3.3.1 Model Derivation

The notation used in the derivation of the OTI ESD-HMM is as follows. Let K be the number of states that will be visited in the model and let T be the number of observations. The sequence of states visited for some $K \leq T$ is said to be $Q = [q_1 \dots q_k \dots q_K]$ with the duration spent in each of the states given as $\Phi = [\phi_1 \dots \phi_k \dots \phi_K]$ such that $\sum_{k=1}^{K} \phi_k = T$. The observed starting times of each state q_k is given by the vector $\tau = [\tau_1 \dots \tau_k \dots \tau_K]$ which provides a mapping from k to t such that $\tau_1 = 1$ and $\tau_K + \phi_K - 1 = T$. The observed ending times of each state q_k is similarly given by the vector $\tau' = [\tau'_1 \dots \tau'_k \dots \tau'_K]$ which provides a mapping from k to t such that $\tau'_k = \tau_k + \phi_k - 1$ and $\tau'_K = T$. The observations emitted over time t are $O = [o_1 \dots o_t \dots o_T]$.

The conditional dependencies of the model shown in Figure 3.14 allow the joint probability to be defined as

$$\begin{aligned}&\Pr(o_1 \dots o_T, q_1 \dots q_K, \phi_1 \dots \phi_K) \\ &\quad = \Pr(q_1)\Pr(\phi_1|q_1)\Pr\left(o_{\tau_1} \dots o_{\tau'_1}\right)\prod_{k=2}^{K}\Pr(q_k|q_{k-1})\Pr(\phi_k|q_k)\Pr\left(o_{\tau_k} \dots o_{\tau'_k}|q_k\right). \quad (3.7)\end{aligned}$$

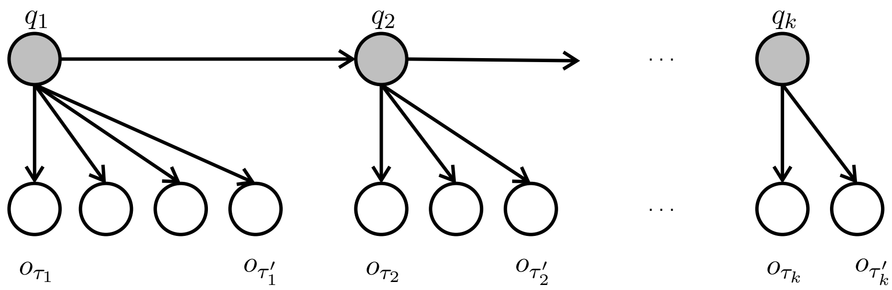


Figure 3.14: The OTI ESD-HMM rolled out over time. The states $q_1 \ldots q_K$ are seen to produce the observations $o_{\tau_1} \ldots o_{\tau'_K}$ over k sequence segments. The starting and ending time of each state k is given as τ_k and τ'_k respectively. The duration of each state is ϕ_k such that $\tau'_k - \tau_k = \phi_k + 1$.

Forwards recursion will seek:

$$\begin{aligned}
&\Pr\left(o_1 \ldots o_{\tau'_k}, q_k, \phi_k\right) \\
&\quad = \sum_{q_{k-1}} \Pr\left(o_1 \ldots o_{\tau'_k}, q_k, \phi_k, q_{k-1}\right) && (3.8) \\
&\quad = \sum_{q_{k-1}} \Pr\left(o_{\tau_k} \ldots o_{\tau'_k} | q_k\right) \Pr\left(q_k | q_k\right) \Pr\left(\phi_k | q_k\right) \\
&\qquad \times \Pr\left(o_1 \ldots o_{\tau'_{k-1}}, q_1 \ldots q_{k-1}, \phi_1 \ldots \phi_{k-1}\right). && (3.9)
\end{aligned}$$

The forwards path variable $\alpha_k(j,d)$ is defined as:

$$\alpha_k(j,d) \triangleq \Pr\left(o_1 \ldots o_{\tau'_k}, q_k = j, \phi_k = d\right) \tag{3.10}$$

$$\begin{aligned}
\alpha_k(j,d) = &\sum_{\substack{i=1 \\ i \neq j}}^{N} \Pr\left(o_{\tau_k} \ldots o_{\tau'_k} | q_k = j\right) \Pr\left(q_k = j | q_{k-1} = i\right) \Pr\left(\phi_k = d | q_k = j\right) \\
&\times \Pr\left(o_1 \ldots o_{\tau'_{k-1}}, q_1 \ldots q_{k-1}, \phi_1 \ldots \phi_{k-1}\right).
\end{aligned} \tag{3.11}$$

Taking care to note that:

$$\alpha_1(j,d) \triangleq \Pr\left(o_1 \ldots o_{\tau'_1}, q_1 = j, \phi_1 = d\right) \tag{3.12}$$

$$\alpha_1(j,d) = \Pr\left(o_1 \ldots o_{\tau'_1} | q_1 = j\right) \Pr\left(q_1 = j\right) \Pr\left(\phi_1 = d | q_1 = j\right). \tag{3.13}$$

Backwards recursion will seek:

$$
\begin{aligned}
&\Pr\left(o_{\tau_{k+1}} \dots o_T, q_{k+1}, \phi_{k+1} | q_k\right) \\
&\quad = \sum_{q_{k+1}} \Pr\left(o_{\tau_{k+1}} \dots o_T, q_{k+1}, \phi_{k+1} | q_k\right) && (3.14) \\
&\quad = \sum_{q_{k+1}} \Pr\left(o_{\tau_{k+1}} \dots o_{\tau'_{k+1}} | q_{k+1}\right) \Pr\left(q_{k+1} | q_k\right) \Pr\left(\phi_{k+1} | q_{k+1}\right) \\
&\qquad \times \Pr\left(o_{\tau_{k+2}} \dots o_T, q_{k+2} \dots q_K, \phi_{k+2} \dots \phi_K | q_{k+1}\right). && (3.15)
\end{aligned}
$$

The backwards path variable $\beta_k(i,d)$ is defined as:

$$
\beta_k(i,d) \triangleq \Pr\left(o_{\tau_{k+1}} \dots o_T, q_{k+1} = j, \phi_{k+1} = d | q_k = i\right) \tag{3.16}
$$

$$
\begin{aligned}
\beta_k(i,d) = &\sum_{\substack{j=1 \\ j \neq i}}^{N} \Pr\left(o_{\tau_{k+1}} \dots o_{\tau'_{k+1}} | q_{k+1} = j\right) \Pr\left(q_{k+1} = j | q_k = i\right) \Pr\left(\phi_{k+1} | q_{k+1} = j\right) \\
&\times \Pr\left(o_{\tau'_{k+2}} \dots o_T, q_{k+2} \dots q_K, \phi_{k+2} \dots \phi_K | q_{k+1} = j\right).
\end{aligned} \tag{3.17}
$$

We arbitrarily set $\beta_K(i,d) = 1$ given that the calculation of $\beta_K(i,d)$ requires the non-existent observation symbol at $o_{\tau k+1}$.

Sufficient Statistics

The model will now be parameterised. Let A be the state transition matrix where $a_{i,j}$ is the $(i,j)^{\text{th}}$ entry in the state transition matrix and represents the probability $\Pr(q_k = j | q_{k-1} = i)$.

Let Π be a vector holding the initial state distribution such that π_i is the i^{th} entry of the initial state distribution and represents the probability $\Pr(q_1 = i)$.

Let B be the emission likelihood distribution where $b_i(m)$ is the probability of the i^{th} state emitting the observation symbol $V_m \in [v_1 \dots v_M]$.

Let P be the state duration likelihood distribution where $p_i(d)$ is the probability of the i^{th} state being active for d consecutive time steps.

The parameters of the entire model are denoted with the shorthand notation $\lambda = (\Pi, A, B, P)$. The identity function $\mathrm{I}()$ defines that $\mathrm{I} = 1$ iff all arguments passed are true.

The joint probability of the model in Equation 3.7 can now be parameterised:

$$
\begin{aligned}
&\Pr\left(o_1 \ldots o_T, q_1 \ldots q_K, \phi_1 \ldots \phi_K\right) \\
&\quad = \Pr\left(q_1\right) \Pr\left(\phi_1 | q_1\right) \Pr\left(o_1 \ldots o_{\tau_1} | q_1\right) \prod_{k=2}^{K} \Pr\left(q_k | q_{k-1}\right) \Pr\left(\phi_k | q_k\right) \\
&\qquad \times \Pr\left(o_{\tau_k} \ldots o_{\tau'_k} | q_k\right) && (3.18) \\
&\quad = \pi_{q_1} p_{q_1}\left(\phi_1\right) \prod_{t=1}^{\tau_1} b_{q_1}\left(o_t\right) \prod_{k=2}^{K} \left[a_{q_{k-1},q_k} p_{q_k}\left(\phi_k\right) \prod_{t=\tau_k}^{\tau'_k} b_{q_k}\left(o_t\right)\right] && (3.19) \\
&\quad = \pi_{q_1} p_{q_1}\left(\phi_1\right) \left[\prod_{k=2}^{K} a_{q_{k-1},q_k} p_{q_k}\left(\phi_k\right)\right] \left[\prod_{k=1}^{K} \prod_{t=\tau_k}^{\tau'_k} b_{q_k}\left(o_t\right)\right] && (3.20) \\
&\quad = \log\left\{\pi_{q_1} p_{q_1}\left(\phi_1\right) \prod_{k=2}^{K} a_{q_{k-1},q_k} p_{q_k}\left(\phi_k\right) \prod_{k=1}^{K} \prod_{t=\tau_k}^{\tau'_k} b_{q_k}\left(o_t\right)\right\} && (3.21) \\
&\quad = \log\left\{\prod_{i=1}^{N} \pi_i^{\mathrm{I}(q_1=i)}\right\} + \log\left\{\prod_{k=2}^{K} \prod_{i=1}^{N} \prod_{\substack{j=1 \\ j \neq i}}^{N} a_{i,j}^{\mathrm{I}(q_{k-1}=j, q_k=j)}\right\} \\
&\qquad + \log\left\{\prod_{k=1}^{K} \prod_{d=1}^{D} \prod_{i=1}^{N} p_i\left(d\right)^{\mathrm{I}(q_k=i, \phi_k=d)}\right\} \\
&\qquad + \log\left\{\prod_{k=1}^{K} \prod_{t=\tau_k}^{\tau'_k} \prod_{m=1}^{M} \prod_{i=1}^{N} b_i\left(m\right)^{\mathrm{I}(q_k=i, o_t=m)}\right\} && (3.22) \\
&\quad = \sum_{i=1}^{N} \mathrm{I}\left(q_1 = i\right) \log\left(\pi_i\right) + \sum_{k=2}^{K} \sum_{i=1}^{N} \sum_{\substack{j=1 \\ j \neq i}}^{N} \mathrm{I}\left(q_{k-1} = i, q_k = j\right) \log\left(a_{i,j}\right) \\
&\qquad + \sum_{k=1}^{K} \sum_{d=1}^{D} \sum_{i=1}^{N} \mathrm{I}\left(q_k = i, \phi_k = d\right) \log\left(p_i\left(d\right)\right) \\
&\qquad + \sum_{k=1}^{K} \sum_{t=\tau_k}^{\tau'_k} \sum_{m=1}^{M} \sum_{i=1}^{N} \mathrm{I}\left(q_k = i, o_t = m\right) \log\left(b_i\left(m\right)\right). && (3.23)
\end{aligned}
$$

Thus, for the fully observed case the sufficient statistics are defined as:

$$\mathrm{SS}\left(\pi_i\right) = \mathrm{I}\left(q_1 = i\right) \tag{3.24}$$

$$\mathrm{SS}\left(a_{i,j}\right) = \sum_{k=2}^{K} \mathrm{I}\left(q_{k-1} = i, q_k = j\right) \tag{3.25}$$

$$\mathrm{SS}\left(b_i\left(m\right)\right) = \sum_{k=1}^{K} \sum_{t=\tau_k}^{\tau'_k} \mathrm{I}\left(q_k = i, o_t = m\right) \tag{3.26}$$

$$\mathrm{SS}\left(p_i\left(d\right)\right) = \sum_{k=1}^{K} \mathrm{I}\left(q_k = i, \phi_k = d\right). \tag{3.27}$$

Parameter Estimation For The Hidden Variable Case

Here the variables $q_i \dots q_T$ are hidden while the variables $o_1 \dots o_T$ and $\phi_1 \dots \phi_K$ are observed.

Given the parameterised path variables:

$$\alpha_1\left(j, d\right) = \pi_j p_j\left(d\right) \prod_{t=1}^{\tau'_1} b_j\left(o_t\right) \tag{3.28}$$

$$\alpha_k\left(j, d\right) = \sum_{\substack{i=1 \\ i \neq j}}^{N} \alpha_{k-1}\left(i, \phi_{k-1}\right) a_{i,j} p_j\left(d\right) \prod_{t=\tau_k}^{\tau'_k} b_j\left(o_t\right) \tag{3.29}$$

$$\beta_K\left(i, d\right) = 1 \tag{3.30}$$

$$\beta_k\left(i, d\right) = \sum_{\substack{j=1 \\ j \neq i}}^{N} a_{i,j} p_j\left(d\right) \left[\prod_{t=\tau_{k+1}}^{\tau'_{k+1}} b_j\left(o_t\right)\right] \beta_{k+1}\left(j, \phi_{k+2}\right) \tag{3.31}$$

The expected sufficient statistics are thus:

$$\mathrm{ESS}\left(\pi_i\right) = \mathrm{E}\left[\mathrm{I}\left(q_1 = i\right) | o_1 \dots o_T, \phi_1 \dots \phi_K\right] \tag{3.32}$$

$$= \Pr\left(q_1 = i | o_1 \dots o_T, \phi_1 \dots \phi_K\right) \tag{3.33}$$

$$= \Pr\left(q_1 = i | o_1 \dots o_{\tau'_1}, o_{\tau_2} \dots o_T, \phi_1, \phi_2 \dots \phi_K\right) \tag{3.34}$$

$$= \pi_i p_i\left(\phi_1\right) \left[\prod_{t=1}^{\tau'_1} b_i\left(o_t\right)\right] \beta_1\left(i, \phi_2\right) \tag{3.35}$$

$$\mathrm{ESS}\left(a_{i,j}\right) = \mathrm{E}\left[\sum_{k=2}^{K} \mathrm{I}\left(q_{k-1} = i, q_k = j\right) | o_1 \ldots o_T, \phi_1 \ldots \phi_K\right] \tag{3.36}$$

$$= \sum_{k=2}^{K} \Pr\left(q_{k-1} = i, q_k = j | o_1 \ldots o_T, \phi_1 \ldots \phi_K\right) \tag{3.37}$$

$$= \sum_{k=2}^{K} \Pr\left(q_k = j | q_{k-1} = i\right) \Pr\left(o_1 \ldots o_T, \phi_1 \ldots \phi_K | q_{k-1} = i\right) \tag{3.38}$$

$$= \sum_{k=2}^{K} \Pr\left(q_k = j | q_{k-1} = i\right)$$

$$\times \ \Pr\left(o_1 \ldots o_{\tau'_{k-1}}, o_{\tau_k} \ldots o_{\tau'_k}, o_{\tau_{k+1}} \ldots o_T, \phi_1 \ldots \phi_{k-1}, \phi_k, \phi_{k+1} \ldots \phi_K | q_{k-1} = i\right) \tag{3.39}$$

$$= \sum_{k=2}^{K} \alpha_{k-1}\left(i, \phi_{k-1}\right) a_{i,j} p_j\left(\phi_k\right) \left[\prod_{t=\tau_k}^{\tau'_k} b_j\left(o_t\right)\right] \beta_k\left(j, \phi_{k+1}\right) \tag{3.40}$$

$$\mathrm{ESS}\left(b_i\left(m\right)\right) = \mathrm{E}\left[\sum_{k=1}^{K} \sum_{t=\tau_k}^{\tau'_k} \mathrm{I}\left(q_k = i, o_t = m\right) | o_1 \ldots o_T, \phi_1 \ldots \phi_K\right] \tag{3.41}$$

$$= \sum_{k=1}^{K} \sum_{t=\tau_k}^{\tau'_k} \mathrm{I}\left(o_t = m\right) \Pr\left(q_k = i, o_t = m | o_1 \ldots o_T, \phi_1 \ldots \phi_K\right) \tag{3.42}$$

$$= \sum_{t=1}^{\tau'_1} \mathrm{I}\left(o_t = m\right) \Pr\left(q_1 = i | o_1 \ldots o_T, \phi_1 \ldots \phi_K\right) \delta(o_t = m)$$

$$+ \sum_{k=2}^{K} \sum_{t=\tau_k}^{\tau'_k} \mathrm{I}\left(o_t = m\right) \Pr\left(q_k = i | o_1 \ldots o_T, , \phi_1 \ldots \phi_K\right) \tag{3.43}$$

$$= \sum_{t=1}^{\tau'_1} \mathrm{I}\left(o_t = m\right) \pi_i p_i\left(\phi_1\right) \left[\sum_{t=1}^{\tau'_1} b_i\left(o_t\right)\right]$$

$$+ \sum_{k=2}^{K} \sum_{t=\tau_k}^{\tau'_k} \mathrm{I}\left(o_t = m\right) \alpha_k\left(i, \phi_k\right) \beta_k\left(i, \phi_{k+1}\right) \tag{3.44}$$

$$\begin{aligned}
\mathrm{ESS}\left(p_i\left(d\right)\right) &= \mathrm{E}\left[\sum_{k=1}^{K} \mathrm{I}\left(q_k = i, \phi_k = d\right) | o_1 \ldots o_T\right] && (3.45)\\
&= \sum_{k=1}^{K} \mathrm{I}\left(\phi_k = d\right) \Pr\left(q_k = i, \phi_k = d | o_1 \ldots o_T\right) && (3.46)\\
&= \sum_{k=1}^{K} \mathrm{I}\left(\phi_k = d\right) \Pr\left(q_k = i | o_1 \ldots o_T\right) \Pr\left(\phi_k = d | q_k = i\right) && (3.47)\\
&= \sum_{k=1}^{K} \mathrm{I}\left(\phi_k = d\right) \Pr\left(q_k = i | o_1 \ldots o_{\tau'_{k-1}}, o_{\tau_k} \ldots o_T\right) \Pr\left(\phi_k = d | q_k = i\right) && (3.48)\\
&= \sum_{k=1}^{K} \mathrm{I}\left(\phi_k = d\right) \alpha_k\left(i, d\right) \beta_k\left(i, \phi_{k+1}\right) && (3.49)
\end{aligned}$$

Finally, the re-estimation formulas are given as:

$$\hat{\pi}_i = \frac{\pi_i p_i\left(\phi_1\right) \left[\sum_{t=1}^{\tau'_1} b_i\left(o_t\right)\right] \beta_1\left(i, \phi_2\right)}{\sum_{j=1}^{N} \pi_i p_i\left(\phi_1\right) \left[\sum_{t=1}^{\tau'_1} b_i\left(o_t\right)\right] \beta_1\left(i, \phi_2\right)} \tag{3.50}$$

$$\hat{a}_{i,j} = \frac{\sum_{k=2}^{K} \alpha_{k-1}\left(i, \phi_{k-1}\right) a_{i,j} p_j\left(\phi_k\right) \left[\prod_{t=\tau_k}^{\tau'_k} b_j\left(o_t\right)\right] \beta_k\left(j, \phi_{k+1}\right)}{\sum_{j=1}^{N} \sum_{k=2}^{K} \alpha_{k-1}\left(i, \phi_{k-1}\right) a_{i,j} p_j\left(\phi_k\right) \left[\prod_{t=\tau_k}^{\tau'_k} b_j\left(o_t\right)\right] \beta_k\left(j, \phi_{k+1}\right)} \tag{3.51}$$

$$\hat{b}_i\left(m\right) = \frac{\sum_{t=1}^{\tau'_1} \mathrm{I}\left(o_t = m\right) \pi_i p_i\left(\phi_1\right) \left[\sum_{t=1}^{\tau'_1} b_i\left(o_t\right)\right] + \sum_{k=2}^{K} \sum_{t=\tau_k}^{\tau'_k} \mathrm{I}\left(o_t = m\right) \alpha_k\left(i, \phi_k\right) \beta_k\left(i, \phi_{k+1}\right)}{\sum_{t=1}^{\tau'_1} \pi_i p_i\left(\phi_1\right) \left[\sum_{t=1}^{\tau'_1} b_i\left(o_t\right)\right] + \sum_{k=2}^{K} \sum_{t=\tau_k}^{\tau'_k} \alpha_k\left(i, \phi_k\right) \beta_k\left(i, \phi_{k+1}\right)} \tag{3.52}$$

$$\hat{p}_i(d) = \frac{\sum_{k=1}^{K} \mathrm{I}(\phi_k = d)\, \alpha_k(i,d)\, \beta_k(i,\phi_{k+1})}{\sum_{j=1}^{N}\sum_{k=1}^{K} \mathrm{I}(\phi_k = d)\, \alpha_k(j,d)\, \beta_k(j,\phi_{k+1})}. \tag{3.53}$$

3.3.2 Sequence Classification

The OTI ESD-HMM was applied to the problem of sequence classification using the data sets described in Section 3.1.2. A single Gaussian distribution was again used to estimate the state duration probabilities to avoid the need for an unrealistic amount of data for training. The state transition times used were manually generated when it was observed that the subject left an area of interest. The optimal number of states for the fully connected OTI ESD-HMM was found to be five. The sequence confusion matrix presented in Table 3.6 demonstrates that the model is capable of correctly classifying previously unseen activity sequences with the same level of accuracy as the three state ESD-HMM and the two state left-right ESD-HMM. Unlike these latter models, however, the five state OTI ESD-HMM is able to recreate the activity sequences when the generative process is applied.

3.3.3 Duration Abnormality

The duration abnormality experiment from Section 3.2.3 was repeated to investigate the suitability of the OTI ESD-HMM as a classifier of normal and abnormal activity sequences. Classification was again performed by thresholding on the highest log-

Table 3.6: Test sequence confusion matrix for the five state ESD-HMM with observed time indices.

	Cereal	Toast	Cook Dinner	Reheat Dinner	Bacon & Eggs
Cereal	14	0	0	0	0
Toast	0	14	0	0	0
Cook Dinner	0	0	14	0	0
Reheat Dinner	0	0	0	14	0
Bacon & Eggs	0	0	0	0	14

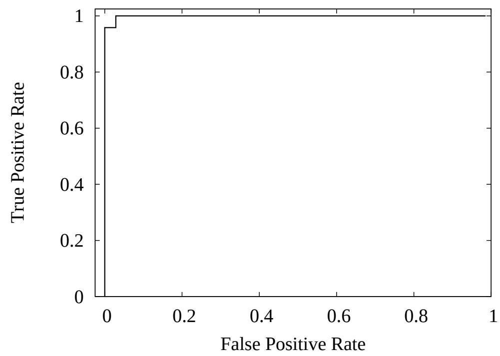


Figure 3.15: ROC curve for the five state ESD-HMM with observed time indices.

likelihood returned by each of the trained models when queried on the likelihood of generating the unseen normal and abnormal sequences described in Section 3.2.1. The model's classification performance is demonstrated via the ROC curve in Figure 3.15. The curve displays a slight improvement in classification accuracy over the two state left-right ESD-HMM in Figure 3.12.

A comparison between the fully connected OTI ESD-HMM and a left-right constrained OTI ESD-HMM was not possible on these data sets at the time this work was undertaken due to numerical underrun limitations. Such an investigation remains as future work.

3.3.4 Conclusion

This section has investigated the application of OTI ESD-HMM towards modelling sequences of human activity augmented with pressure pad sensor information. The ability of the five state OTI ESD-HMM to correctly classify unseen normal sequences was shown to match the accuracy of the three state ESD-HMM and the two state left-right ESD-HMM. The OTI ESD-HMM showed improved classification accuracy over to the standard ESD-HMM when applied to the detection of abnormal sequences via thresholding. The comparable performance of the OTI ESD-HMM combined with reduced parameter estimation complexity and improved sequence representation makes this model a viable alternative to the standard ESD-HMM when the state transition times are known.

3.4 Real World Application

Much of the focus of this chapter has been on the use of the HMM and ESD-HMM as classifiers of normal and abnormal behaviour from simple activity sequences recorded in a laboratory environment. This focus will now be shifted to consider the performance of the models using data collected from a real world environment with a comparison of the standard and left-right constrained HMM, and the fully connected and left-right constrained OTI ESD-HMM.

3.4.1 Experimentation Methodology

Two cameras were installed into the home of a volunteer for a period of two weeks to record the subject's daily morning routine as they prepared to leave for work. Figure 3.16 shows a layout of the subject's home and details the position of the cameras, the virtual pressure mat sensors and the regions of interest. The camera view of the home is shown in Figure 3.17.

The Gaussian background model used in the tracker for the laboratory experimentation was replaced with the Statistical And Knowledge-Based Object deTector (SAKBOT) algorithm proposed by Cucchiara et al. (2003), the former being unable to cope with the harsh lighting conditions experienced inside the home. The SAKBOT was modified to use the lighting model proposed by Greenhill et al. (2004) to increase the resistance of the algorithm to rapid changes in lighting. Further modification introduced a feedback loop from the Kalman filter to the background elimination model so as to collect an object's motion history for use in preventing the background model from moving an object with a history of motion into the background the moment it comes to rest. This modification was necessary to prevent the background model from classifying the subject as part of the background when they stand motionless in the scene.

As in previous experiments, the subject's proximity to known areas of interest was used to infer the nature of their activity. The observation codebook used in this experiment consisted of "`No Observation`", for when the subject was unable to be tracked, "`Front Door`," "`Rear Door`," "`Kitchen Table`," "`Food Preparation Area`," "`Kitchen Hallway`," "`Kitchen<->Lounge Hallway`," "`Lounge Hallway`," "`Lounge Rug`," "`Lounge Sofa`," "`Laundry`," "`Television`" and "`Laundry Entry`." Three further observations of "`Toilet`," "`Bathroom`" and "`Bedroom`" were manually annotated to avoid the need for cameras to be deployed in these rooms in order to protect the privacy of the subject.

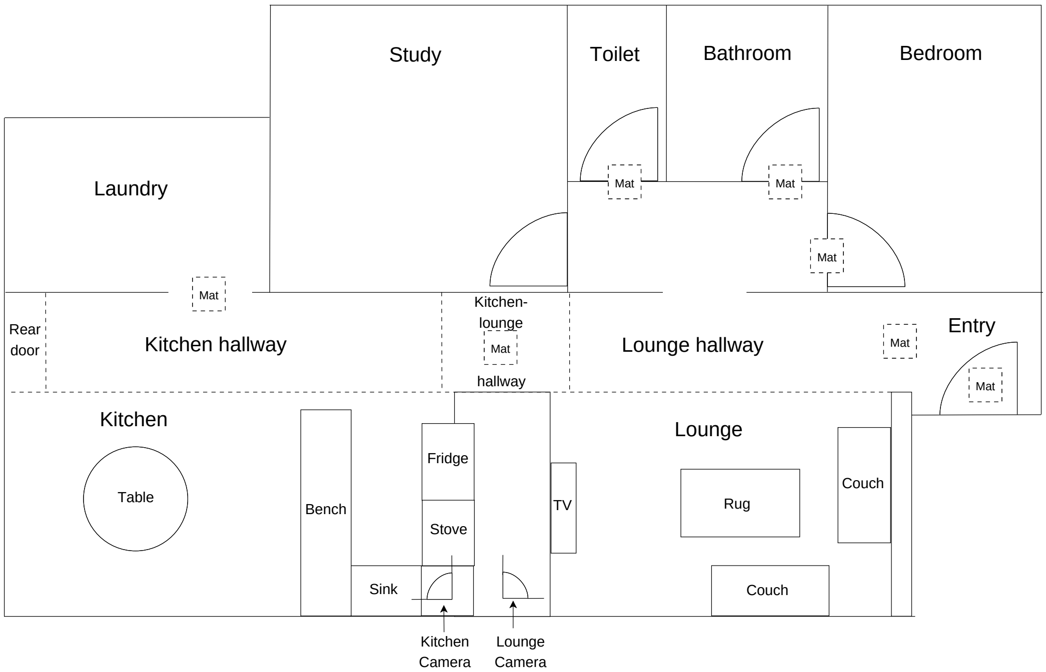


Figure 3.16: Approximate layout of the real world environment showing the location of the areas of interest, cameras and the virtual pressure mats.

(a) (b)

Figure 3.17: The real world scene as viewed by cameras deployed in (a) the kitchen and (b) the lounge room.

Nine usable days of footage were obtained from the ten weekdays captured, the discarded recording being unusable due to low lighting conditions. The recorded video was manually segmented into seven classes of often repeated activity: `Drink orange juice`, `Leave home`, `Walk from rear room to kitchen`, `Walk from kitchen to rear room`, `Visit bathroom`, `Visit bedroom` and `Visit toilet`. The numerical underrun limitation of the OTI ESD-HMM implementation, however, constrained experimentation to the `Drinking orange juice`, `Leave home`, `Walk from rear room to kitchen` and `Walk from kitchen to rear room` classes. The number of samples available for the four classes were 20, 11, 37 and 35 respectively.

The subject's location was again sampled once every 25 frames using the visual tracker. The state switching times were obtained through manual annotation of the video with the times that the subject is seen to trigger pressure mats virtually deployed around the home as outlined in Figure 3.16. Rather than assuming a Gaussian distribution, the state duration probabilities of the OTI ESD-HMM were linearly interpolated for this experiment. Domain knowledge was applied to find the number of states used to model the sequence classes. These are shown in Table 3.7.

3.4.2 Sequence Classification

Classification accuracy for each of the HMM, left-right HMM, OTI ESD-HMM and the left-right OTI ESD-HMM was measured using leave-one-out cross validation. The confusion matrices shown in Tables 3.8 through 3.11 display only minor variation between the models, suggesting that they are all equally suited to differentiating between the available sequences.

Confusion between the `Walk from rear room to kitchen` and the `Walk from kitchen to rear room` classes is apparent in all four models. Investigation revealed

Table 3.7: The number of states used to model the four real world sequence classes for the four models.

	HMM	LR HMM	OTI ESD-HMM	LR OTI ESD-HMM
Drink OJ	13	13	4	8
Leave Home	15	15	3	6
Rear Room to Kitchen	13	13	4	8
Kitchen to Rear Room	9	9	3	6

Table 3.8: Confusion matrix for the fully connected HMM.

	Drink OJ	Leave Home	Rear Room to Kitchen	Kitchen to Rear Room
Drink OJ	20	0	0	0
Leave Home	0	10	0	1
Rear Room to Kitchen	0	0	29	8
Kitchen to Rear Room	0	1	13	21

Table 3.9: Confusion matrix for the left-right constrained HMM.

	Drink OJ	Leave Home	Rear Room to Kitchen	Kitchen to Rear Room
Drink OJ	20	0	0	0
Leave Home	0	10	0	1
Rear Room to Kitchen	0	0	29	8
Kitchen to Rear Room	0	1	12	22

Table 3.10: Confusion matrix for the observed time indices ESD-HMM.

	Drink OJ	Leave Home	Rear Room to Kitchen	Kitchen to Rear Room
Drink OJ	19	0	0	0
Leave Home	0	9	1	0
Rear Room to Kitchen	0	2	29	6
Kitchen to Rear Room	1	0	9	25

Table 3.11: Confusion matrix for the observed time indices ESD-HMM with left-right constraints.

	Drink OJ	Leave Home	Rear Room to Kitchen	Kitchen to Rear Room
Drink OJ	17	1	0	0
Leave Home	1	10	0	0
Rear Room to Kitchen	0	2	28	7
Kitchen to Rear Room	1	0	12	22

that a high level of noise is present in these sequences in the form of "`No Observation`" observations and that the main cause of this noise is due to latency between the time the subject enters the scene and the tracker classifying them as a trackable object. Classification of these sequences is hence inherently difficult given that the mean duration of sequences within these two classes is approximately four seconds.

A number of unclassified sequences are present in the duration model results in Table 3.10 and Table 3.11. A single sequence was unable to be classified by the OTI ESD-HMM in both the `Drinking orange juice` and the `Leave home` classes. The cause of this error is due to the use of interpolated state duration probabilities combined with the leave-one-out cross validation; the unclassified sequence in each instance possessing a state duration outside the bounds permitted by the trained model. Two unclassified `Drinking orange juice` sequences in the results for the left-right OTI ESD-HMM, minor misclassification of a `Walk from kitchen to rear room` sequence as a `Drinking orange juice` sequence in the OTI ESD-HMM results, and minor misclassification between members of the `Drinking orange juice`, `Leave home` and the `Walk from rear room to kitchen` classes in the left-right OTI ESD-HMM results were likewise caused by the interpolated state durations.

3.4.3 Abnormality Detection

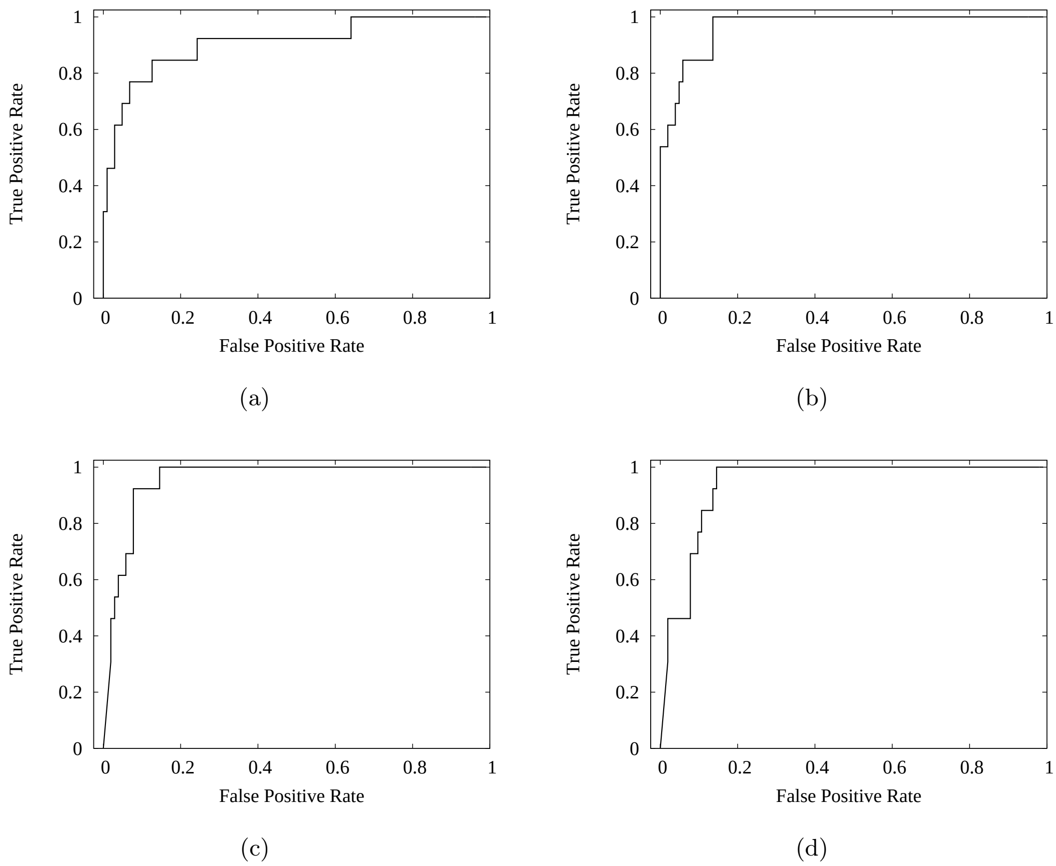
As in Section 3.2.3 and Section 3.3.3, thresholding will be used to measure the relative accuracy of the models in detecting anomalous behaviour. The highest log-likelihoods returned when querying each model on the probability of generating a sequence is used. The log-likelihoods are normalised by sequence length such that a global threshold may be applied. Thirteen sequences of irregular behaviour uncovered during the video segmentation phase are used and compared against the normal sequences. Leave-one-out cross validation is again employed to ensure that models are not queried on sequences used in training.

It should be noted that the anomaly in the sequences used lies only in their infrequency; none of the sequences are comprised of true abnormality that would justify the triggering of an alarm or intervention by an outside party. A summary of the abnormal sequences, the type of abnormality present and possible confusion with the known normal classes is given in Table 3.12.

The ROC curves in Figure 3.18 depict the relative performance of each of the models. Surprisingly, the left-right constrained HMM in Figure 3.18(b) shows the most promis-

Table 3.12: Summary of the real world abnormality sequences.

#	Description	Abnormality Type	Possible Confusion
1	Return to unit	Sequence	Leave home
2	Make breakfast	Sequence, duration	Drink Orange Juice
3	Eat breakfast	Sequence, duration	
4	Ponder life on sofa	Duration	Kitchen to rear room, rear room to kitchen
5	Open rear door	Sequence, duration	
6	Retrieve key from front door	Sequence, duration	Leave home
7	Aborts leaving home	Sequence, duration	Leave home
8	Clean kitchen	Duration	
9	Place jacket on sofa while walk through lounge	Sequence, duration	Kitchen to rear room, rear room to kitchen
10	Divert to hat stand while walk through lounge	Sequence, duration	Kitchen to rear room, rear room to kitchen, leave home
11	Place bag on sofa while walk through lounge (1)	Sequence, duration	Kitchen to rear room, rear room to kitchen
12	Place bag on sofa while walk through lounge (2)	Sequence, duration	Kitchen to rear room, rear room to kitchen
13	Pick up item by front door while walk through lounge	Sequence, duration	Kitchen to rear room, rear room to kitchen, leave home



(a) (b) (c) (d)

Figure 3.18: ROC curves showing the performance of (a) the HMM, (b) the left-right HMM, (c) the ESD-HMM and (d) the left-right ESD-HMM for the detection of abnormality among the real world sequences.

ing performance followed by the observed time indices ESD-HMM in Figure 3.18(c). The worse performance is seen in the standard HMM in Figure 3.18(a) and the left-right constrained OTI ESD-HMM in Figure 3.18(d). Of the two latter models, the HMM shows a slightly steeper initial curve compared to the slower rise of the left-right OTI ESD-HMM but is unable to find the complete set of abnormal sequences until an 80% false positive rate has been reached. The left-right OTI ESD-HMM, in contrast, attains a 100% abnormal classification rate at around the 18% false postive rate. Similar properties are shown in the curves for the left-right HMM and the observed time indices ESD-HMM. Here, also, the OTI ESD-HMM displays a slightly slower initial curve than the left-right HMM. Both models attain a 100% abnormal classification rate at around the 17% and 18% false positive mark.

The similar performance amongst the models can be clarified by plotting the log-likelihoods of each of the tested sequences. The log-likelihoods returned by the fully connected HMM in Figure 3.19(a) shows that most of the abnormal sequences return a likelihood well below those of the majority of normal sequences. Only the duration abnormal "ponder life on sofa" and "clean kitchen" sequences, and the sequence and duration abnormal "make breakfast," sequence appear amongst the majority of normal sequences with a high log-likelihood. The inability of the HMM to detect pure durational abnormality in real world conditions has been affirmed here. The high log-likelihood of the "make breakfast" sequence is due to a close similarity between this sequence and the `Drink Orange Juice` class; the repeated subsequences within the "make breakfast" sequence being permitted by the fully connected transition structure of the model.

The majority of classification error, however, relates to the low log-likelihoods returned by members of the `Leave home` class. Great variation exists among the sequences in this class due to noise introduced by rapid changes in lighting when the subject opens the front door. This results in less accurate trained models and difficulty in accurately identifying members of this class.

The log-likelihoods returned by the left-right constrained HMM is shown in Figure 3.19(b). Only the log-likelihoods of the two duration abnormal sequences remain within close proximity to the majority of normal sequences. The majority of classification error is again due to an inability to properly model the noisy members of the `Leave home` class.

The query results for the OTI ESD-HMM and the left-right OTI ESD-HMM are shown in Figure 3.19(c) and Figure 3.19(d) respectively. Here, the likelihoods of the duration abnormal "ponder life on sofa" and "clean kitchen" sequences have decreased such that they are – aside from the `Leave home` class – clearly differentiable from the normal sequences. It is interesting to note that while the probabilities of the abnormal sequences have, on average, decreased, so has the tolerance to noise; the likelihoods of the `Leave home` sequences have fallen in line with the abnormal sequences and several examples of the `Drink Orange Juice` and `Walk from rear room to kitchen` classes have moved down into abnormal regions of the charts.

Investigation reveals that several of the `Drink Orange Juice` and `Walk from rear room to kitchen` sequences possess highly unlikely state durations due to interpolation while others are hampered by noise and unexpected emissions that are can not be adequately accounted for without a state transition. The likelihood of a single `Walk`

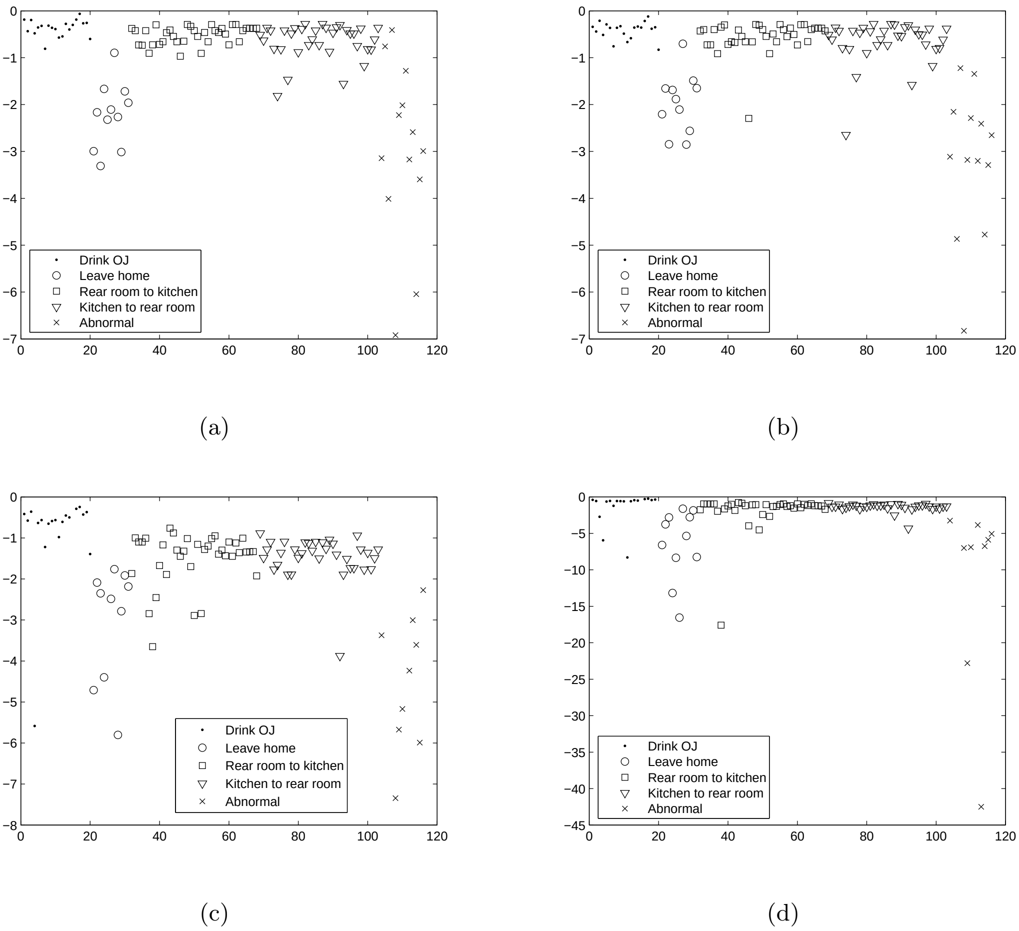


Figure 3.19: The normalised log-likelihoods of the normal and abnormal sequences for (a) the HMM, (b) the left-right HMM, (c) the ESD-HMM and (d) the left-right ESD-HMM for the detection of abnormality among the 116 real world sequences.

`from kitchen to rear room` sequence has also been greatly reduced due to an unlikely state duration.

3.4.4 Longer Term Duration Abnormality

The longer term duration abnormality experiment in Section 3.2.4 demonstrated that only the left-right constrained ESD-HMM was capable of detecting long term durational abnormality caused by a subject remaining at an otherwise normal activity for a significantly long period of time. This section revisits the ability of the models to cope with such anomaly by artificially adjusting the duration for the real world subject standing in the lounge hallway in a sequence from the `Walk from rear room to kitchen` class. The usual time spent in the middle of the room is typically two to four seconds.

The log-likelihood, again normalised by the length of the observation sequence, of the adjusted sequence was plotted for each of the HMM, left-right HMM, OTI ESD-HMM and left-right OTI ESD-HMM as the time spent standing in the hallway is incremented. The results for the HMM in Figure 3.20(a) and the left-right HMM in Figure 3.20(b) continue to show that the dynamic time warping nature of these models makes them unsuitable for finding this type of abnormality.

The results for the OTI ESD-HMM in Figure 3.20(c) and the left-right OTI ESD-HMM in Figure 3.20(d) show that these models are able to detect an anomaly within seconds when thresholding at the -3 log-likelihood. Unlike the ESD-HMM featured in Section 3.2.4, the observed time indices of the fully connected ESD-HMM make it impossible for the model to smooth the likelihood of the artificially adjusted sequence via state transitions. Both the OTI ESD-HMM and the left-right OTI ESD-HMM are therefore suitable for detecting this type of duration anomaly.

3.4.5 Conclusion

This section has used activity sequences gathered from a real world volunteer subject to investigate the use of the HMM and the OTI ESD-HMM as both sequence classifiers and as detectors of abnormality. The models showed only minor variations in performance when applied to sequence classification, due largely to a high presence of noise in two of the available classes.

Similar performance among the models was also discovered in an investigation into

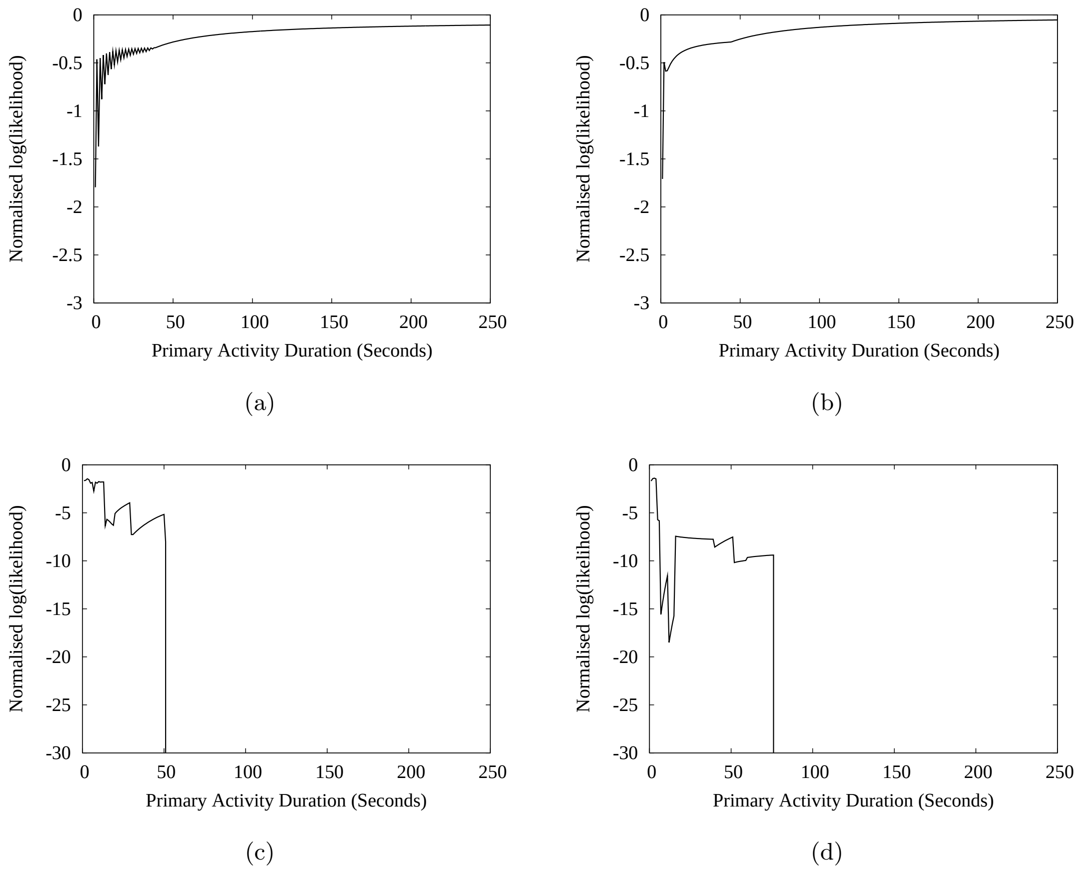


Figure 3.20: The normalised log-likelihood of (a) the HMM, (b) the left-right HMM, (c) the OTI ESD-HMM and (d) the left-right OTI ESD-HMM as the time spent standing in the lounge hallway in a sequence from the `Walk from rear room to kitchen` class is artificially adjusted.

the application of the models to the detection of abnormality through thresholding. Although the duration models showed some advantages over the HMM – affirming the benefit of incorporating duration into a model of human activity – the benefit gained was diminished by a decreased tolerance to noise.

Both the fully connected and left-right constrained OTI ESD-HMM performed well on the longer term duration abnormality task; the explicit state transition component ensured that the models were unable to smooth the likelihood of generating the artificially adjusted activity sequences through state transitions. It was shown that the HMM and left-right HMM were incapable of detecting this type of anomaly in a timely manner.

3.5 Conclusion

This chapter has explored the use of stochastic models as classifiers and detectors of abnormality in sequences of human activity using data gathered from a laboratory environment and from a real world volunteer subject.

Section 3.1 demonstrated that the HHMM is a viable model for the learning and recognition of human activity. The hierarchical nature of the model allows for model reuse in the training of higher level models that represent the inter-activity relationships of the lower level models. The lack of duration modelling in the standard HHMM, however, is of concern given the visual tracking system used.

Section 3.2 explored the use of the Explicit State Duration HMM (ESD-HMM) as a means of modelling human activity. It was shown that incorporating duration into models of human activity allowed durational abnormalities, defined as an otherwise normal activity being performed for an unusually short or unusually long period of time, to be detected. The HMM, in contrast, was shown to be unsuitable for detecting such anomalies due to its dynamic time warping property. It was demonstrated that the left-right constrained ESD-HMM was the only model capable of detecting longer term duration anomalies.

A modified ESD-HMM, in which the state transition times are known, was introduced in Section 3.3. The Observed Time Indices ESD-HMM (OTI ESD-HMM) was shown to perform on a par with the ESD-HMM and left-right ESD-HMM when applied to sequence classification. Marginal improvement was seen over these models when the OTI ESD-HMM was employed as a classifier of normal and abnormal sequences.

Finally, in Section 3.4, the performance of the left-right and fully connected HMM, and the left-right and fully connected OTI ESD-HMM were compared using two weeks of real world data gathered from the home of a volunteer subject. The model affirmed that benefit can be gained by explicitly including activity duration into an activity model. It was also shown that the OTI ESD-HMM was readily capable of detecting the presence of long term durational abnormality, a task to which the HMM is ill suited. It could not be shown, however, that the OTI ESD-HMM is a more suitable model for classification and abnormality detection on this data set due to difficulties encountered with noise. It is believed, however, that results would show the OTI ESD-HMM model to be more reliable than the HMM were a more robust tracking system to become available or were the experiment repeated in an environment with less hostile lighting conditions.

Chapter 4

Emergent Intertransaction Association Rule Mining: Discovering New and Changing Human Behaviours

Training and inferencing using graphical models such as those used in Chapter 3 is computationally expensive and is generally limited to applications in which it is reasonable to assume that human activity can be represented as repeatable sequences of asynchronous activities or events. The task of modelling human behaviour as precise sequences of events is made difficult by our tendency to interleave our activities and to adjust our behaviour when we are unexpectedly interrupted. In this chapter, a novel application of intertransaction association rule mining (Lu et al., 1998, 2000) for the detection and analysis of emergent human behaviour is proposed as a means of tackling this issue. Intertransaction Association Rules (IARs), formally introduced in Section 4.1, are implication rules that can be used to capture the associative, non-sequential, relationship of events observed within an intelligent environment while retaining some of the higher level temporal context in which these events occur. Traditional intratransaction associations retrieved during the IAR mining process function as a model of the events that are expected to occur in close temporal proximity while the intertransaction associations capture the larger context in which the intratransaction events occur. The temporal component of the rules also provides a predictive element that can be used to forecast the events we can expect to see in a future interval given events already observed.

Intelligent environments, however, generate many sensor events over short periods of time, resulting in dense data sets where the number of frequently occurring events, or items, can be numerous. This poses a problem for the current EH-Apriori Lu et al. (2000) and FITI Tung et al. (2003) algorithms for IAR mining as they rely on a computationally costly candidate-generation-then-test approach for rule discovery. This technique requires that k passes over a database are made to retrieve the set of frequent rules up to length k. Each pass over the data requires the generation of candidates – the set of all *possibly* frequent associations given those found in a previous pass. The scalability of such algorithms is hence limited due to the computational complexity of generating and testing the frequency of a combinatorial number of candidates. The number of candidates generated at each pass k of a worst case scenario given n database items and an intertransaction window of length w is $\sum_{r=0}^{k-1}\left[\binom{n}{k}\binom{nw}{r}\right]$. The computational complexity of testing the frequency of the candidates, many of which may be infrequent, can become intractable.

Addressing this issue, in Section 4.2 it is shown how pattern growth may be employed as an alternative to the Apriori based approach for the mining of intertransaction associations using the Extended FP-Tree (EFP-Tree), an adaptation of the Frequent Pattern Tree (FP-Tree) (Han et al., 2000, 2004) to the intertransaction association rule mining problem. Pattern growth is a more computationally efficient method for association rule mining that eliminates the need for candidate itemset generation by first transposing a transactional database into an intermediate form that aids subsequent mining. The original FP-Tree structure and mining algorithm, however, only finds frequent intratransaction association rules and is not suitable for intertransaction rule mining. Experimental results are presented that show an order of magnitude improvement in computational performance over the existing algorithms on synthetic dense data and on real world data captured in the homes of two volunteer subjects.

The second issue faced is how to gain insight into a person's behaviour so as to detect abnormality from an overwhelming number of rules that the mining process is likely to uncover. The use of emergent IARs as a novel means of finding patterns of behaviour that are of interest is proposed in Section 4.3.1. Emergent IARs are those rules that display significant growth from one data set to another and their presence may indicate abnormality evident as either a previously unseen pattern of events or unusually frequent occurrences of behaviour that would otherwise be considered normal. Emergent IARs offer a convenient means of identifying changes that would otherwise be difficult to discern through manual inspection of the rule sets. For example, the real world event logs from a single week of data can produce around 7,200 patterns yet these can be distilled down to approximately 150 emergent rules that are likely to be of interest to us. The same data set mined with a slightly higher support threshold will retrieve circa 2,700 patterns of which 46 are emergent.

4.1 Intertransaction Association Rules

Consider the set of all items $I = \left\{a^1 a^2 \ldots a^i \ldots a^M\right\}$ occurring in a database $DB = \langle T_1 T_2 \ldots T_N \rangle$ of transactions T_i $(1 \leq i \leq N)$ such that $T_i\,(x) \in I$ $\forall$ items x in T_i. At any transaction T_i the items are said to form the set $S_{T_i} = \left\{a^i_{T_i} \ldots a^k_{T_i}\right\}$. For the case of a single intertransaction dimension attribute, an intertransaction sliding window of size w transactions is passed over the transactions in DB to extract the extended transaction items such that the extended transaction at T_i is $E_{T_i} = \{S_{T_i}, S_{T_i+1} \ldots S_{T_i+w}\}$ and the set of all possible extended transaction items is $E = \left\{a^1_0 a^2_0 \ldots a^i_d \ldots a^M_w\right\}$. The mining problem reduces to the traditional intratransaction case when $w = 0$, that is, when only intratransaction items are included in an extended transaction itemset.

The superscript notation is dropped when the value of an item is known. For example, the extended transaction items retrieved with a sliding intertransaction window of size $w = 5$ starting at transaction ID 300 from the example database in Table 4.1 are C_0, B_0, A_0, C_2, E_2, B_3, E_3, B_4, and A_4, given that the dimensional attribute is the transaction time.

Intertransaction association rules are implication rules such that $X \Rightarrow Y$ with the following properties (Tung et al., 2003):

$$X \subseteq E, Y \subseteq E \tag{4.1}$$

$$\exists a^i_0 \in X \tag{4.2}$$

$$\exists a^i_d \in Y, d > 0 \tag{4.3}$$

$$X \cap Y = \emptyset \tag{4.4}$$

The support and confidence measures of an itemset are calculated as $\frac{|T_{xy}|}{N}$ and $\frac{|T_{xy}|}{|T_x|}$ respectively where $|T_{xy}|$ is the number of extended transactions containing all items in $X \cup Y$, $|T_x|$ is the number of extended transactions containing all items in X and N is the number of extended transactions.

In this work the extended transaction items $\left\{a^1_0 \ldots a^M_0\right\}$ are referred to as intraitems and the extended transaction items $\left\{a_1^1 \dots a_M^w\right\}$ are referred to as interitems.

4.2 Extended Frequent Pattern Tree

The proposed Extended Frequent Pattern Tree (EFP-Tree) is a tree structure of descending frequency ordered intraitem nodes with zero or one interitem Frequent Pattern Tree (FP-Tree) subtrees. The frequency ordering of the interitem FP-Tree is conditioned on the intratransaction item parent. Each node in the tree contains an item ID which maps to a codebook of item descriptors, a frequency counter, a link to its parent node, links to zero or more children and a link to the next node in the tree of the same item ID. Interitem nodes also carry the dimensional offset of the item to ensure that the intertransaction relationship of the item relative to its intraitem parent is maintained.

Nodes are placed into the tree such that the entire set of frequent items for an arbitrary intertransaction can be restored by traversing the tree. The ordering of nodes into descending frequency increases the likelihood of items placed into the tree sharing common nodes, creating a compact representation of the database transactions that captures the associative relationship of the transaction items.

4.2.1 Tree Construction

Three passes, detailed in Algorithm 4.1, over a database are required to build the tree structure. As in the FP-Tree, the frequency of single items is gathered in an initial pass over the database to build the set of frequent single intraitems, or 1-itemsets, and the set of frequent interitems given a minimum support threshold. The intraitems are ordered by descending frequency to become the item lookup header table for the intraitem tree. The frequent items are those items whose frequency count is greater than or equal to the minimum support threshold α.

The intraitem FP-Tree is built and the conditional frequencies of the interitems are found in the second pass. The intraitems for each transaction T_i are first filtered to remove items not present in the known frequent intraitem set from the first pass and are sorted in order of descending frequency. The ordered item list is recursively inserted into the tree such that at each level l in the tree the child node with the ID of the l^{th} item in the ordered array is traversed and its frequency count is incremented. Children nodes that do not exist will be created prior to traversal and kept in codebook ID order so that binary search can be used when traversing the tree. The linked list of nodes of

Algorithm 4.1: EFP-Tree Construction

Input: Transaction Database DB, sliding intertransaction window size *maxSpan*, tree building support threshold α
Output: EFP-Tree *tree*

Method:

```
// First Pass
```
Count the frequency of single items in DB to build the set of frequent items F such that support $(F_j) \geq \alpha \ \ (1 \leq j \leq |F|)$;

```
// Second Pass
```
Create the intratransaction root node *tree*;
For each intratransaction $T_i \in DB$ **do**
  $A \leftarrow$ intratransaction items such that $A_j \in T_i, A_j \in F \ \ (1 \leq j \leq |A|)$ ordered by descending frequency;
  $I \leftarrow$ count of intertransaction items $|T_i$, *maxSpan*;
  recursively insert the nodes A onto *tree* finishing at node *interParent*;
  interParent.interFreq $\leftarrow$ *interParent.interFreq* $+ I$;
end

```
// Third Pass
```
For each intratransaction $T_i \in DB$ **do**
  $A \leftarrow$ intratransaction items such that $A_j \in T_i, A_j \in F \ \ (1 \leq j \leq |A|)$ ordered by descending frequency;
  interParent $\leftarrow$ intraitem node in *tree* corresponding to A;
  $E \leftarrow$ intertransaction items such that $E_j \in T_i, E_j \in F \ \ (1 \leq j \leq |E|)$ ordered by descending frequency | *leaf.interFreq*;
  create the root node *interParent.interTree*;
  recursively insert the nodes E onto *interParent.interTree*;
end
return *tree*

same item ID that originates from the root node header table is updated whenever a new node is created. The frequency of the interitems relative to T_i are incremented in the final intraitem node that is traversed. The ordered lists of frequent intraitems are stored for use in the third pass. The root node is said to be at level $l = 0$.

The third and final pass over the database builds the interitem sub-trees in the EFP-Tree structure. At each transaction T_i the cached ordered list of frequent intraitems are used to traverse the intraitem tree and locate the intraitem node that will become the root node of the interitem subtree. The extended items within the intertransaction sliding window at T_i are filtered to remove known infrequent interitems and the remaining items are sorted in order of local descending frequency given the intraitem parent. The ordered interitems are then recursively inserted into the interitem subtree as before.

The example database in Table 4.1 and Table 4.2 is used to demonstrate the construction of the EFP-Tree structure in Figure 4.1 using a minimum support threshold of $\alpha = 3$ and a sliding intertransaction window size of $w = 5$.

Table 4.1: Example database with the unsorted and descending frequency ordered items. The time at which each transaction occurs is shown.

Trans. ID	Time	Raw Items	Ordered Items
100	1	A C B	B C A
200	2	B	B
300	3	C A B	B C A
400	5	E C	C E
500	6	B E	B E
600	7	A B	B A
700	9	C	C
800	10	C D B	B C D
900	11	C B A	B C A

Table 4.2: Extended transactions retrieved from the example database in Table 4.1 using a sliding intertransaction window of size $w = 5$.

Time	Extended transaction items
1	B_0 C_0 A_0 B_1 B_2 C_2 A_2 C_4 E_4 B_5 E_5
2	B_0 B_1 C_1 A_1 C_3 E_3 B_4 E_4 B_5 A_5
3	B_0 C_0 A_0 C_2 E_2 B_3 E_3 B_4 A_4
5	C_0 E_0 B_1 E_1 B_2 A_2 C_4 B_5 C_5 D_5
6	B_0 E_0 B_1 A_1 C_3 B_4 C_4 D_4 B_5 C_5 A_5
7	B_0 A_0 C_2 B_3 C_3 D_3 B_4 C_4 A_4
9	C_0 B_1 C_1 D_1 B_2 C_2 A_2
10	B_0 C_0 D_0 B_1 C_1 A_1
11	B_0 C_0 A_0

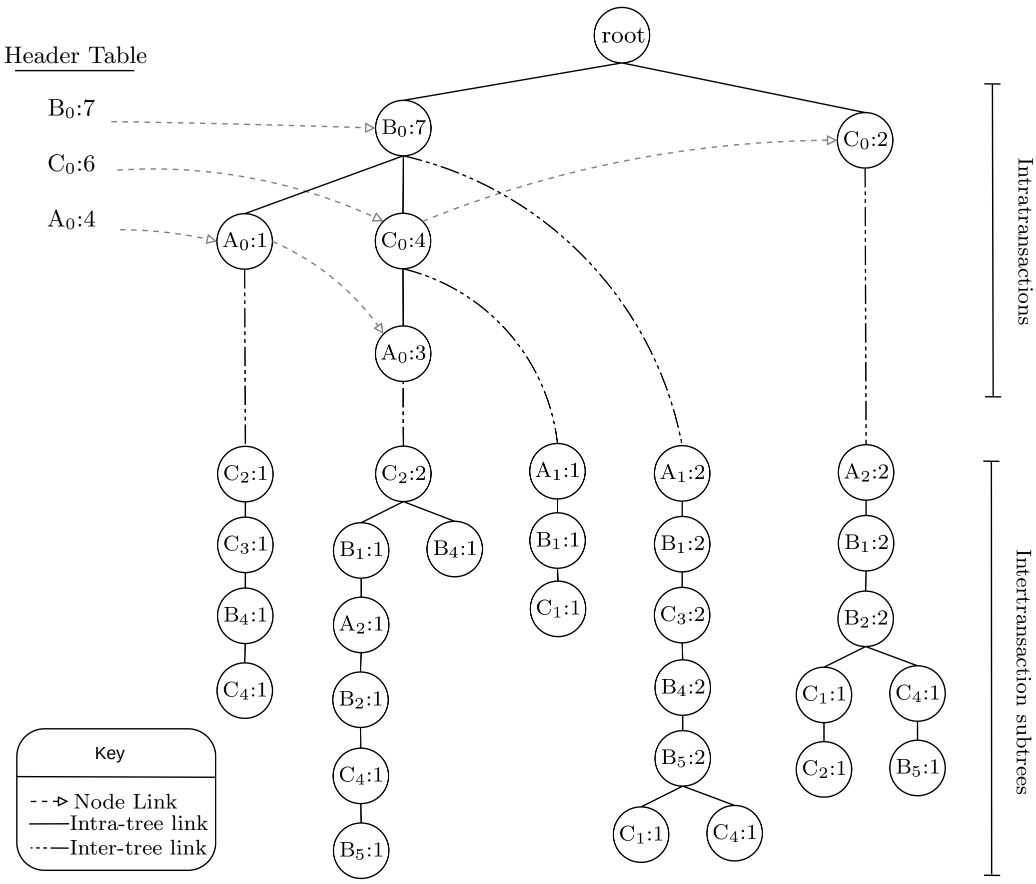


Figure 4.1: The Extended FP-Tree for the example database with $w = 5$ and $\alpha = 3$. Subscript numbers represent the dimensional offset, in this case the item time, relative to the intratransaction items while colon delineated numbers depict the node frequency. The header tables of the intertransaction item subtrees have been omitted for clarity.

The frequent items are found in the first pass over the database from Table 4.2. The frequent intraitems found given the minimum support threshold $\alpha = 3$ are B_0:7, C_0:6 and A_0:4 with frequency counts of 7, 6 and 4 respectively. The frequent interitems meeting the same minimum support threshold are A_1:3, A_2:3, B_1:6, B_2:3, B_4:4, B_5:4, C_1:3, C_2:4, C_3:3 and C_4:4.

In the second pass, the intraitems B_0, C_0 and A_0 are added to the root node of an empty tree such that all nodes are recursively created to produce the FP-Tree in Figure 4.2(a). The frequency of item B_0 is incremented by the next transaction. The items B_0, C_0 and A_0 are added again, the existing nodes are traversed and their frequency counts are each incremented. The next two transactions see the node C_0 created as the second child of the root node and an increment to the count of B_0. The FP-Tree up to this point is in Figure 4.2(b). Figure 4.2(c) shows the state of the tree when B_0 and A_0 are added such that A_0 becomes the second child of B_0 and the count of B_0 is incremented once more. The count of the nodes representing the intratransaction associations C_0 and B_0, C_0 are incremented by the next two transactions. Finally, the counts of the items in the path B_0, C_0, A_0 are once again incremented. The final intraitem FP-Tree is presented in Figure 4.2(d).

The third and final pass over the example database now begins. The known frequent items found in the first pass allow the extended transaction items at time 1 to be reduced to B_0, C_0, A_0, B_1, A_2, B_2, C_2, C_4 and B_5 given that the items E_4 and E_5 are known to not meet the minimum support threshold. The interitems will be inserted at the node identified by following the path B_0, C_0, A_0 through the tree in Figure 4.2(d). The insertion of the ordered list of interitems from the first extended transaction will create the children interitem nodes C_2, B_1, A_2, B_2, C_4 and B_5 as shown in Figure 4.3. Next, the extended items at time 2 are reduced to A_1, B_1, C_1, C_3, B_4 and B_5. The items are ordered and inserted as new children of B_0 to produce the EFP-Tree in Figure 4.4. The nodes B_0, C_0 and A_0 will again be traversed at time 3. The child node C_2 will then be incremented and a new node for item B_4 will be inserted as a child of C_2 to create the EFP-Tree in Figure 4.5.

The interitems from Table 4.2 at time 5 are reduced to B_1, A_2, B_2, C_4 and B_5 and inserted as new children of C_0 to produce the tree shown in Figure 4.6. Next, the interitems at time 6 are filtered and inserted as children of B_0 such that the counts of A_1, B_1, C_3, B_4 and B_5 are each incremented and the node C_4 is inserted as a new child of B_5. The resulting EFP-Tree is given in Figure 4.7. At time 7 the intraitems B_0 and A_0 are traversed and the interitems C_2, C_3, B_4 and C_4 are inserted as children of A_0 to create the tree shown in Figure 4.8. The next transaction at time 9 sees the count

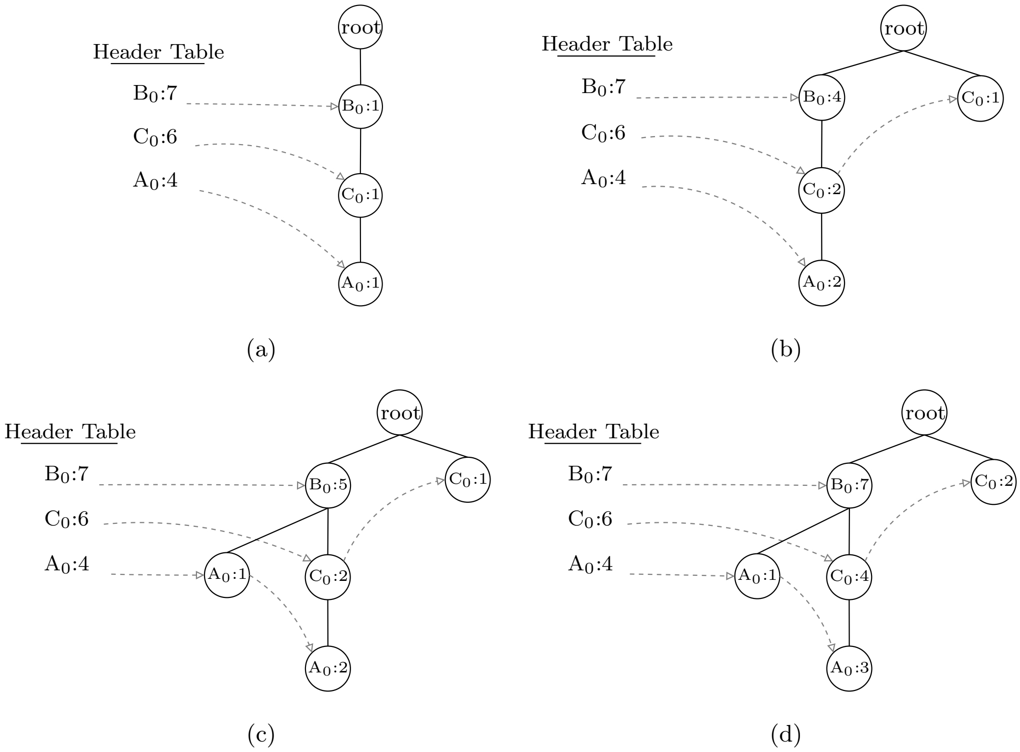


Figure 4.2: The intraitem FP-Tree at various stages of construction after the intraitems from the extended transactions in Table 4.2 at (a) time 1 are added, (b) the tree after time 6, (c) after time 7 and, finally, (d) after time 11.

of the nodes A_2, B_1 and B_2, as children of C_0, incremented and the interitems C_1 and C_2 inserted as children of B_2. The resulting EFP-Tree is shown in Figure 4.9. The final tree, in Figure 4.1, is obtained by traversing B_0, C_0 and appending the frequent interitems from the extended transaction at time 10.

No interitems exist in the extended items in Table 4.2 at time 11 so no further action is required.

4.2.2 Data Mining

A method to extract the intertransaction associations present in the EFP-Tree structure and to generate the IARs is now required. As in the FP-Tree, retrieval of association rules from the EFP-Tree is made possible by the pattern growth property (Han et al., 2000, 2004). Pattern growth uses a divide and conquer approach that recursively builds the entire set of frequent associations by constructing trees conditioned on known fre-

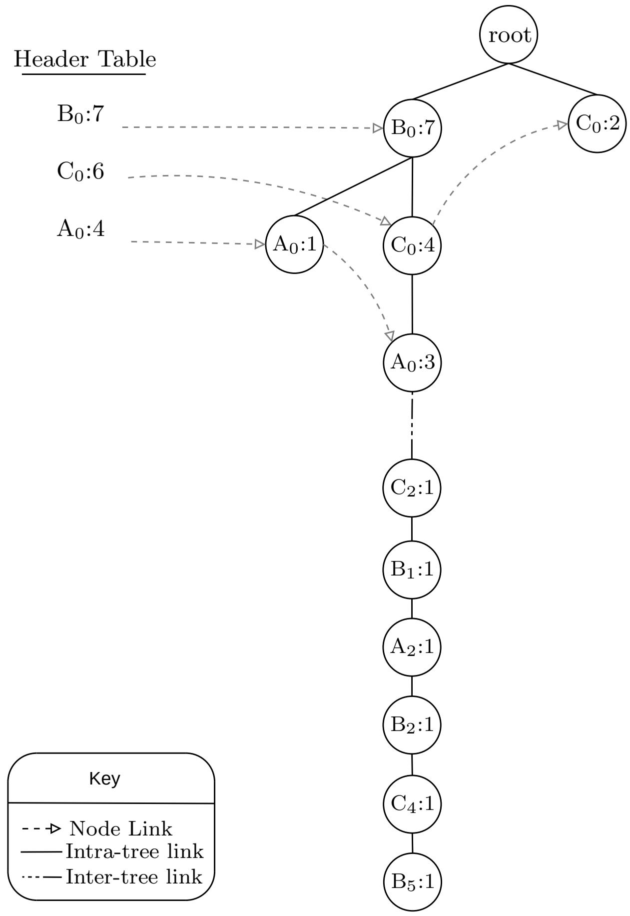


Figure 4.3: EFP-Tree construction as the frequent interitems from Table 4.2 at time 1 are inserted. The header tables of the intertransaction item subtrees have been omitted for clarity.

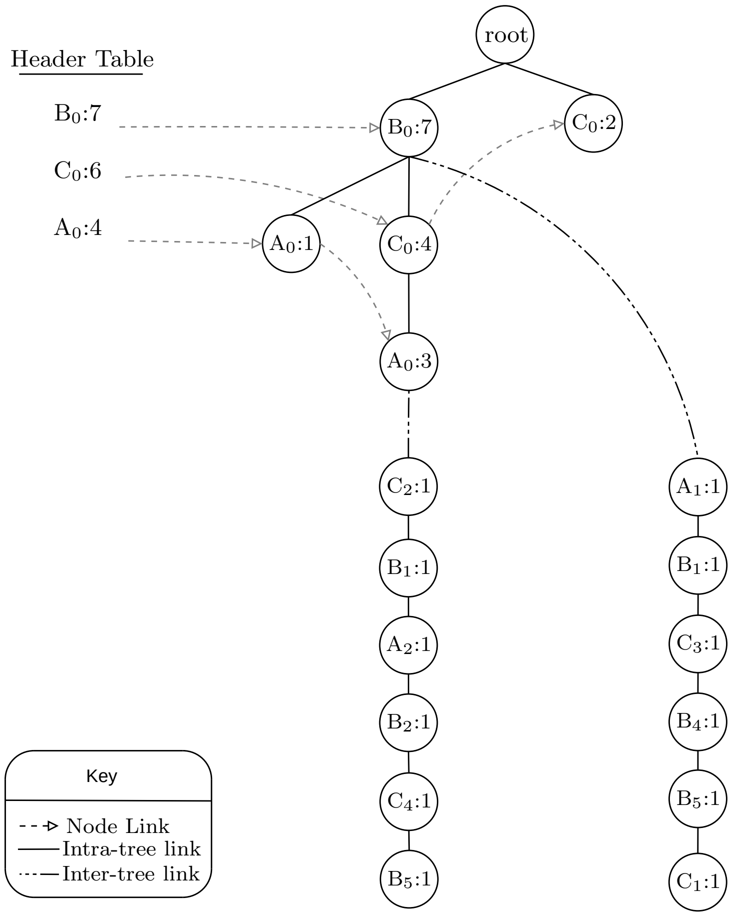


Figure 4.4: EFP-Tree construction as the frequent interitems from Table 4.2 at time 2 are inserted into the EFP-Tree in Figure 4.3. The header tables of the intertransaction item subtrees have been omitted for clarity.

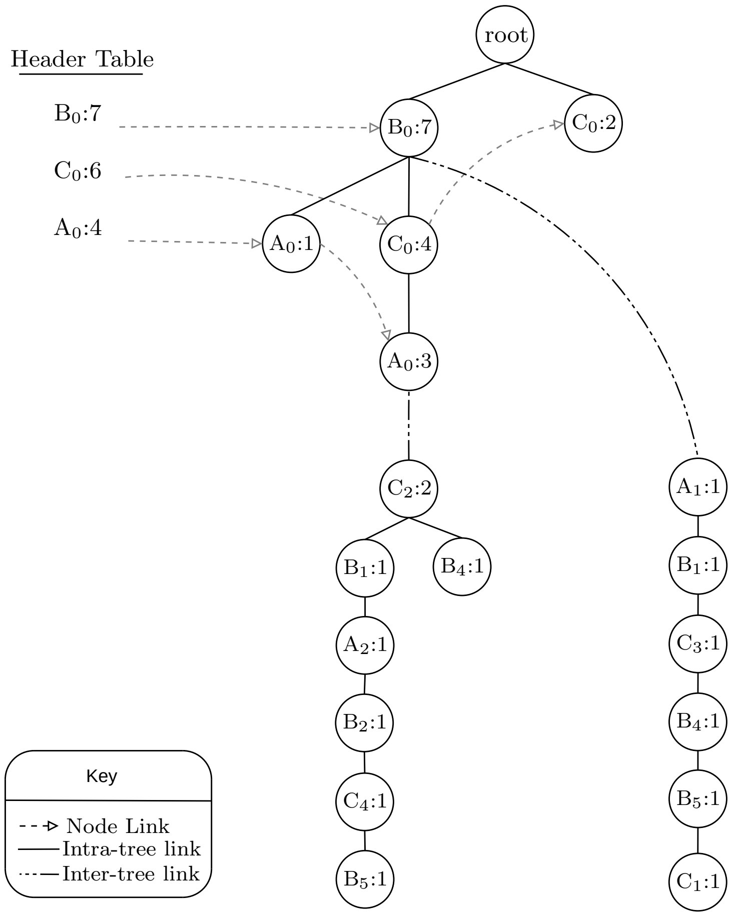


Figure 4.5: EFP-Tree construction as the frequent interitems from Table 4.2 at time 3 are inserted into the EFP-Tree in Figure 4.4. The header tables of the intertransaction item subtrees have been omitted for clarity.

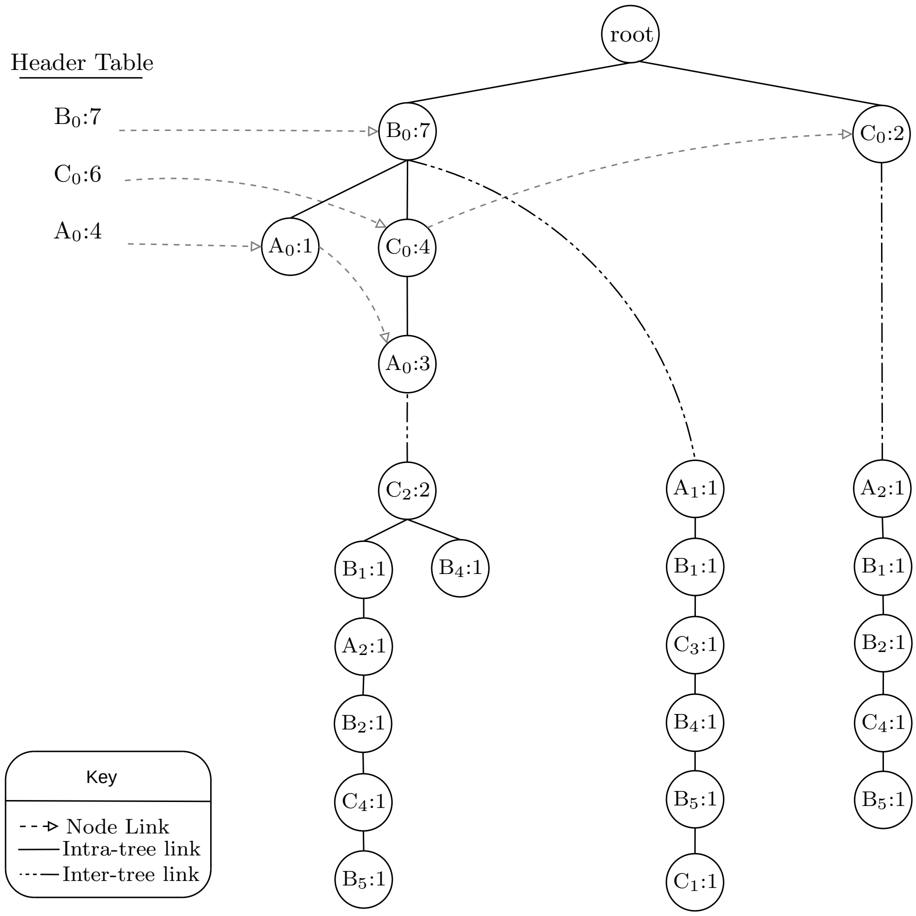


Figure 4.6: EFP-Tree construction as the frequent interitems from Table 4.2 at time 5 are inserted into the EFP-Tree in Figure 4.5. The header tables of the intertransaction item subtrees have been omitted for clarity.

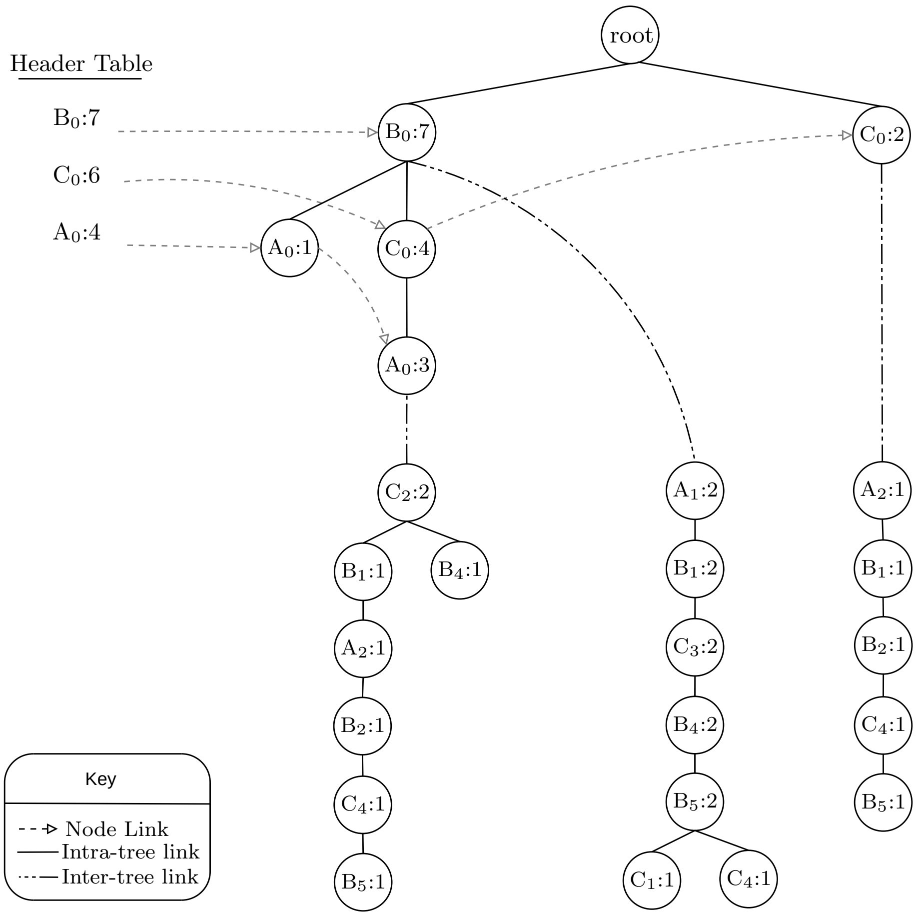


Figure 4.7: EFP-Tree construction as the frequent interitems from Table 4.2 at time 6 are inserted into the EFP-Tree in Figure 4.6. The header tables of the intertransaction item subtrees have been omitted for clarity.

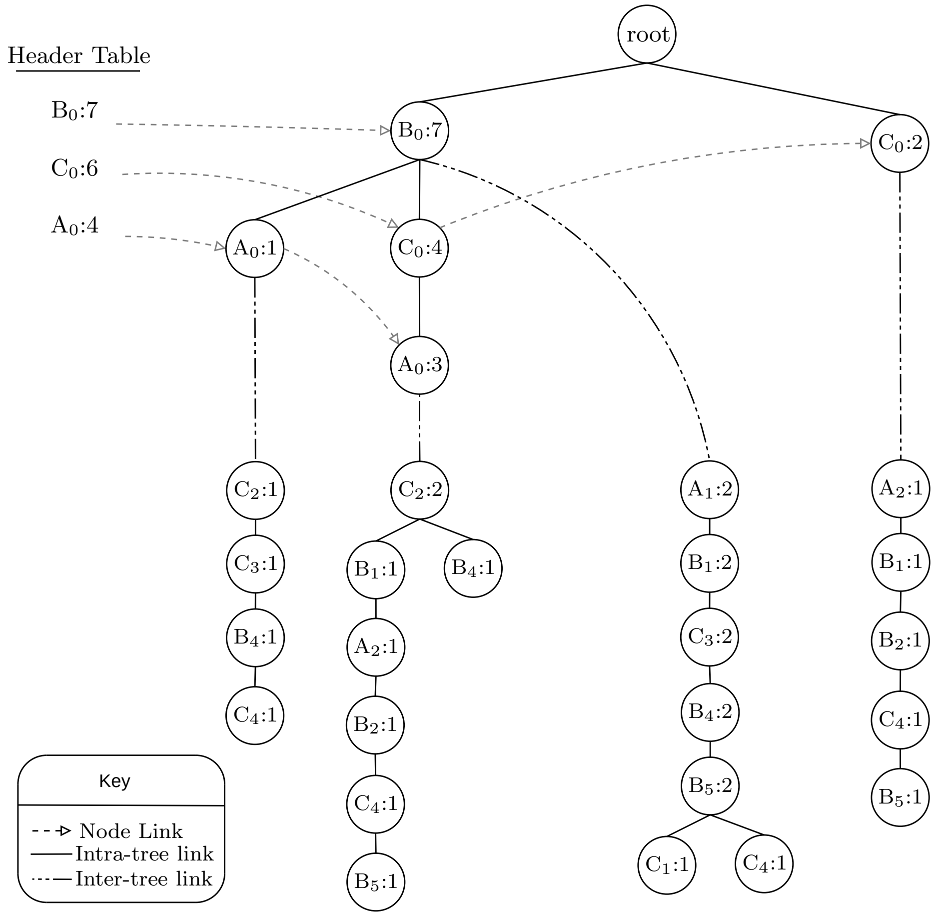


Figure 4.8: EFP-Tree construction as the frequent interitems from Table 4.2 at time 7 are inserted into the EFP-Tree in Figure 4.7. The header tables of the intertransaction item subtrees have been omitted for clarity.

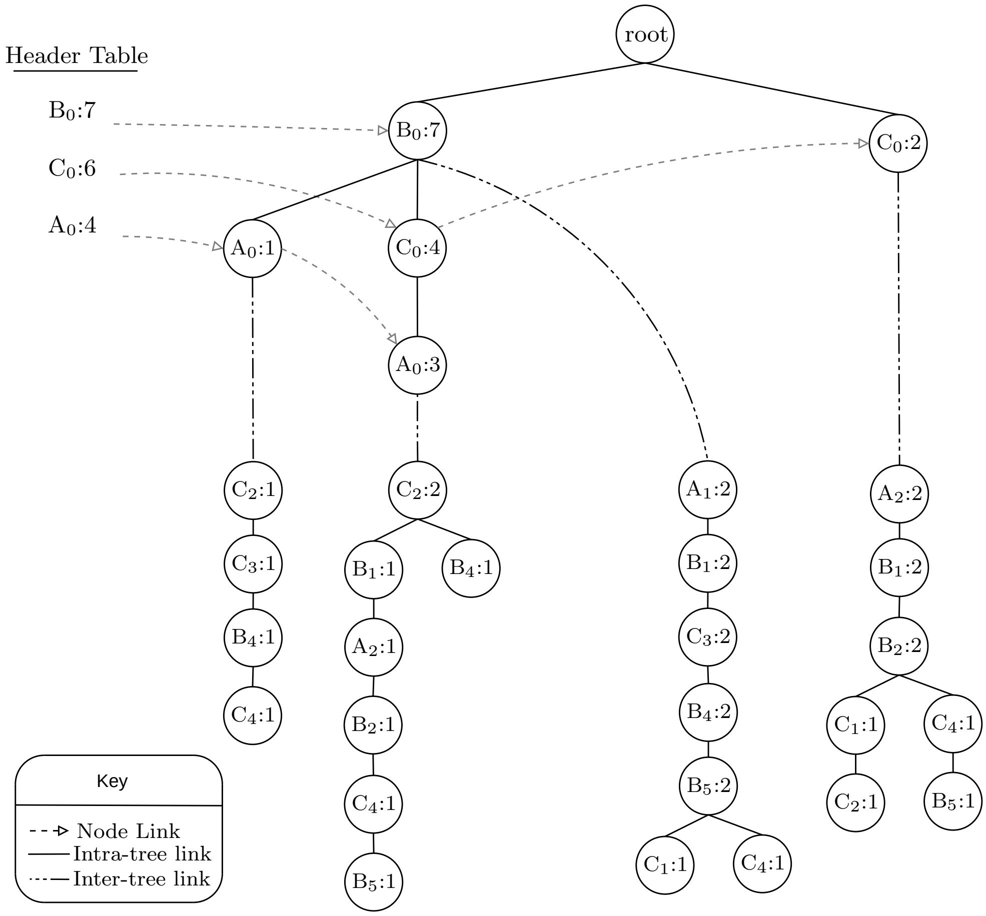


Figure 4.9: EFP-Tree construction as the frequent interitems from Table 4.2 at time 9 are inserted into the EFP-Tree in Figure 4.8. The header tables of the intertransaction item subtrees have been omitted for clarity.

quent base rules and taking the dot product of the frequent items in the conditional tree and the conditional base itemset to produce new rules. These new rules then become the conditional base for the next set of conditional trees to be mined.

The Frequent Pattern Growth (FP-Growth) algorithm from the FP-Tree differs to the Extended FP-Growth (EFP-Growth) algorithm used to mine the EFP-Tree in that the latter must consider intertransaction relationship inheritance along the intraitem nodes.

Property 4.2.1. (Intertransaction inheritance property) Intratransaction item nodes inherit the intertransaction item relationships of their intratransaction item children.

Interitems are inserted into a subtree whose root node is the last intraitem node traversed to when an extended transaction itemset is sorted into descending frequency order and placed into the EFP-Tree. As a given interitem subtree can only be reached by traversing the intraitem tree in the presence of all parent intraitem nodes it follows that the relationship between an intraitem node and the items in the subtree must apply to all nodes traversed to reach the intertransaction item subtree.

Example 4.2.1. The extended transaction $L = \mathrm{A}_0, \mathrm{B}_0, \mathrm{C}_0, \mathrm{B}_1, \mathrm{C}_3$ is inserted as an ordered item list into an empty tree. The item nodes A_0, B_0 and C_0 are created as a single branch in the intraitem tree and the items B_1 and C_3 are in turn inserted as interitem nodes as children of C_0. Given Property 4.2.1 we can infer that there exists a relation $\mathrm{A}_0 \Rightarrow B_1$ which is known to exist as both $\mathrm{A}_0 \in L$ and $\mathrm{B}_1 \in L$. $\square$

The EFP-Growth algorithm, detailed in Algorithm 4.2, will now be described. Starting with an EFP-Tree T and an empty conditional base, or rule suffix, EFP-Growth iterates over the set of intraitems I in T to build a conditional tree T_c conditioned on I for each frequent I. At each recursion, I is prepended to the conditional base to generate, or grow, a new association rule and to build the conditional tree for the next recursive step. No candidate generation is necessary as the frequency of the items is stored in the tree structure and all generated rules are guaranteed to be frequent.

Two types of conditional tree are used in EFP-Growth, a conditional EFP-Tree T_c used for finding the related intraitems and interitems that can be used to extend the present intraitem rule suffix and a FP-Tree T_e of the interitems inherited by the conditional base. This latter tree is used to find the interitem associations for a given intraitem rule suffix and is required as not all interitems inherited by the conditional base may be included in T_c.

Algorithm 4.2: EFP-Growth

Input: EFP-Tree N, mining support threshold α, sliding intertransaction window size *maxSpan*
Output: Set of frequent rules
Method:

minedRules $\leftarrow \emptyset$;
For each item a_i in header table of N from least to most frequent such that support $(a_i) \geq \alpha$ **do**
  Find the conditional prefix path and the extended items for a_i, propagate the intertransaction items of each occurrence of a_i to its parent and build the conditional EFP-Tree T_c;
  If T_c contains a single intratransaction path P such that no non-leaf node contains an intertransaction subtree **then**
    $T_c \leftarrow T_c$ with P removed;
    singlePathRules $\leftarrow$ all combinations of intratransaction nodes in P;
    singlePathRules $\leftarrow$ *singlePathRules* $\times$ rules returned by call to FP-Growth (leaf node of P, α) as in Algorithm 2.3;
  end
  returnedRules $\leftarrow$ call EFP-Growth $(T_c, \alpha, maxSpan)$;
  Build the intertransaction item FP-Tree T_e using the extended items from a_i;
  interRules $\leftarrow$ call FP-Growth (T_e, α) as in Algorithm 2.3;
  ruleSet $\leftarrow$ *returnedRules* $\cup$ *interRules* $\cup$ *singlePathRules*;
  For each rule $R \in$ *ruleSet* **do**
    add a_i to R with support $(R) = \min(\text{support}(R), \text{support}(a_i))$;
  end
  add a_i to *ruleSet* with support = frequency of a_i in N;
  minedRules $\leftarrow$ *minedRules* $\cup$ *ruleSet*;
end
Return *minedRules*

Given a tree T, the conditional tree T_c conditioned on some I is found by collecting the set of extended transactions formed through the union of the prefix path and the inherited interitems for each node in T whose item ID is I and whose immediate parent is an intraitem node. The prefix path for any given node is the set of its parent nodes and corresponding frequencies as stored in the EFP-Tree. All nodes in T of item ID I are found by following the linked list of "same item" ID nodes, the head of which is stored in the intraitem header table of T. The extended transactions are then used to build T_c as described in Section 4.2.1.

The conditional interitem tree T_e for a given conditional base is found by constructing an FP-Tree of the interitem transactions inherited by the conditional base rule and using FP-Growth to mine the resulting tree. Taking the dot product of the conditional base and the set of interitem associations returned by FP-Growth produces the entire set of intertransaction associations related to the conditional base.

This process continues recursively until no more conditional trees are built or until only a single intraitem path exists in T_c. If T_c contains a single intraitem path we can avoid recursion and find the complete set of rules given the conditional base by finding the

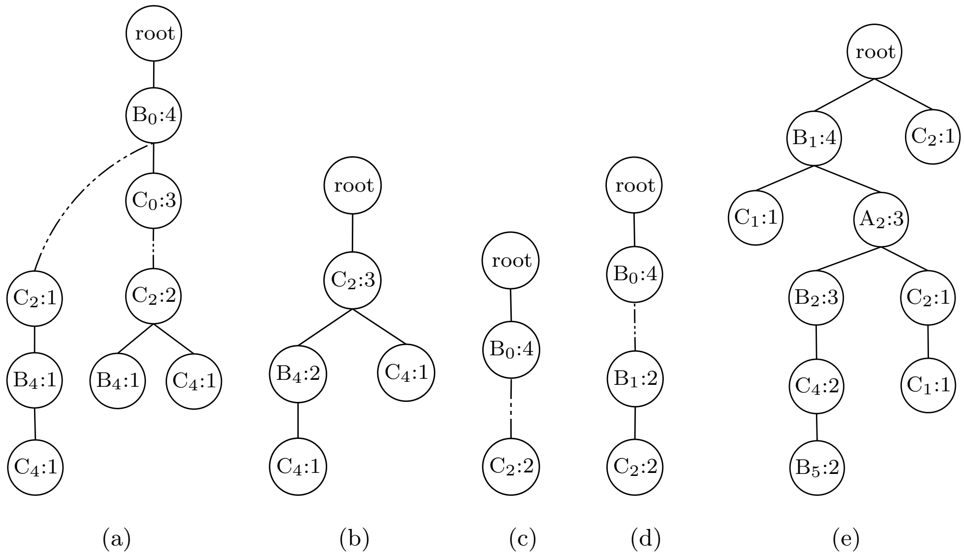


Figure 4.10: The conditional EFP-Tree tree T_c and conditional interitem FP-Tree T_e for (a) $T_c|\mathrm{A}_0$, (b) $T_e|\mathrm{A}_0$ and $T_e|\mathrm{B}_0\mathrm{A}_0$, (c) $T_c|\mathrm{C}_0\mathrm{A}_0$, (d) $T_c|\mathrm{C}_0$ and (e) $T_e|\mathrm{C}_0$ when the EFP-Tree in Figure 4.1 is mined with a minimum support threshold of $\alpha = 2$.

dot product of the intraitem combinations in T_c and the interitem associations returned when calling FP-Growth on the interitem subtree.

Although the example EFP-Tree T in Figure 4.1 was built with a minimum support level of $\alpha = 3$, the support threshold will be set to $\alpha = 2$ for mining the tree in order to demonstrate the mining process in finer detail than is possible at the original support setting.

Conditioning T on A_0, we find the conditional prefix paths $\langle \mathrm{B}_0{:}1 \rangle$ and $\langle \mathrm{B}_0{:}3\ \mathrm{C}_0{:}3 \rangle$. The intertransaction items $\langle \mathrm{C}_2{:}1\ \mathrm{C}_3{:}1\ \mathrm{B}_4{:}1\ \mathrm{C}_4{:}1 \rangle$ are related to $\langle \mathrm{B}_0{:}1 \rangle$ and the interitems $\langle \mathrm{C}_2{:}1\ \mathrm{B}_1{:}1\ \mathrm{A}_2{:}1\ \mathrm{B}_2{:}1\ \mathrm{C}_4{:}1\ \mathrm{C}_5{:}1 \rangle$ and $\langle \mathrm{C}_2{:}1\ \mathrm{B}_4{:}1 \rangle$ are found for $\langle \mathrm{B}_0{:}3\ \mathrm{C}_0{:}3 \rangle$. The conditional tree $T_c|\mathrm{A}_0$ with $\alpha = 2$ is shown in Figure 4.10(a) and the interitem FP-Tree $T_e|\mathrm{A}_0$ in Figure 4.10(b).

The set of intertransaction rules associated with the conditional base A_0 is found by taking the dot product of A_0 and the interitem associations returned by FP-Growth from the tree in Figure 4.10(b). The resulting rules are $\mathrm{A}_0 \Rightarrow \mathrm{C}_2{:}3$, $\mathrm{A}_0 \Rightarrow \mathrm{B}_4{:}2$, $\mathrm{A}_0 \Rightarrow \mathrm{C}_4{:}2$, $\mathrm{A}_0\mathrm{C}_2 \Rightarrow \mathrm{B}_4{:}2$ and $\mathrm{A}_0\mathrm{C}_2 \Rightarrow \mathrm{C}_4{:}2$.

Recursing into the conditional tree in Figure 4.10(a), EFP-Growth grows the conditional base by finding the least frequent intraitem whose support meets the support threshold for mining. This item, C_0, is prepended to A_0 to create the new conditional base C_0A_0 and the intratransaction rule $C_0 \Rightarrow A_0{:}3$. The single interitem prefix path $\langle B_0{:}4\rangle$ and its inherited interitems $\langle C_2{:}1\ B_4{:}1\rangle$ and $\langle C_2{:}1\ C_4{:}1\rangle$ form to create the single intraitem path conditional tree shown in Figure 4.10(c). The inherited items of C_0A_0 are used to build a single node FP-Tree T_e containing $C_2{:}2$ resulting in the generation of a single rule $C_0A_0 \Rightarrow C_2{:}2$ when the dot product of the interitem associations found in $T_e|C_0A_0$ and the conditional base is found. Recursively mining the tree in Figure 4.10(c) generates the rules $B_0C_0 \Rightarrow A_0{:}2$ and $B_0C_0A_0 \Rightarrow C_2{:}2$.

The mining now returns to $T_c|A_0$ to create the next conditional base B_0A_0 and generate its respective rule $B_0 \Rightarrow A_0{:}4$. No prefix path of B_0A_0 exists in Figure 4.10(a) so no conditional tree $T_c|B_0A_0$ needs to be built. The mining of the interitem FP-Tree $T_e|B_0A_0$, the same as for $T_e|A_0$ in Figure 4.10(b), generates the rules $B_0A_0 \Rightarrow C_2{:}3$, $B_0A_0 \Rightarrow B_4{:}2$, $B_0A_0 \Rightarrow C_4{:}2$, $B_0A_0C_2 \Rightarrow B_4{:}2$ and $B_0A_0C_2 \Rightarrow C_4{:}2$.

Upon return from a recursive call EFP-Growth will update the immediate parent of each node whose item ID is I such that the interitems are inherited and ready for conditioning on the next frequent item. It is for this reason that the mining algorithm grows rules by recursing into trees conditioned on the least frequent intraitems first.

Returning to the original tree T in Figure 4.1 the recursive mining technique will be applied in turn to the conditional bases C_0 and B_0. The conditional trees $T_c|C_0$ and $T_e|C_0$ are given in Figure 4.10(d) and Figure 4.10(e) respectively.

IARs generated through EFP-Growth may be required to adhere to constraints present in the application domain. For example, the association $B_0C_4 \Rightarrow C_2{:}1$ makes little sense when the intertransaction attribute of the items is temporal and the rules are to be applied to prediction. In this case we can reorder the items to produce the association $B_0C_2 \Rightarrow C_4{:}1$ without affecting the accuracy of the support measure. The confidence measure of a reordered rule can be calculated by returning extra information during the recursive mining step.

4.2.3 Benchmark Comparisons

Both synthetic data, employed to model the best and worst case scenarios for association rule mining, and real world data sets, to indicate the practical application of the mining algorithms, were used to compare the computational performance and peak memory requirements of EFP-Growth with FITI. These performance measures are important as they empirically demonstrate the scalability of the algorithms on input data of varying characteristics. For each data set the ability of the algorithms to scale with respect to the length of the intertransaction window and a decreasing minimum support threshold is observed. Tung et al. (2003) have previously shown FITI to be computationally more efficient than EH-Apriori and so the latter algorithm is not considered in this experiment.

The real world data used (Tapia et al., 2004) are event logs from an array of state-change sensors installed in the homes of two volunteer subjects, a thirty year old working professional and an eighty year old retiree, over a period of sixteen days. The sensors, 77 in the first subject's home and 84 in the second, were fitted to a variety of appliances, containers and furniture to log the times of use. These events were discretised for mining into transactions of five minute intervals to produce 658 transactions for the first subject and 748 transactions for the second. Unique sensor IDs were stripped from the event logs to reduce the sensor information to only include the sensor state and its room and object context. For example, multiple sensors installed on the doors of a cabinet are reduced to `Kitchen/Cabinet true` and `Kitchen/Cabinet false` events. The event codebooks contained 76 and 80 entries for the first and second subjects respectively.

Two synthetic data sets representing sparse and dense data were generated using the method described in Lu et al. (2000); Tung et al. (2003), the same method used to compare the EH-Apriori algorithm to FITI. The data synthesis method is a two step process that first generates a pool of candidate intertransaction associations and then uses this pool to populate the transactional data set. The characteristics and features of the generated data is defined by several parameters that guide the generation process. These parameters include the size of the intertransaction pool, the mean and maximum length of the intertransaction associations, the maximum number of unique items that may be in the data set and the maximum interval span of the associations. Table 4.3 lists the parameters used to create the data sets used in the experimentation.

Intertransaction association rule mining in FITI occurs only after the set of frequent intratransaction associations have been found. Knowledge of these rules is then used to transform the database into a lookup structure that aids intertransaction mining. For this experiment, FITI was implemented using the FP-Tree and FP-Growth algorithm for the initial mining phase. This was necessary in order for a fair comparison of the algorithms to be made, it having previously been shown that FP-Growth performs an order of magnitude faster than the Apriori algorithm used in the original FITI implementation (Han et al., 2000).

Table 4.3: Parameters used in the generation of the synthetic sparse and synthetic dense data sets.

Parameter	Sparse	Dense
Number of intratransactions	500	200
Size of the intertransaction source pool	50	200
Average length of intratransactions	5	25
Maximum length of intratransactions	10	50
Average length of intertransactions	5	8
Maximum length of intertransactions	10	20
Maximum number of unique items in the data	500	100
Maximum interval span of intertransactions	4	6

The algorithms were implemented in Ruby, an interpreted language, and benchmarked on a 3.2GHz Pentium 4 running FreeBSD.

Limitations of the Benchmark Environment

Before discussing results, execution time irregularities should be noted in the EFP-Growth curve in Figure 4.13(a) at $w = 2$ and at $w = 7$. Irregularities also appear for FITI in Figure 4.12(a) at the 1.1% support threshold, in Figure 4.12(b) at 1% support and in Figure 4.13(b) at $w = 4$.

Profiling revealed that these irregularities are caused by an erratic garbage collector in the Ruby interpreter. When triggered, the garbage collector will spend a disproportionally long time seeking memory to free.

This behaviour was consistently reproduced on the FreeBSD 5.3, Linux 2.6 and Windows XP platforms using the 1.6 and 1.8 Ruby interpreter series. This behaviour is independent of the algorithm being run and was found present in the implementations of the EH-Apriori, FITI, FP-Growth and EFP-Growth algorithms. The garbage collector behaved normally for all other points on the graphs and hence the irregularities found do not invalidate the results obtained.

Minimum Support Threshold

For the first set of results, the support threshold was gradually lowered from 1.6% to 0.6% with a fixed intertransaction window size of 4 and from 13% to 8% with a fixed window size of 6 for the synthetic sparse and synthetic dense data sets respectively.

The plot in Figure 4.11(a) shows FITI outperforming EFP-Growth until the 1% support threshold is reached. FITI has an advantage at the higher support thresholds as it is able to remove unnecessary data prior to counting. This benefit is reduced as the number of candidates generated by FITI increases when the support threshold is lowered. EFP-Growth outperforms FITI at the lower support thresholds and especially at the 0.6% level where an explosion in the number of rules, as seen in Figure 4.11(e), results in an exponential increase in the number of candidate itemsets generated and counted by FITI. The plot of the memory requirements of the two algorithms in Figure 4.11(c) shows that although FITI has greatly reduced memory needs compared to EFP-Growth at the higher support levels, it is the latter algorithm that displays more stable memory use as the number of rules increases exponentially. FITI has lower memory requirements at the higher support levels because it is able to discard many known infrequent associations which results in a low number of candidates being generated.

We begin to see an order of magnitude difference in the algorithm execution times on the dense data in Figure 4.11(b). Although FITI marginally outperforms EFP-Growth at the 12.5% and 13% support threshold, FITI is overwhelmed by the number of candidate itemsets generated at the lower thresholds. The memory requirements for the dense data set in Figure 4.11(d) shows FITI has an advantage at all but the 9% and lower support levels. Here the memory usage continues to increase rapidly for FITI whereas the peak memory requirement of EFP-Growth remains stable. The number of rules discovered at each support level are shown in Figure 4.12(f).

The execution times in Figure 4.12(a) and peak memory usage in Figure 4.12(c) for the first real world data set compares the algorithms' performance as the support threshold is lowered from 1.5% to 0.7%. An order of magnitude difference in the running times exists at the lower support levels due to the large number of discovered rules and a high number of FITI generated candidates. An exponential increase in the number of rules discovered, shown in Figure 4.13(e), is reflected in a jump in the peak memory use of the two algorithms. Both algorithms use similar amounts of memory up until this point.

Figure 4.12(b) depicts the execution time of EFP-Growth and FITI on the second real world data set over a support threshold range of 0.4% to 1.3%. EFP-Growth is able to maintain its computational advantage over FITI at all support levels. Memory use, shown in Figure 4.12(d), sees FITI again having an advantage only at the higher support levels where the number of rules, shown in Figure 4.14(f), and hence the number of candidates generated remains low.

Intertransaction Sliding Window Size

The intertransaction window size in Figure 4.13 is incremented from $w = 0$ to $w = 10$ for the sparse data and $w = 8$ for the dense data with fixed minimum support thresholds of 1% and 10% respectively.

Figure 4.13(a) shows that EFP-Growth has only a marginal computational advantage on the sparse data set, the number of rules found and the number of candidates generated by FITI remaining relatively low. The memory requirements in Figure 4.13(c) are seen to be increasing at a similar pace with FITI requiring slightly less memory than EFP-Growth until $w = 9$. Figure 4.13(e) depicts the number of rules retrieved with each window size. FITI has similar execution times to EFP-Growth on the dense data in Figure 4.13(b) set until the intertransaction size $w = 5$. The curves begin to diverge at this point, the FITI execution time eventually being an order of magnitude greater than EFP-Growth at $w = 8$. The FITI memory needs, from Figure 4.13(d), are overall lower than that for EFP-Growth but are growing exponentially with respect to the sliding window size due to the number of candidate itemsets being created. The number of rules mined at each window size are given in Figure 4.13(f).

Performance on the real world data is compared by incrementing the sliding window size up to $w = 12$ to find associations spanning up to an hour. The support thresholds are fixed at 1% and 0.4% for the first and second data sets.

EFP-Growth outpaced FITI computationally in both real world data sets in Figure 4.14(a) and Figure 4.14(b). A sudden increase in the execution time of the FITI algorithm is seen in Figure 4.14(b) when the sliding window size is increased from $w = 2$ to $w = 3$. This increase is caused by a sudden large jump in the number of rules being discovered as can be seen in Figure 4.14(e). The memory requirements of the algorithms remain similar until $w = 8$ for both Figure 4.14(c) and Figure 4.14(d). The EFP-Growth memory use remains stable while FITI continues to increase linearly as the number of rules, seen in Figure 4.14(e), being discovered begins to taper off at this point.

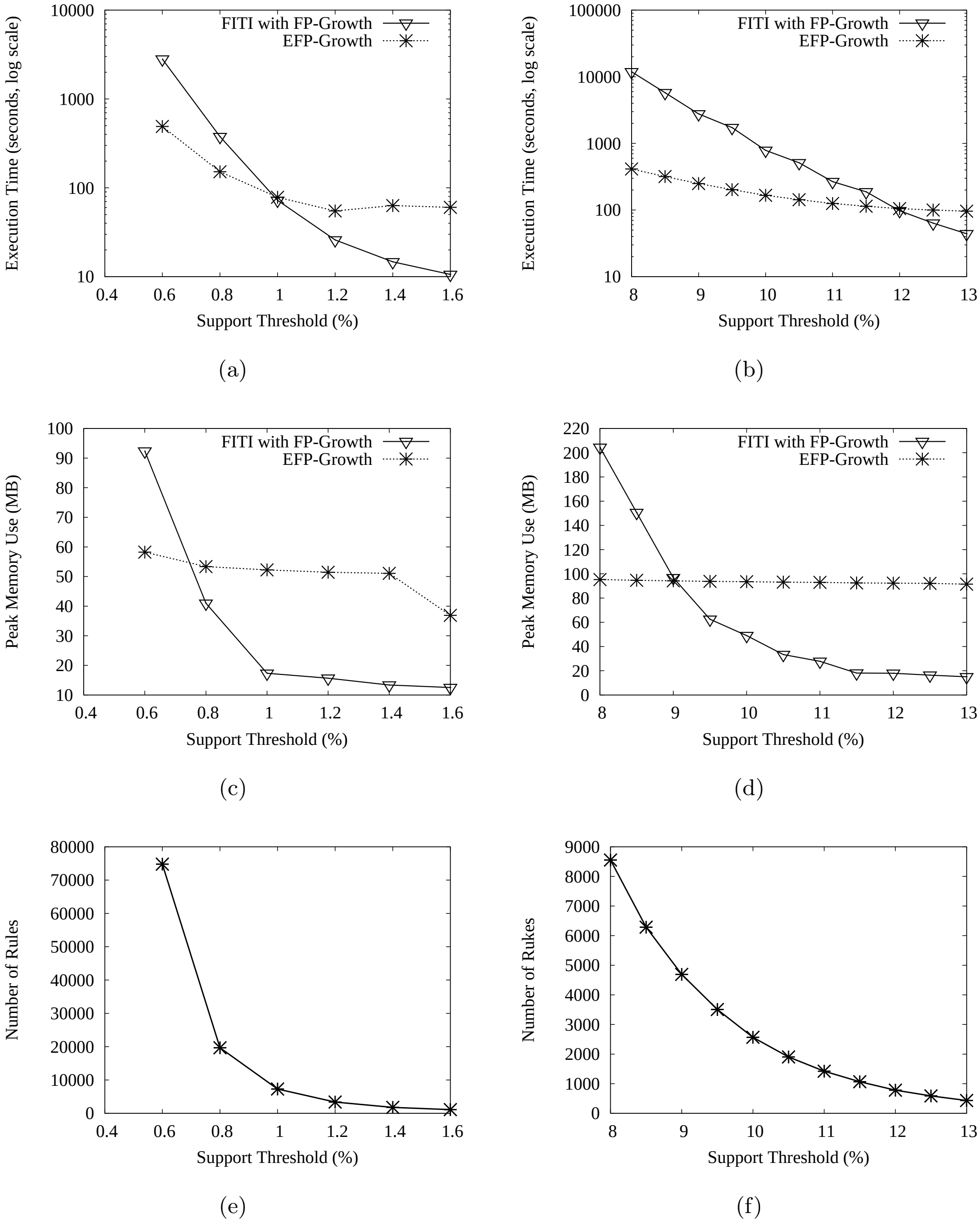


Figure 4.11: The execution time (a), memory use (c) and the number of rules found (e) for the synthetic sparse data set with the intertransaction window size fixed at $w = 4$ and the execution time (b), memory use (d) and the number of rules found (f) for the synthetic dense data set with $w = 6$ as the minimum support threshold is adjusted.

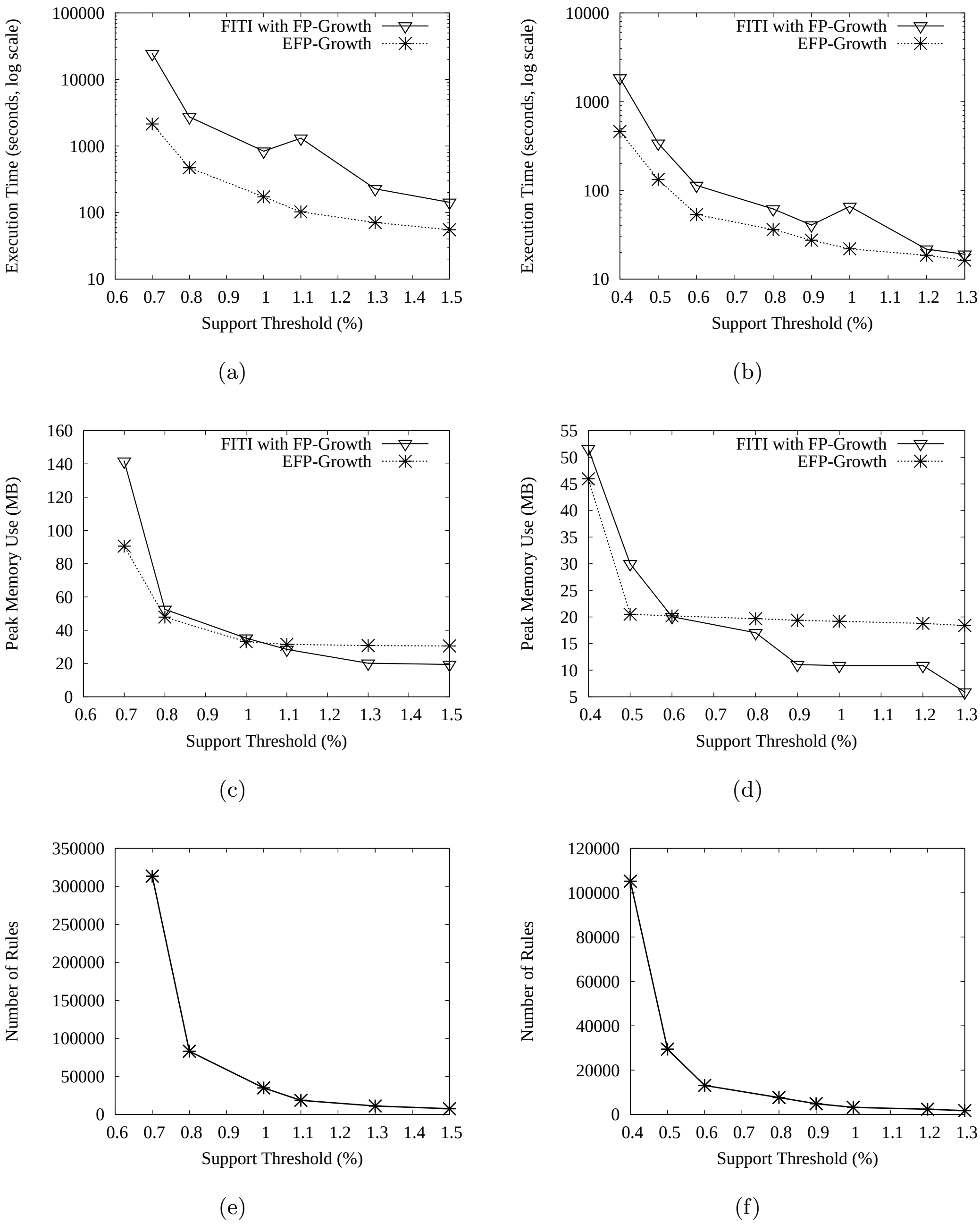


Figure 4.12: The execution time (a), memory use (c) and the number of rules found (e) for the working professional subject and the execution time (b), memory use (d) and the number of rules found (f) for the retiree subject as the minimum support threshold is adjusted. The intertransaction window size is fixed at $w = 6$ with an interval size of 300 seconds for both data sets.

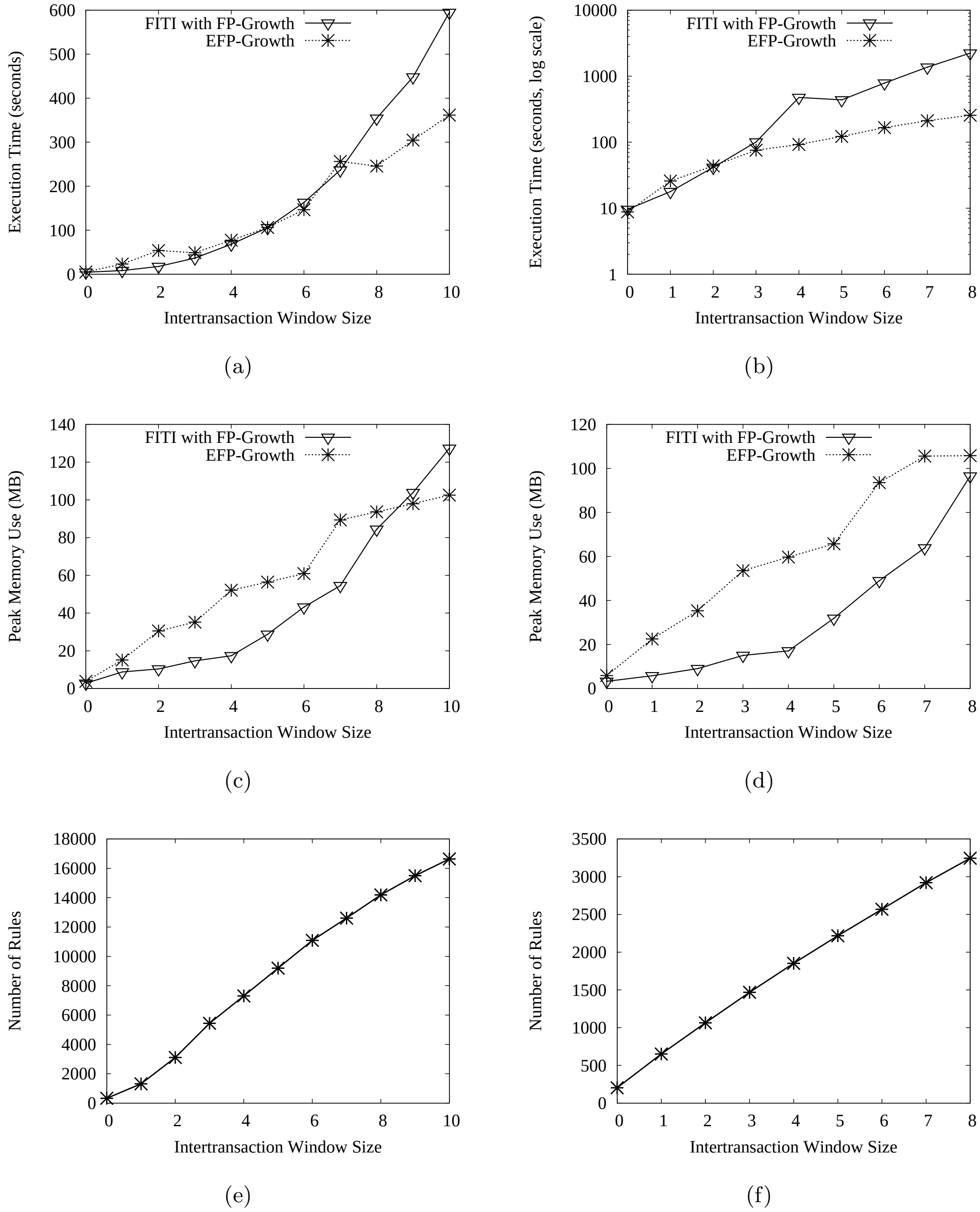


Figure 4.13: The execution time (a), memory use (c) and the number of rules found (e) for the synthetic sparse data set and the execution time (b), memory use (d) and the number of rules found (f) for the synthetic dense data set as the intertransaction window size is increased. The minimum support level was fixed at 1% and 10% for the sparse and dense data sets respectively.

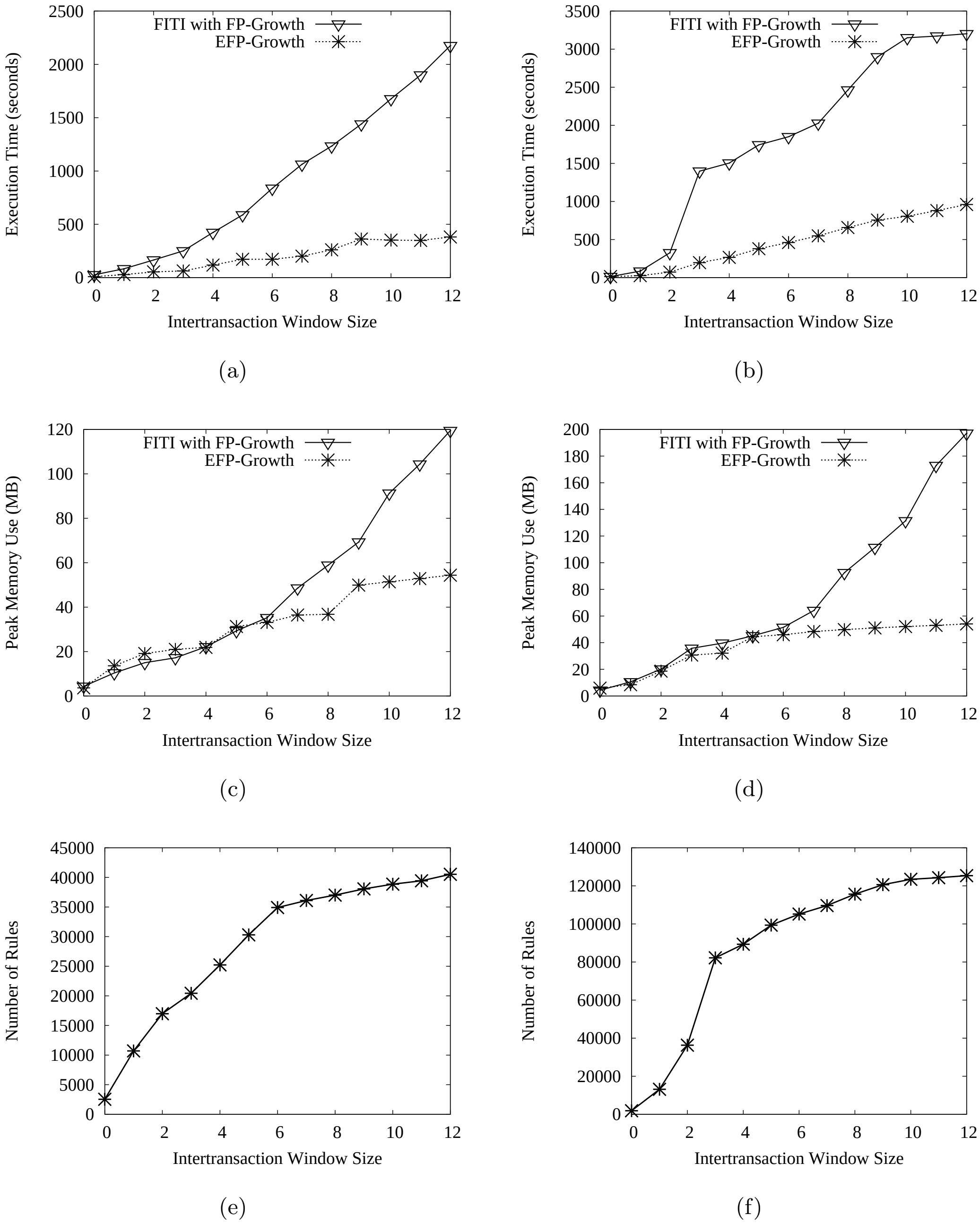


Figure 4.14: The execution time (a), memory use (c) and the number of rules found (e) for the working professional subject with a minimum support threshold $\alpha = 1\%$ and the execution time (b), memory use (d) and the number of rules found (f) for the retiree subject with $\alpha = 0.4\%$ as the intertransaction window size is increased.

Conclusion

It has been shown that EFP-Growth is a more scalable algorithm for IAR mining than FITI. The candidate generation and testing approach of the latter was found to limit its scalability when the number of discovered itemsets, and hence the number of candidates being generated and counted, become too large. In contrast, EFP-Growth was shown to be able to scale well in such cases, especially when applied to the synthetic dense data and at the lower support levels of the synthetic sparse data set. EFP-Growth was also shown to outperform FITI on the real world data sets, making it a more suitable algorithm for IAR mining on the sensor event logs from an intelligent environment than the EH-Apriori and FITI algorithms.

4.3 Identifying Emergent Behaviours

The mining technique presented up to this point offers an efficient structure for the representation and retrieval of the complete set of intertransaction associations at an arbitrary support level. Making sense of the mined rules and identifying patterns that are representative of abnormal behaviour will require further processing, however, as described in this section. Here, emergent IAR mining is introduced and shown how it may be employed in a novel application for the detection of new and changing behaviours. An example application of emergent IAR mining is shown using the real world data sets previously introduced in Section 4.2.3.

4.3.1 Emergent Intertransaction Association Rules

The discovery of emergent intertransaction associations seeks to find those rules that display significant growth in a database of new observations DB_N over a historical data set DB_H. An association rule r is said to be emergent when its growth, the ratio of its support in DB_N to its support in DB_H, is greater than or equal to some threshold δ and a minimum support threshold α on DB_N has been met. The growth function in Equation 4.5 (Dong and Li, 1999) has been adopted for this work.

$$growth\left(r|DB_H, DB_N\right) = \begin{cases} 0, & \text{if } sup\left(r|DB_H\right) = 0 \text{ and } sup\left(r|DB_N\right) = 0 \\ \infty, & \text{if } sup\left(r|DB_H\right) = 0 \text{ and } sup\left(r|DB_N\right) \neq 0 \\ \frac{sup(r|DB_N)}{sup(r|DB_H)} & \text{otherwise} \end{cases} \tag{4.5}$$

Section 4.2 has shown that association rule mining on a database populated with many frequent items can become computationally expensive, especially at the lower support thresholds where exponential growth in the number of rules present may result in a computational explosion. In this application of IAR mining, it is reasonable to expect that the historical data set will be larger and contain a much greater number of associations than will be present in the new data set. It is impractical, therefore, to find the complete set of association rules for both DB_H and DB_N and then compare these sets to find the emergent association rules when the rules in DB_N are expected to be a subset of those in DB_H.

The EFP-Tree is instead employed as an intermediate representation of both DB_H and DB_N and item constraints (Srikant et al., 1997) are applied to the mining of DB_H so that only a desired subset of rules are extracted. This gives the system the ability to "query" DB_H on the historical frequency of only those associations found to be frequent in DB_N.

4.3.2 Minimal Emergent Intertransaction Association Rules

Only the set of minimal emergent rules are sought to prevent the discovery process from returning an overwhelming number of rules, the majority of which would be unlikely to offer any valuable information not already present in the minimal set. Finding the set of minimal rules was achieved by ordering the mined associations by ascending rule length and discarding those rules known to contain emergent association subsets using previously detected emergent rules. An emergent association q is said to be non-minimal when $\exists \{r, t\}$ such that $r(t) \subset q\,(t = 0)$ where $r(t) = \left\{r^1_{d_1+t}, r^2_{d_2+t}, \ldots r^i_{d_i+t}, \ldots r^Z_{d_Z+t}\right\}$ is a known emergent rule whose intertransaction offsets $\{d_1, d_2, \ldots d_i, \ldots d_Z\}$ have been incremented t intervals. For example, if r is an emergent rule $\mathrm{C}_0, \mathrm{D}_2 \Rightarrow \mathrm{E}_2$ and q is the rule $\mathrm{A}_0, \mathrm{B}_0, \mathrm{C}_1, \mathrm{D}_3 \Rightarrow \mathrm{E}_3$ then q is also known to be emergent because $r(1) \subset q$ holds true.

Although filtering was applied as a post process, the EFP-Tree offers an opportunity to move the filtering process into the tree mining algorithm so as to guide the mining of both DB_H and DB_N.

Only those associations whose frequencies were queried in the historical data set are included in the reported rule counts in the observations made on the working professional data set in Section 4.3.3 and the observations made on the retiree data set in Section 4.3.4.

4.3.3 Observations on the Working Professional Data Set

The sixteen days worth of event logs from the first subject were divided into two weeks, each week containing eight days of events. Each week was in turn used as the historical database DB_{H} to find the emergent rules present in the other. Intertransaction association rules were mined using an intertransaction sliding window of size $w = 6$ with transaction intervals of five minutes using a raw minimum support threshold of $\alpha = 8$ for both weeks of data. This support level was chosen to balance the quality and the quantity of the discovered associations, providing an ample number of rules for analysis while remaining resistant to noise. Noise, in this application, refers to IARs in DB_{N} that are classified emergent due only due to a lack of historical data and are not representative of true new behaviour.

Examination of the data sets revealed several instances of unusual behaviour that have been interpreted to be caused by sensor malfunctions. These conclusions are substantiated by Tapia et al. (2004) who note that sensors were observed to fail or were dislodged during the data gathering period. Such analysis is further reinforced through inspection of the sensor event logs which revealed several instances of rapid and repetitive toggling of a sensor's state.

Association rules discovered in DB_{N} were classified emergent if a minimum growth of $\delta = 5$ was measured.

Emergent Behaviour in Week One

Mining the first week of data lead to the discovery of 101 emergent intertransaction associations from 1,808 investigated rules. Examining the context of each rule revealed the existence of four distinct groups of associations.

The first group was found to relate to a flurry of kitchen activity on the third day of the event logs. Here the subject appears to be triggering the sensors on a variety of kitchen appliances and furniture over an unusually long period, behaviour that results in significant growth in kitchen related associations. Manual examination of the emergent rules and the log data did not, however, reveal any discernible behavioural trends aside from the unusual length of the activity and the frequency of the events therein.

The second set of rules highlight significant growth in kitchen door associated events. Inspection of the event logs reveals repeated opening and closing of the door on day four of the data. Although it is possible that these readings were caused by the subject frequently entering and leaving the kitchen, we believe that the length of the activity and the regularity with which the readings occur suggests that a momentary glitch with the sensor has been detected.

Analysis of the emergent rules from the third group shows that it is unusual to observe the medicine cabinet and the foyer door being opened or closed in adjacent transactions. This discovery is attributed to insufficient historical data; discretisation of the event logs resulted in the foyer door and the medicine cabinet events being predominately grouped into adjacent intervals in the first week and into single transactions in the second. The event logs show that it is quite normal in both weeks of data for the subject to walk through the foyer door shortly before opening the medicine cabinet.

The last set of emergent rules features the subject both flushing the toilet and using the bathroom medicine cabinet in a single interval. These two events occur normally one or two intervals apart in the historical data set but not in the same transaction. This behaviour does not appear to be due to any unusual circumstances and again suggests that a larger historical data set is required.

Emergent Behaviour in Week Two

Only 46 of the 2,437 intertransaction associations tested from the second week were identified as emergent. Inspection of these rules again revealed four groups of contextually related associations.

The first group is a set of two single events revealing that the bathroom door is being opened and shut significantly more often in the second week than in the first. This innocuous new behaviour is evident over several days of data and is confirmed through manual inspection of the event logs.

Twenty four rules associating use of the microwave with use of the fridge, the freezer and the kitchen cabinet within a single transaction period make up the second group of associations. The historical data shows that while we can expect some of these kitchen based events to occur within a single interval, the use of the microwave in addition to two or more of these events all within a single transaction is unusual. Examination of the event logs showed that meal preparations appear to be a more involved process in the second week than in the first.

The third set of emergent rules suggest that it is abnormal for the hot and cold water faucets in the bathroom to be turned both on and off in adjacent intervals. It appears that in the first week the hot and cold water is used in shorter sessions and that these events normally occur within a single transaction. The event logs suggest that the subject does appear to be spending more time in the bathroom and is opening the water faucets more frequently than in the first week. It remains conceivable, however, that this phenomenon is caused by a sensor aberration and not a change in behaviour.

Although the rules in the final set of emergent associations highlight no apparent abnormal behaviour, they do document significant growth in the number of times the bedroom jewellery box is opened and the bedroom light switch is activated in contiguous transactions.

4.3.4 Observations on the Retiree Data Set

As for the working professional subject, the sixteen days of event logs from the second subject were divided into two weeks and each week was again used in turn as the historical database DB_{H} to find the emergent rules present in the other. The minimum support threshold for mining the first week of data was reduced to $\alpha = 6$ in order to provide us with a sufficient number of rules for analysis not otherwise available at higher support levels on this particular data set. The support level for the second week was again set to $\alpha = 8$ and a minimum growth measurement of $\delta = 5$ was required for both weeks.

Emergent Behaviour in Week One

Of the 1,061 intertransaction associations tested, 72 were classified emergent. The emergent rules were again grouped into sets of related associations.

The first of these groups reveal that the subject is watching more television in the first week than in the second. The log files confirm that interaction with the TV is common in the first week yet it is rarely seen in the second.

A second group of associations highlight an anomaly centered around the repeated opening and closing of the kitchen door one and two intervals after it had previously been opened or closed. Delving into the event logs reveals a considerable increase in the number of times the door is opened in this week relative to the historical data set. It appears that the subject is carrying out activities that require frequent access to the kitchen over several days of data and that this is not due to a sensor malfunction as was previously discovered in Section 4.3.3 for the working professional.

The next group of emergent rules highlight an unexpected behaviour prevalent on the first and sixth day. Consulting the log data shows that the bathroom door is frequently opened and closed over a fifteen minute interval on these days yet the historical data suggests that the bathroom door is normally left open or shut for periods of time outside the span of the intertransaction sliding window.

A bevy of emergent rules covering significant growth in the relationships among a variety of objects in the kitchen make up the final set. Emergent rules highlight growth between use of the microwave and the garbage disposal unit, the repeated opening and closing of the kitchen door and repeated access to the fridge and microwave. It is difficult to locate any anomalous behaviour in the log files as all these events also appear in close proximity in the second week. This suggests that there is not sufficient historical data available to account for all the combinations of intervals and events found here.

Emergent Behaviour in Week Two

The presence of four distinct groups of associations were once again identified amongst the 23 emergent associations discovered from the 758 that were tested.

The first group of emergent associations is made up of two single events that reveal the subject has accessed their study drawer more frequently in the second week than in the first. Inspection of the logs confirms this find, the study drawer being opened and closed in seven transactions in week two but only twice in week one.

The rules in the second group imply abnormal behaviour in the form of the shower faucet being repeatedly turned on and off that takes place on the first day of the week. Although it is possible that these rules were due to some new behaviour, the frequency and regularity of the readings over a period of fifteen minutes suggests the possibility that they were the result of an error with the sensors.

The next unusual behaviour is a repeated toggling of the light switch in the butler's pantry on day seven. The emergent rules show that the light switch in the butler's pantry is being turned on for several minutes and then turned off again in a cycle that is repeated several times every few minutes. The expected behaviour associated with the butler's pantry is to see the light switch activated for several minutes up to an hour at a time with longer periods of time passing before the light is turned on again. A single rule in this group highlights frequent use of the microwave twenty five to thirty minutes after the cabinet inside the butler's pantry has been accessed. It is difficult to verify whether these patterns are caused by a new behaviour or not without querying the subject.

The final emergent rules present five new associations related to the use of the kitchen cabinet. The first three of these rules show a new association between the cabinet and the microwave, the latter of which is frequently being opened and closed twenty five to thirty five minutes after the cabinet has been accessed. The last two rules show that the closing of the cabinet door, the use of the cold water faucet and the opening and closing of the fridge all within a single transaction is an unexpected but frequent occurrence in the second week. None of these rules suggest a new behaviour that we need to be concerned about.

4.3.5 Automatic Selection of the Intertransaction Sliding Window Size

Selection of a suitable intertransaction sliding window size for the mining of an arbitrary data set can be achieved by noting when the number of rules being discovered begins to taper off as the size of the window is increased. This heuristic can be used to indicate when a large majority of the associations present in the data have been discovered and when further increases in the length of the window is likely to only provide a marginal number of new rules.

For example, Figure 4.15 shows the number of rules being discovered as the window size is increased for each week of data for both the working professional and the retiree subject given a raw minimum support threshold of $\alpha = 6$ and five minute transaction intervals. The number of rules mined for the first week of data from the working professional, presented in Figure 4.15(a), appears to taper off when a window size of circa $w = 16$ is reached. For the second week, the number of rules being discovered in Figure 4.15(b) begins to fall off when a window size of approximately $w = 19$ is reached. The point at which the number of rules in the retiree data set begins to taper off is approximately $w = 9$ for both the first and second week of data as shown in Figure 4.15(c) and Figure 4.15(d) respectively.

The number of rules returned by the mining process begins to taper off at around $w = 12$ for the first week of data from the working professional in Figure 4.16(a) when the support threshold is set to $\alpha = 8$. Figure 4.16(b) shows that the majority of rules present in the second week of the working professional data have been found by $w = 4$. For the retiree subject, the number of rules being returned tapers off at $w = 5$ for the

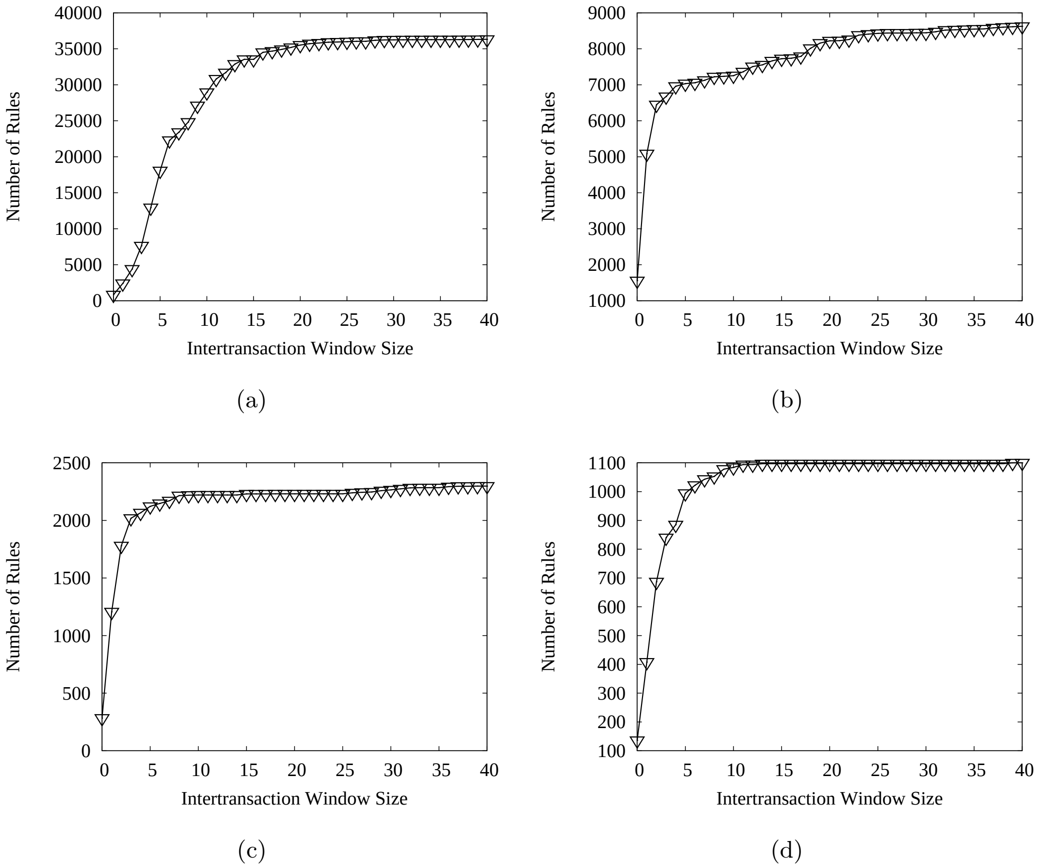
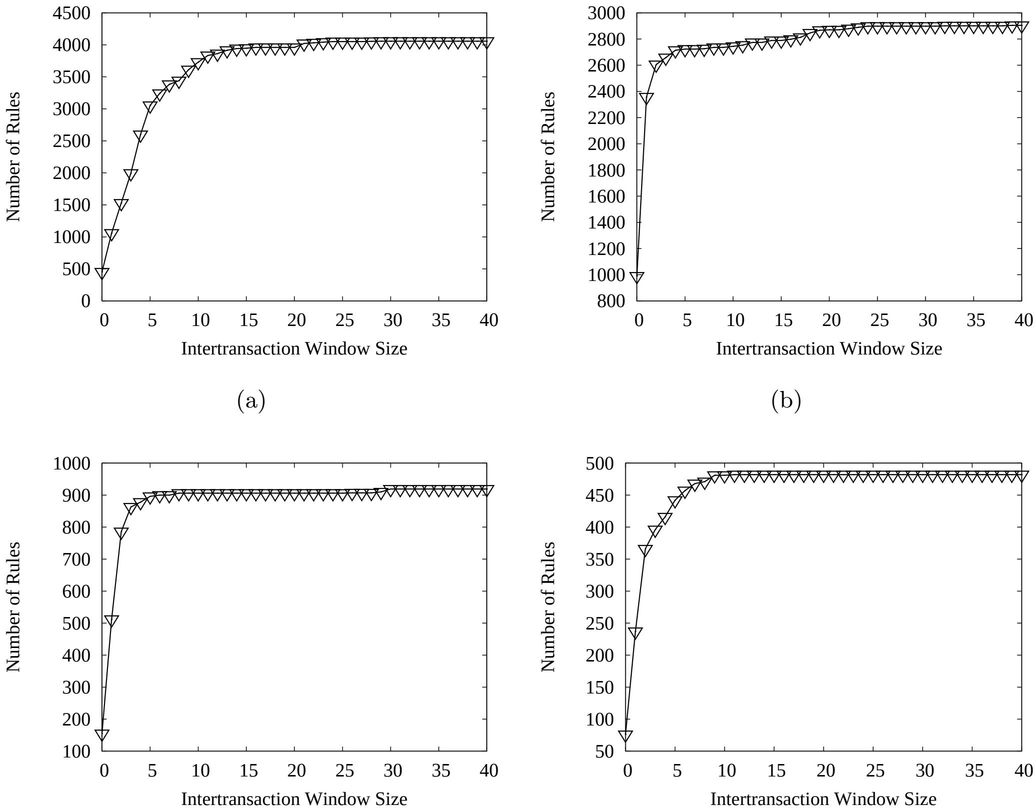


Figure 4.15: The number of rules discovered in the (a) first and (b) second week of the working professional data set and the (a) first and (b) second week of the retiree data set as the sliding intertransaction window size is increased. Five minute transaction intervals and a raw minimum support threshold of $\alpha = 6$ was used.



(a) (b)

(c) (d)

Figure 4.16: The number of rules discovered in the (a) first and (b) second week of the working professional data set and the (a) first and (b) second week of the retiree data set as the sliding intertransaction window size is increased. Five minute transaction intervals and a raw minimum support threshold of $\alpha = 8$ was used.

first week and at $w = 9$ for the second week of data. The number of rules discovered here are shown in Figure 4.16(c) and Figure 4.16(d) for the first and second week respectively.

In the experimentation in the Section 4.3.3 and Section 4.3.4 an intertransaction window of size of $w = 6$ with transaction intervals of five minutes was chosen to limit the intertransaction associations to a half hour period. Were this constraint not in place then the heuristic could be applied to select a suitable window size.

4.4 Conclusion

Motivated by the limited application of graphical models towards the recognition of interleaved activities, this chapter has introduced a data mining inspired approach for the recognition of new and changing human behaviour using emergent intertransaction association rule mining.

The EFP-Tree and EFP-Growth algorithms for intertransaction association rule mining have been described and experimental results comparing the computational performance of EFP-Growth to FITI have been presented. EFP-Growth was shown to scale well with respect to the minimum support threshold and the intertransaction spanning window length, particularly with the synthetic dense data set and data from the homes of two real world subjects where the presence of many frequent items results in a combinatorial explosion of candidate itemsets at the lower support thresholds and with the larger intertransaction window sizes. The benefit gained by FITI pruning known infrequent intratransaction item combinations prior to intertransaction mining is diminished when the number of discovered rules, and hence the number of candidate itemsets that are required to be counted, grows too large.

The EFP-Tree was used in the mining of emergent intertransaction association rules and their application to the discovery of behavioural changes and sensor aberrations present in the state-change sensor event logs from the homes of the two volunteer subjects. Item constraints were applied to the EFP-Growth mining as an efficient means of querying the EFP-Tree of an intertransaction association's historical support measure. Experimental results show that emergent associations are able to be applied to the discovery of short term abnormalities and to detect changes that occur in a subject's behavioural patterns over a span of several days. Both intratransaction and intertransaction anomalies were detected, attesting to the benefit of incorporating temporal associative relationships into the mining process.

Unfortunately, the interpretation of the discovered emergent rules through the manual inspection of the original event logs remains a tedious and time consuming occupation. A more streamlined method for the examination and analysis of discovered rules is required for the intelligent analysis of the emergent rules. This will be the topic of the next chapter.

Chapter 5

Visual Exploration and Analysis of Emergent Behaviours

Manual inspection of the sensor event logs has been noted to be a tedious and time consuming approach to analysing the context in which emergent intertransaction associations occur. The underlying issue here is the overwhelming amount of information that must be manually trawled in order to derive insight into the rules. The linear presentation of the event data makes understanding the context of a person's emergent behaviour difficult for several reasons. First, it is impossible for users to rapidly gain an overview of the data. Users are instead forced to scroll through the log files and remember the date and time of events. Also, navigation to events of interest in an event log file is limited to string matching searches and manual identification of instances of emergent rules within the log files is cumbersome and error prone.

A visual data mining alternative that overcomes these limitations is needed, the design and application of which is the focus of this chapter. The interface that is proposed assists users in interpreting the emergent rules by highlighting their occurrence in the context of the original data. An overview of the interface and the rationale behind its design are discussed in Section 5.1 and Section 5.2 respectively. Experimentation on the real world data sets from Chapter 4 is then repeated in Section 5.3 using the visual interface. Observations made using the tool are compared to insight gained from the manual inspection of the event logs previously described in Section 4.3.3 and Section 4.3.4.

5.1 An Interface for the Visual Exploration of Emergent IARs

The previous works of Brunk et al. (1997); Agrawal et al. (1996); Wong et al. (1999); Ong et al. (2002); Blanchard et al. (2003) have attempted to tackle the issue of visual exploration of complete sets of mined association rules. In contrast, the visualisation approach proposed in this chapter seeks to limit users to consider only those IARs found to be emergent. These rules are likely to be a minority of the rules discovered and hence they will be difficult to discern or gain meaning from using any form of visualisation that covers all rules. A way of visualising the effect of the emergent rules in isolation and in the context of the original data is necessary. This is done by mapping the emergent IARs back onto the original data in order to establish the original context in which the rules occur. Doing so allows users to see both the date and time that the emergent behaviour occurred and which other sensors were triggered around this time. Although it was not available for the data set used in the experimentation, a corresponding video of each instance of the unusual behaviour could also be retrieved and shown to the user.

A screen capture of the proposed interface with a sample data set loaded is shown in Figure 5.1. The emergent rules are displayed in a table in the top portion of the screen. The rules are selectable and can be ordered by their support, growth measures or by rule similarity. The main display element is a grid that displays a compact view of the sensors triggered in each transaction interval. The horizontal axis represent the date and time of the transaction intervals while the horizontal axis shows the sensor events grouped by their room location. Triggered events are indicated with blue coloured cells while cells that correspond to currently selected rules are highlighted in red.

Users are able to select whether they wish to view a compact representation of the time line, where only intervals in which events were recorded are shown, or the full grid. The thickness of the vertical lines between cells is used to indicate jumps in time in the compact view, the cells of contiguous intervals being delineated by hairlines while the cells of non-contiguous intervals are separated by thick lines. All cells in the full time line view are contiguous and hence are delineated using hairlines. Figure 5.2(a) presents a magnified view of the grid in which the effect of the compact view is more visible. In contrast, Figure 5.2(b) shows a magnified view of a small subset of the same time period displayed using the expanded view. The start of a new day is indicated with a gap in the time line in both the compact and full view modes. Users are able to view the exact date and time of an interval by positioning the mouse cursor over a cell,

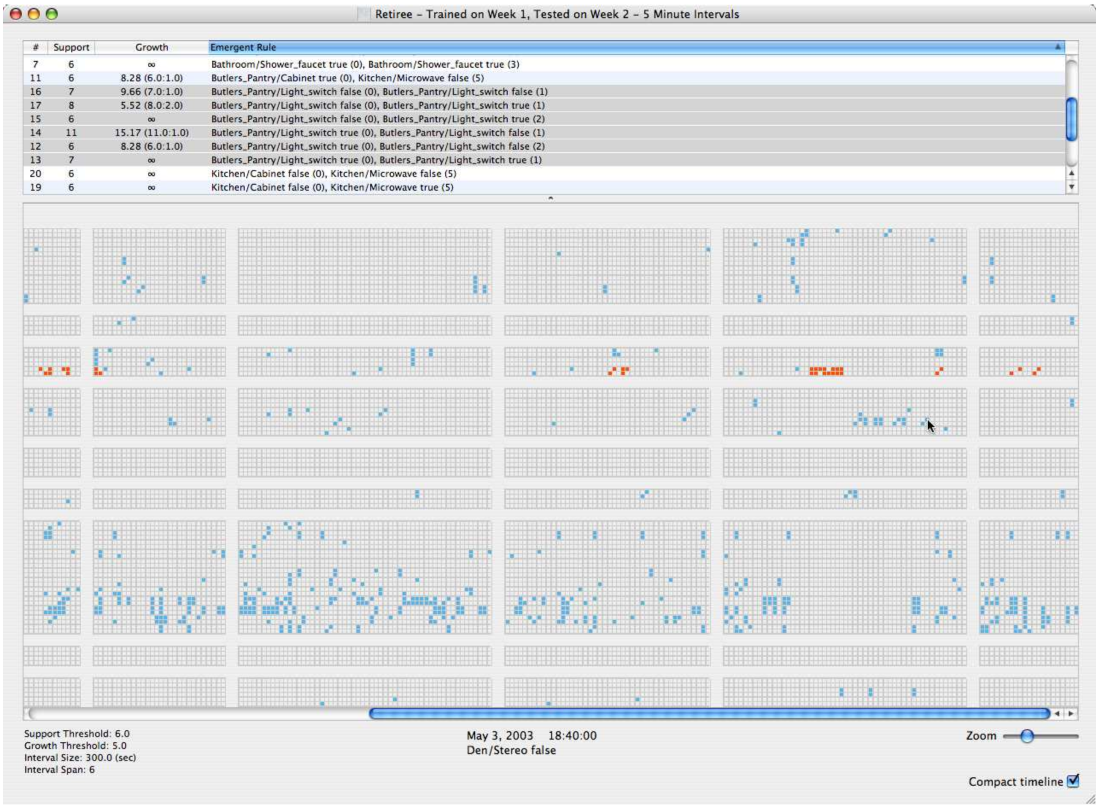
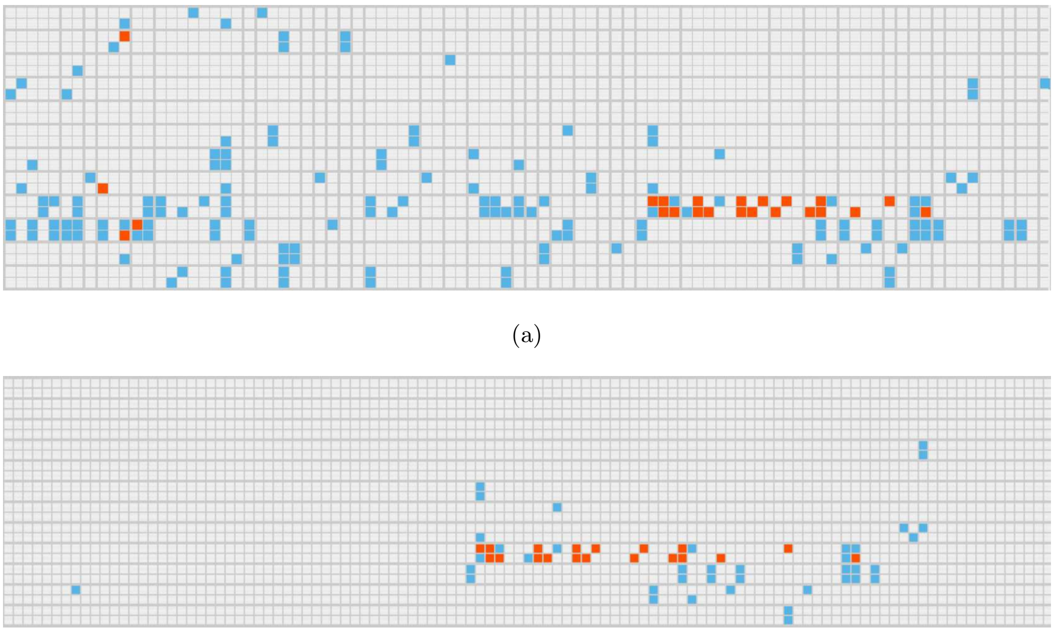


Figure 5.1: The visual interface highlighting the presence of selected rules (top) on the event time line. The horizontal axis represents the date and time of the intervals while the vertical axis represents the sensor events. The date and time and the name of the sensor event that the mouse cursor is pointing at is being displayed.

the time being indicated in a text box horizontally centered underneath the grid. If the cell being pointed at represents a triggered sensor event then the name of that event is also displayed in the text box. The screen capture in Figure 5.1, for example, shows the descriptive name of the sensor event, and the date and time of the interval in which it occurs, of the cell that the mouse cursor is presently pointed at. In this example, the event represents the switching off of the stereo in the retiree subject's den.

Cells are similarly partitioned on the vertical axis by the sensor events that they represent. Sensors are grouped by their location in the home with gaps on the grid delineating rooms. Sensor events are further grouped by their textual description such that all events related to one particular sensor, currently limited to "on" and "off" events in our application, appear in contiguous cells whose boundary is drawn using hairlines. Boundaries between the sensors in a room are delineated using thick horizontal lines.



(a)

(b)

Figure 5.2: Magnified view of (a) the compact time line showing the sensor events within a single room over a period of one day and (b) the expanded view showing approximately three hours from the same room and period.

Partitioning the sensor events in this way provides users with a clear overview of the time and location of events being triggered and their relation to other sensors and rooms.

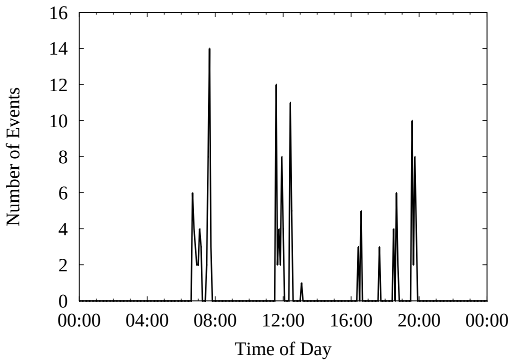
The bottom portion of the interface is used to display meta information about the data being displayed. The minimum support and growth thresholds, the interval size and the intertransaction window size used to mine the rules are shown here. A text box horizontally centered underneath the cell grid displays information about a cell whenever the user positions the mouse cursor over it. This area also contains a slider that allows the user to adjust the magnification factor of the grid and a check box with which to toggle the compact time line representation.

5.2 Design Rationale

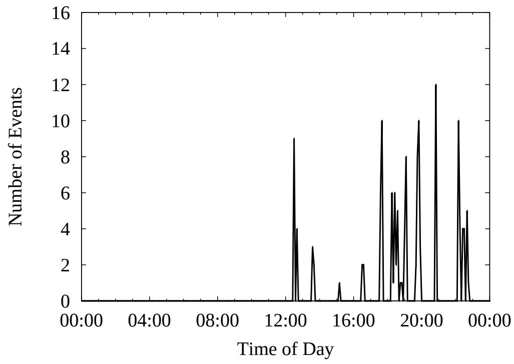
The interface design was motivated by a desire to provide users with an intuitive and easy to use system for analysing the emergent IARs in the context of the original data. The first logical step in the design was to map the rules back onto the original data space so as to re-establish the context in which the behaviours occur.

Analysis of the levels of activity present in the data sets revealed the presence of many regions of time in which no sensor events were recorded. For example, the amount of activity present in the working professional subject data set, as a count of the number of sensor events per five minute interval, is shown in Figure 5.3, Figure 5.4 and Figure 5.5. Activity levels for the retiree subject is similarly shown in Figure 5.6, Figure 5.7 and Figure 5.8. These show the presence of long periods of inactivity during the night for both subjects as well as periods of inactivity during the day for the working professional. The display of such periods of inactivity offer users negligible benefit when interpreting the emergent rules, it being the presence of unusually frequent combinations of activities and not their absence that is important and captured by the mining process. Providing users with the ability to select a compact representation of the time line is therefore logical given that users may consider the maximisation of the amount of information displayed on screen more valuable than displaying these vacant intervals.

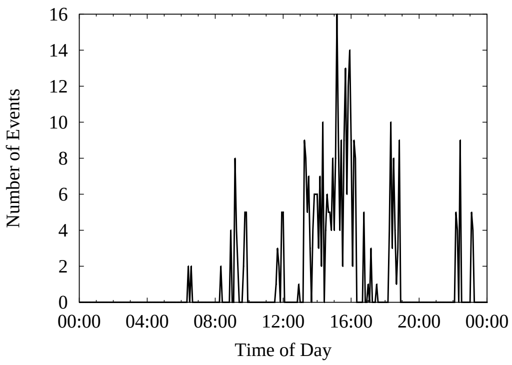
The elimination of the empty time intervals from the time line requires a method by which the now non-linear passage of time can be highlighted. Thick vertical lines and vertical hairlines were chosen as logical visual markers for the non-contiguous and contiguous cell boundaries respectively. The inclusion of gaps on the time line was found to be necessary in order to delineate between days and to prevent them from



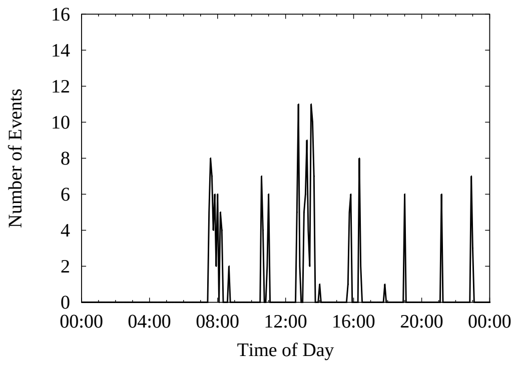
(a) Day 1



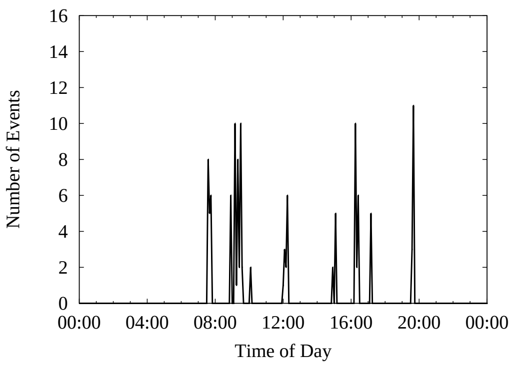
(b) Day 2



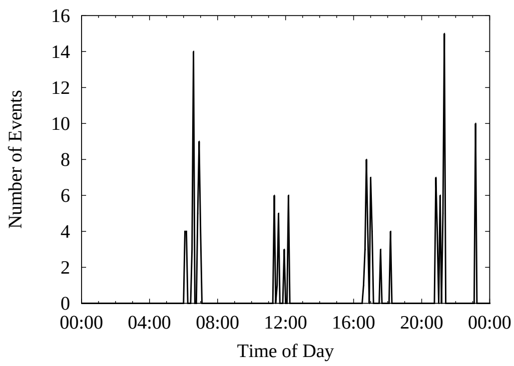
(c) Day 3



(d) Day 4

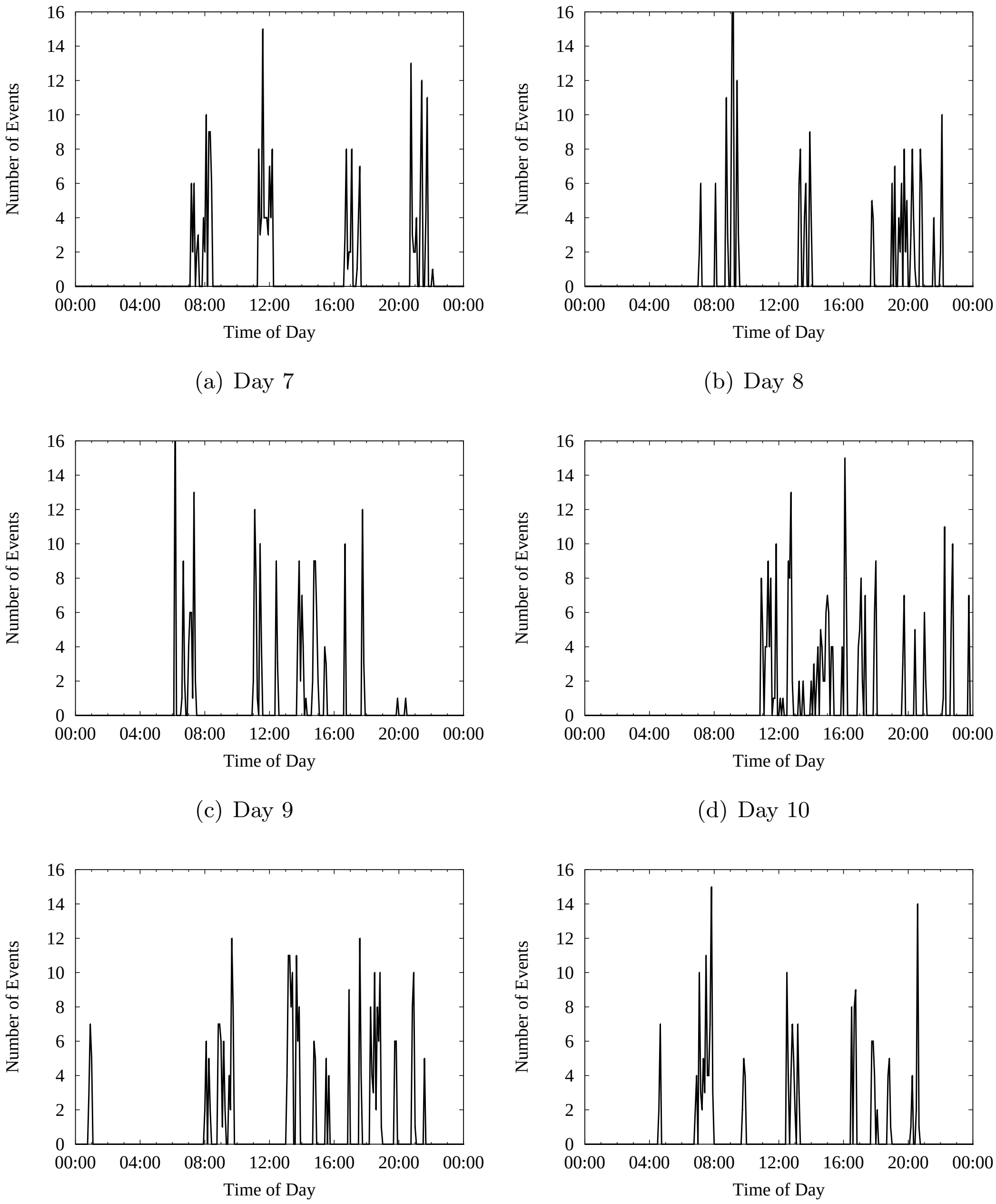


(e) Day 5



(f) Day 6

Figure 5.3: The number of sensor events triggered per five minute interval over the first six days of data from the working professional subject data set.

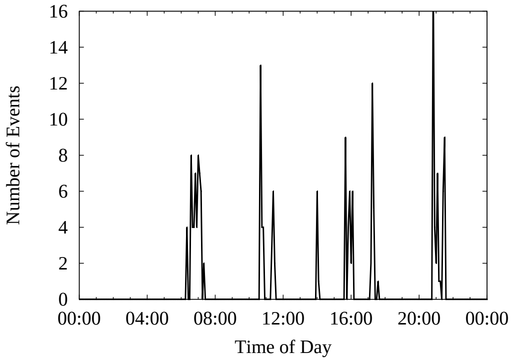


(a) Day 7 (b) Day 8

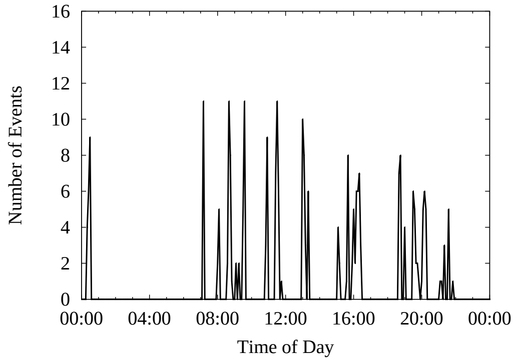
(c) Day 9 (d) Day 10

(e) Day 11 (f) Day 12

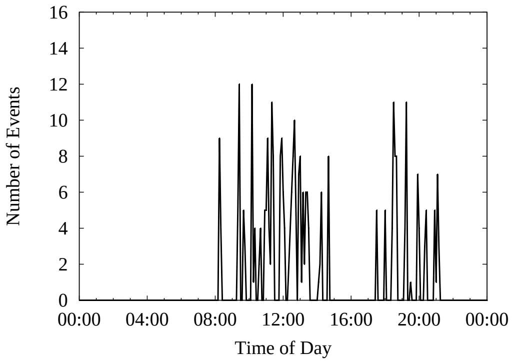
Figure 5.4: The number of sensor events triggered per five minute interval over the days 7 to 8 of data from the working professional subject data set.



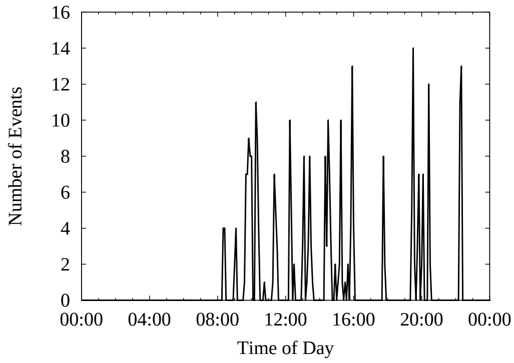
(a) Day 13



(b) Day 14

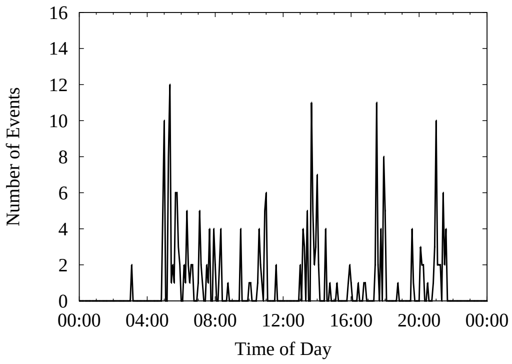


(c) Day 15

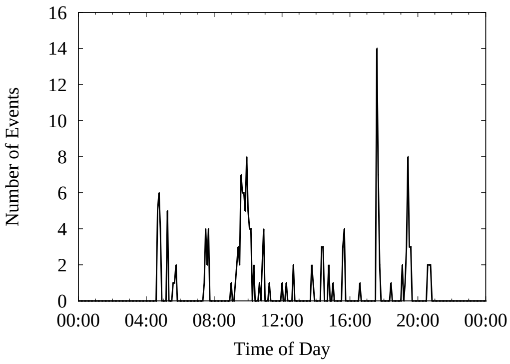


(d) Day 16

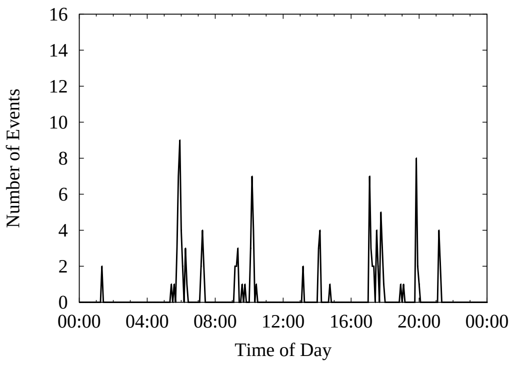
Figure 5.5: The number of sensor events triggered per five minute interval over the last four days of data from the working professional subject data set.



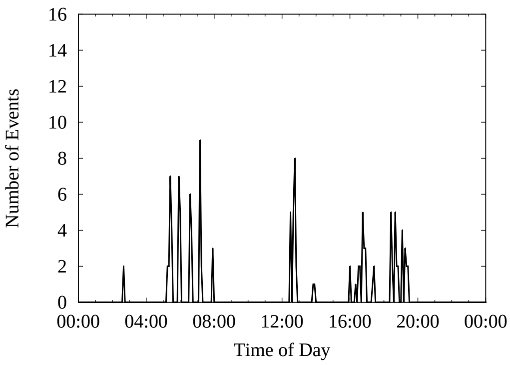
(a) Day 1



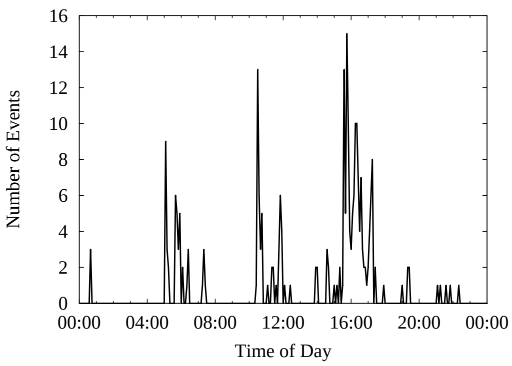
(b) Day 2



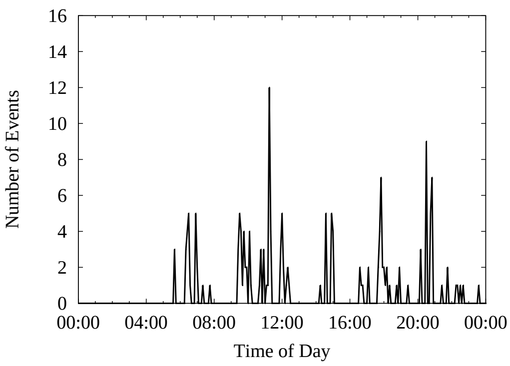
(c) Day 3



(d) Day 4

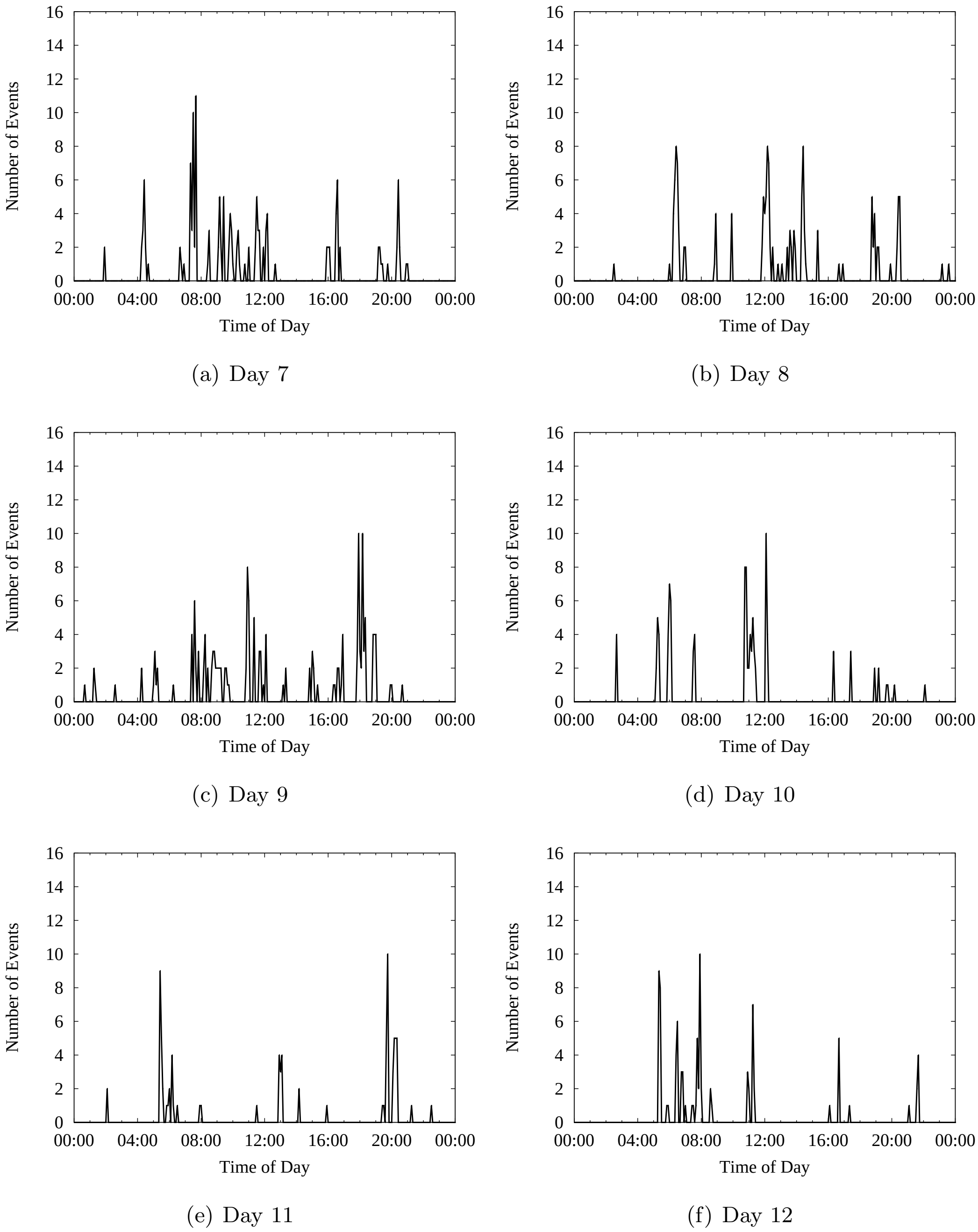


(e) Day 5



(f) Day 6

Figure 5.6: The number of sensor events triggered per five minute interval over the first six days of data from the retiree subject data set.

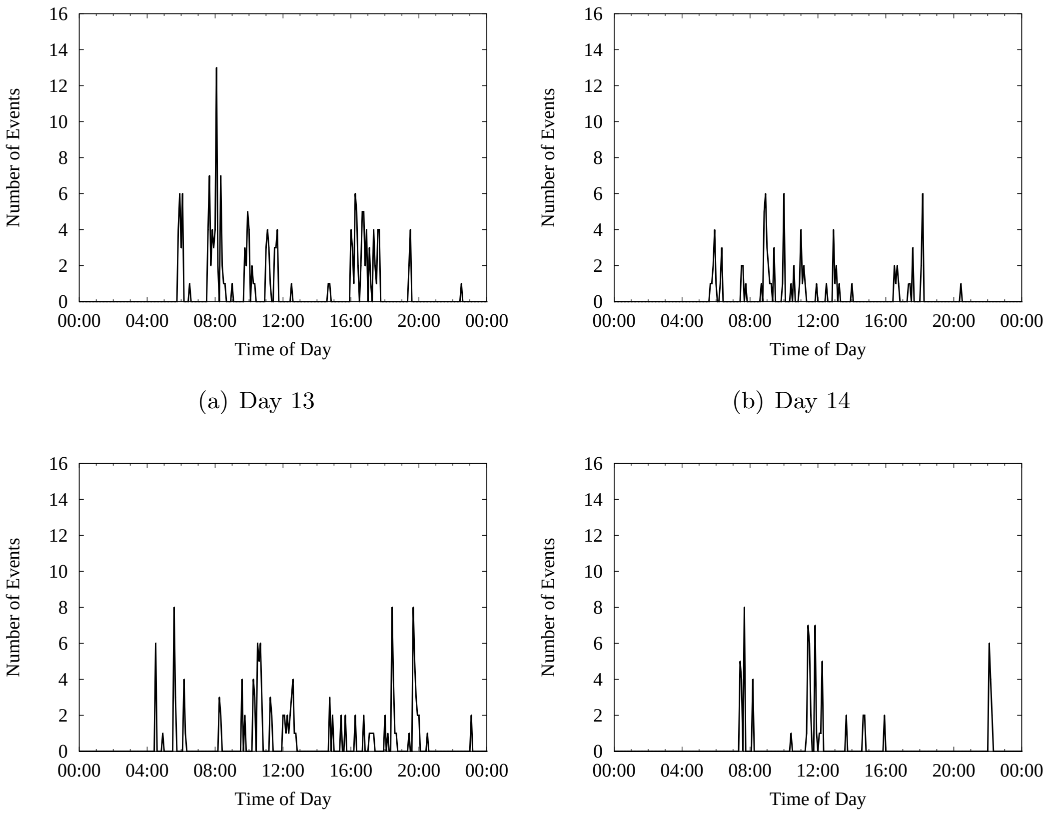


(a) Day 7 (b) Day 8

(c) Day 9 (d) Day 10

(e) Day 11 (f) Day 12

Figure 5.7: The number of sensor events triggered per five minute interval over days 7 to 8 of data from the retiree subject data set.



(a) Day 13

(b) Day 14

(c) Day 15

(d) Day 16

Figure 5.8: The number of sensor events triggered per five minute interval over the last four days of data from the retiree subject data set.

melding together. A common complaint from users prior to the inclusion of these gaps was that it was possible to lose one's place in the grid when scrolling the time line view.

The decision to show the transaction intervals on the time line rather than the event sequences was made to avoid a mismatch between the associative relationships modeled by transactional mining and the event ordering implied by such an event time line. To have done so would be counter-intuitive to users and introduce unnecessary complexity in both the display and in the interpretation of the rules. It would also work against the previous goal of displaying as much information on screen as possible with the compact time line view.

Separate rows are used to display the state change events of the sensors. This may, at first, appear to be an inefficient method of visualising the sensor states given that only a single row per sensor is needed when colour or texture is used to represent the sensor state. The use of multiple rows per sensor was, however, a conscious decision to allow for future data sets where an arbitrary number of states may be assigned to a sensor. A more sophisticated light switch may, for example, indicate whether it is in a state of `off`, `on` or `dimmed`. The application of colour or texture to indicate such states is likely to quickly become confusing as the number of sensor states within a home increases. The problem is exacerbated when more than one state is assigned to a single event. The aforementioned light switch sensor may, for instance, trigger both `on` and `dimmed` events when it becomes activated.

Given that multiple event rows are assigned to the sensors, it naturally follows that events should be grouped by their associated sensor. Thick horizontal lines are used to visually mark these groupings. Having grouped the sensor events in this fashion, the next logical step is to further group the sensors by their room location in order to establish their spatial context. Gaps in the grid structure are used here to clearly delineate the room groupings. Grouping the sensor events helps achieve a logical and intuitive ordering of the sensors on the grid structure.

Users are allowed to select multiple rules from the emergent IAR table as more than one rule is likely to describe an emergent behaviour. Cells belonging to different rules describing the same behaviour can therefore be simultaneously highlighted.

The choice to provide meta information only about a single cell over which the user's mouse is positioned stems from the numerous number of sensor events that can be displayed on screen. A textual overlay onto or adjacent to the grid for each and every visible event would produce clutter and make the display difficult to discern at the lower magnification levels. The use of the meta information text box is therefore a compromise between providing users with too much information and giving users the ability to extract further information about the events and rules that they are investigating. The resulting display is a compact time line representation that, when combined with the sensor groupings, provides visual cues and reference points to the relative time and location of items of interest while the meta information text box provides detailed information about events the user wishes to focus on.

Finally, the shading of the red and blue colours used to paint triggered sensor events and to highlight the emergent rules was made with consideration of colourblind users in mind. The colours "Vermilion" and "Blue Sky" were chosen from the proposed colour pallet of Okabe and Ito (2002). These colours are said to be unambiguous to people of both full colour vision and of all types of colour blindness.

The final design is the culmination of a series of logical design steps and interaction with users aimed at producing an easy to use and intuitive interface for the exploration of emergent intertransaction association rules in the original spatio-temporal context of the data.

5.3 Repeat Experimentation on Real World Data Sets

The visual interface for emergent IAR exploration was applied to the real world data sets previously discussed in Section 4.3.3 and Section 4.3.4. Observations previously made through manual inspection of the discovered rules and the event logs will, in this section, be compared against those made using the visual tool. It was found that the use of this tool helped attain greater insight into the emergent behaviours discovered, and in a shorter period of time, than was possible to obtain through manual inspection of the rules in the log file context.

Mining was performed with interval sizes of one to five minutes using one minute increments. The sliding intertransaction window length was adjusted to maintain, as closely as possible, a thirty minute window for each of the chosen interval sizes. Mining using the four and five minute intervals generally provided more stable, noise free, results than the one to three minute intervals due to the limited amount of training data available. The benefit of reduced noise was diminished somewhat with the increase in the granularity of the results. Importantly, the cause of the vast majority of emergent behaviours, whether valid or the result of a lack of historical data, was apparent regardless of the interval size chosen.

5.3.1 Observations on the Working Professional Data Set

As in Chapter 4, intertransaction association rules were mined using a raw minimum support threshold of $\alpha = 8$ and a minimum growth of $\delta = 5$ for both weeks of data. The new interpretations of these rules, and any deviations from earlier observations, are discussed.

Week One

A large group of emergent IARs related to a flurry of activity in the subject's kitchen were discovered in the third day from the first week of data. The discovered rules describe the temporal relationships of the kitchen drawer, cabinet and refrigerator spanning transaction intervals over a thirty minute period. This suggests that the length of time in which these events frequently occur is unusual rather than the underlying activity itself as only the minimal emergent rules are mined. These patterns do not, however, appear to signify any anomalous behaviour given the overall context of kitchen related activity being carried out. The interpretation of these rules concurs with the observations previously made in Section 4.3.3. A screen capture of this kitchen activity with all of these emergent associations highlighted is shown in Figure 5.9.

An abnormality does appear during this kitchen activity, however, in the form of a repeated opening and closing of the kitchen door. The regularity, frequency and the length of time over which this occurs is unusual. The lack of activity outside of the kitchen reinforces the suspicion that a glitch with the door sensor has been discovered here. The phenomenon is repeated again over an eight minute period early the next day. Again, this interpretation concurs with observation made in Section 4.3.3. A magnified view of the interface showing this behaviour using an interval length of two minutes is shown in Figure 5.10.

The third and fourth set of rules of bathroom related events described for the first week of data in Section 4.3.3 are readily classified as noise when viewed in their original context using the interface.

Week Two

A significant portion of the emergent rules found in the second week again relate to activity in the kitchen. Unlike the kitchen related rules from the first week, these rules describe emergent relationships that are new combinations of kitchen sensors

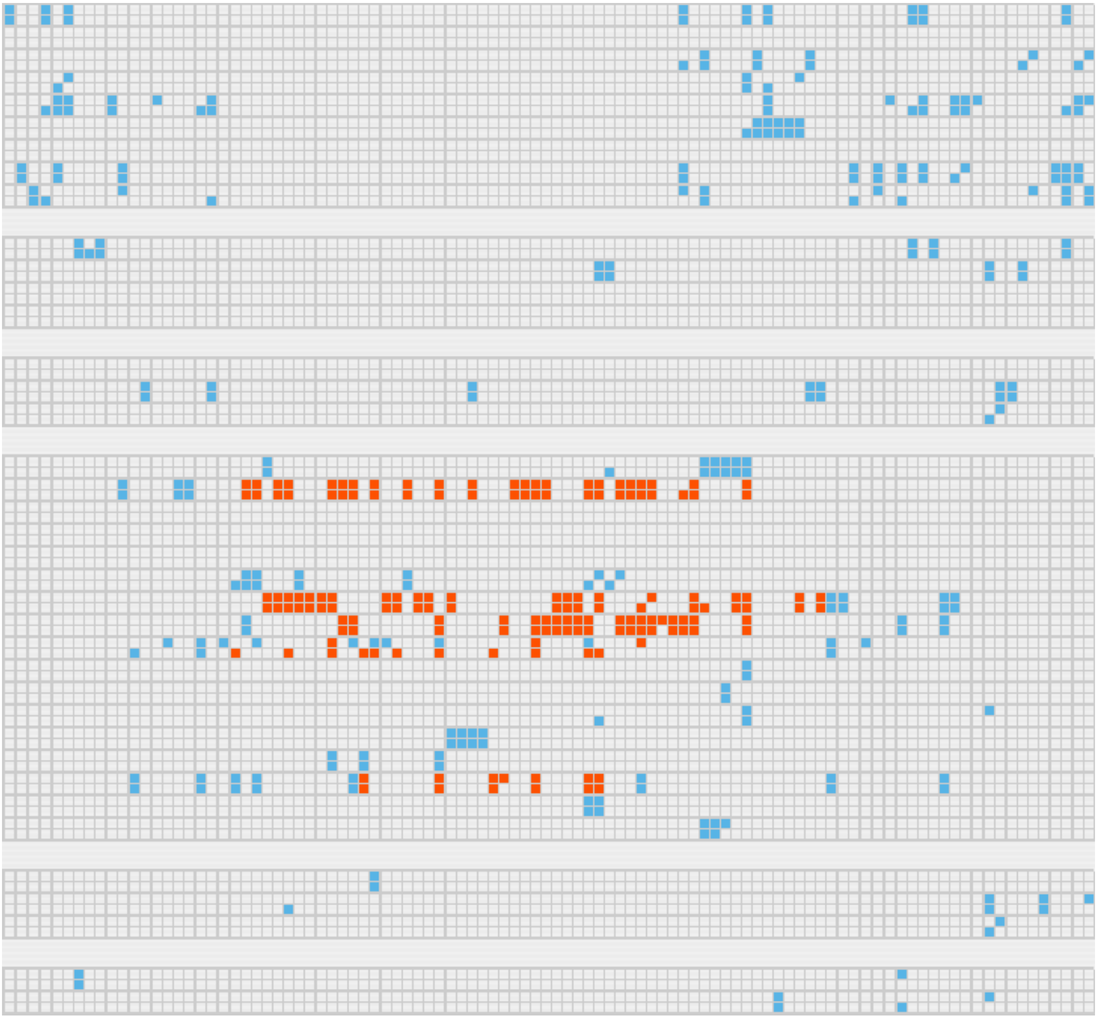


Figure 5.9: Visualisation of the flurry of kitchen activity on the third day of the first week of the working professional data using a three minute transaction interval. The discovered rules describe temporal relationships between the kitchen drawer, cabinet and refrigerator not seen in the training data. Triggered events are coloured in blue while events matching the emergent IARs are coloured in red.

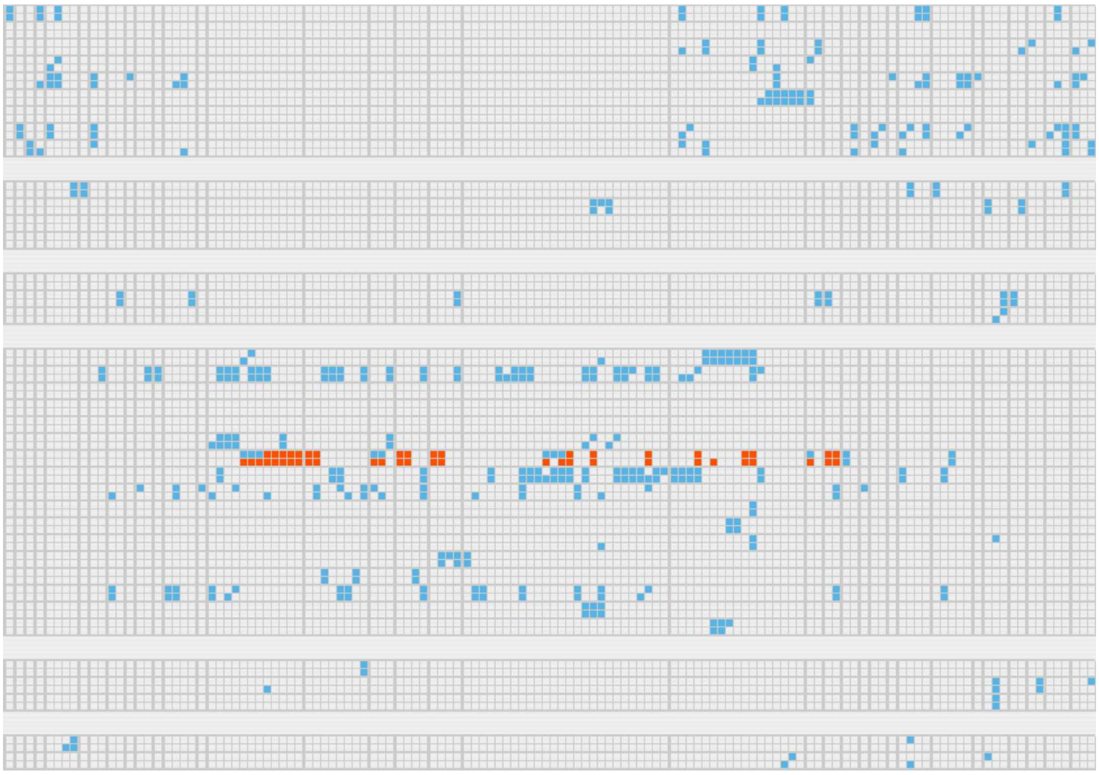


Figure 5.10: Visualisation of the abnormal kitchen door activity using a two minute transaction interval on the third day of the first week of the working professional data. The regularity, frequency and length of time over which the kitchen door is repeatedly opened and closed is unusual. Triggered events are coloured in blue while events matching the emergent IARs are coloured in red.

whose temporal relationship span only a few intervals. This suggests that the kitchen activities in this week are more involved and that not enough historical data is available to account for these patterns.

A lack of historical data also explains the emergent IARs describing the opening and closing of the bathroom sink faucets one to two intervals after their previous use. This behaviour, highlighted in Figure 5.11, is rarely seen in the first week where we see these events within single transaction intervals without being repeated for some time.

A mundane change in the subject's behaviour is apparent, however, with the use of the bathroom door being significantly more frequent this week; the bathroom door is rarely used in the first week.

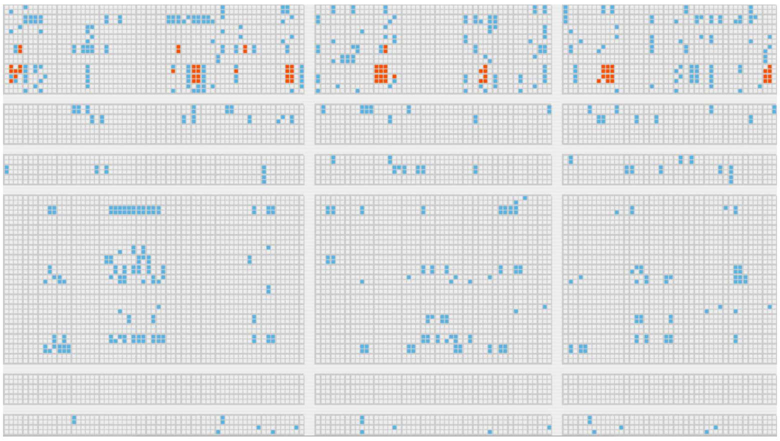
The interpretation of all the rules from this week concur with the observations previously made in Section 4.3.3.

5.3.2 Observations on the Retiree Data Set

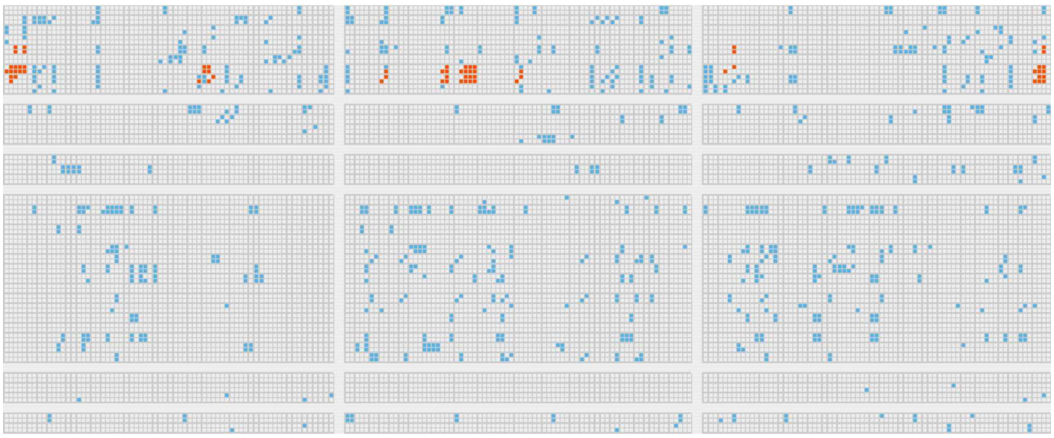
As in Section 4.3.4, the rules from the second subject were again mined using a minimum support threshold of $\alpha = 6$ and $\alpha = 8$ for the first and second week respectively. A minimum growth measurement of $\delta = 5$ was required for both weeks of data. The new interpretation of these rules, and any deviations from earlier observations, are again discussed.

Week One

A noticeable change in the retiree subject's use of the bathroom is evident on the first morning in the first week. Here, we see a forty-five minute period in which the bathroom door is repeatedly being opened and closed. The television in the living room is operated and the microwave and the refrigerator in the kitchen are also accessed during this time. In this context, the emergent IARs suggest that the subject is making frequent short visits to the bathroom, a behaviour that is unusual and should warrant further investigation. Although this behaviour was also present in the observations from Section 4.3.4, the extent of the abnormality was not as apparent; this new behaviour was previously dismissed as unusual but not worthy of further investigation. Figure 5.12 shows a screen capture of this behaviour highlighted on the interface. This discovery highlights the difficulty of analysing the emergent associations using manual inspection of the sensor event logs due to the overwhelming number of recorded events and the



(a)



(b)

Figure 5.11: Visualisation of the opening and closing of the bathroom sink faucets one to two minutes after their previous use on (a) days three to five and on (b) days six to eight of the second week of the working professional. The rules are shown using a three minute transaction interval. Triggered events are coloured in blue while events matching the emergent IARs are coloured in red.

linear format in which they are presented.

Emergent rules were discovered that describe the kitchen door being repeatedly opened and closed throughout a twenty minute period on the second day. The behaviour we expect to see is this door being left in one state for longer periods of time or to see it being both opened and closed within a single time interval. This behaviour, depicted in Figure 5.13, appears to be innocuous given the activity seen in the rest of the home.

The subject appears to be carrying out more kitchen related activity in this week than in the next. This explains the discovery of a host of emergent IARs involving kitchen sensor events. These patterns do not indicate any unusual behaviour, however. Rather, they reinforce the notion that we do not possess enough historical data to account for the wide variety of patterns present in normal kitchen behaviour. The same conclusion was made through manual inspection of the emergent IARs and the event logs in Section 4.3.4. Figure 5.14 shows these rules occurring for this week.

A behaviour that is unusual, however, is the frequent use of the television throughout this week. This behaviour is unusual given that the subject rarely turns on the television in the historical data set.

All of the above observations were previously discovered in Section 4.3.4. A new observation that previously went unnoticed, however, is an unusual pattern of toaster usage on the first day of data from this week. Here, the toaster is seen to be repeatedly toggled on and off over an eleven minute period in what appears to be the mundane activity of making toast. This innocuous behaviour, seen in Figure 5.15, is emergent due to the repetition of the toaster events over an unusually long interval.

Week Two

A prominent example of abnormal behaviour is apparent on the first day in the second week of data. Here, emergent IARs highlight a malfunction in the hot and cold shower faucet sensors as evident through the regular and repeated triggering of faucet events over a forty-five minute interval. The theory that this behaviour is due to a hardware glitch is further reinforced by the conspicuous absence of any further shower faucet events for the remainder of the week. The absence of this sensor from the remainder of the week went unnoticed in the observations in Section 4.3.4. Figure 5.16 shows the day on which the malfunction occurs.

On the seventh day the subject seems to repeatedly enter and leave the butler's pantry

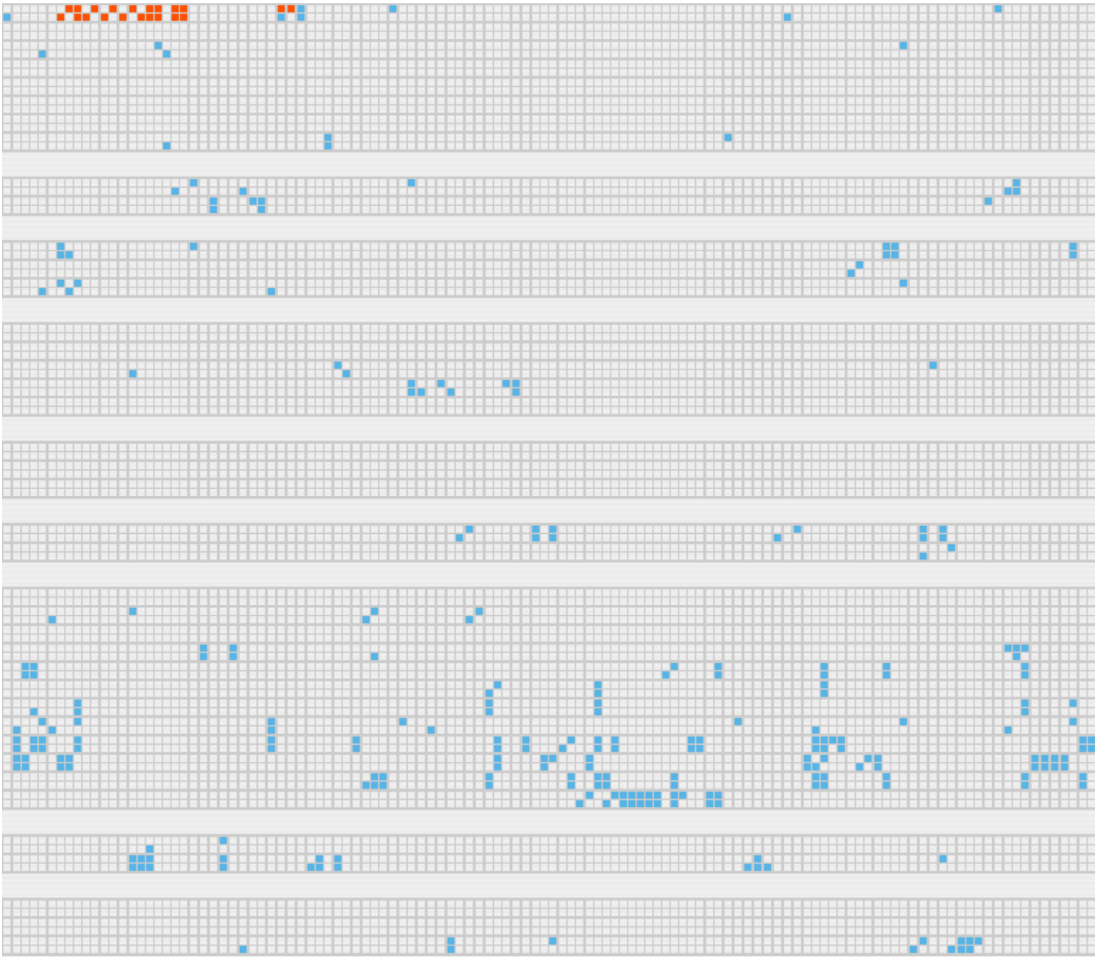


Figure 5.12: Visualisation of the abnormal bathroom activity using a two minute transaction interval on the first day of the first week of the retiree data. The subject is seen to repeatedly open and close the bathroom door over a forty-five minute interval. Activity in the lounge room and the kitchen suggests that frequent short visits to the bathroom are being made. Triggered events are coloured in blue while events matching the emergent IARs are coloured in red.

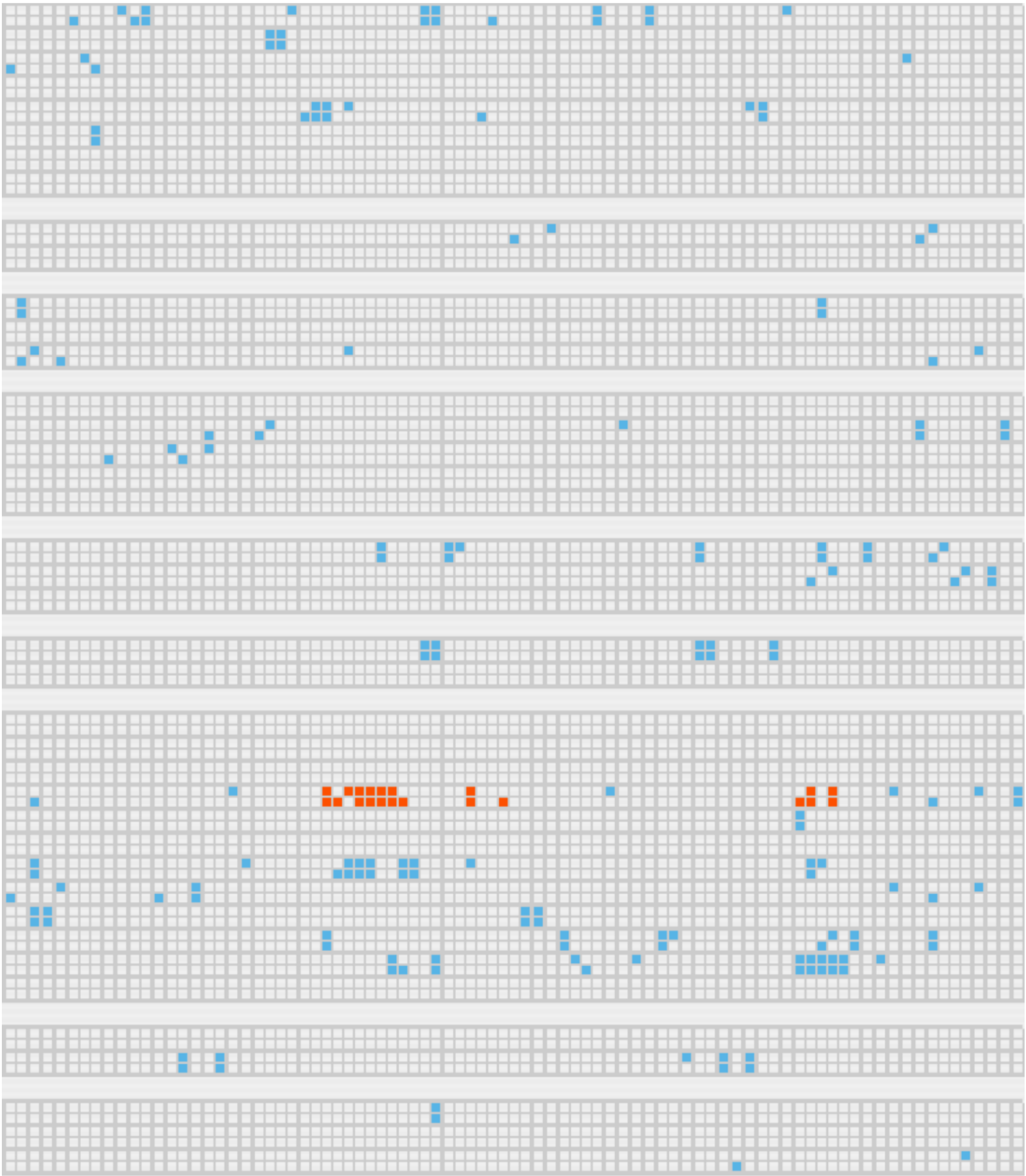
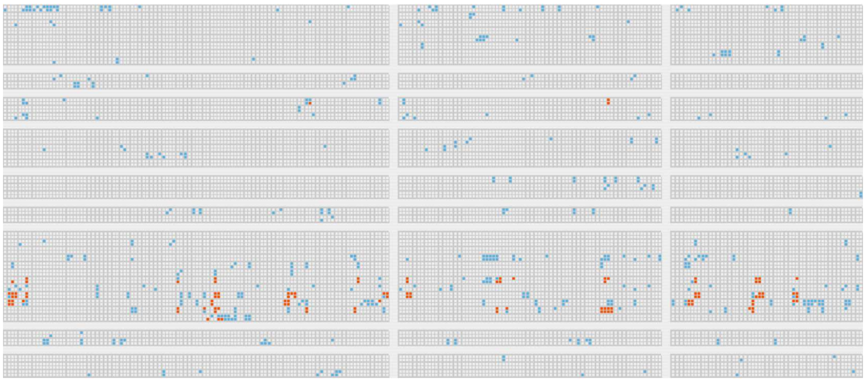
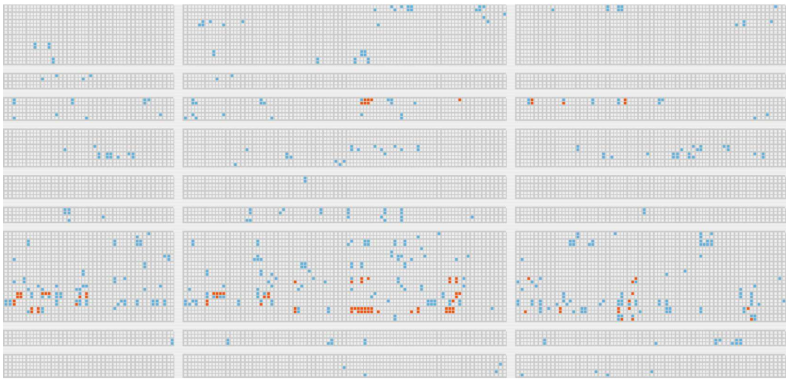


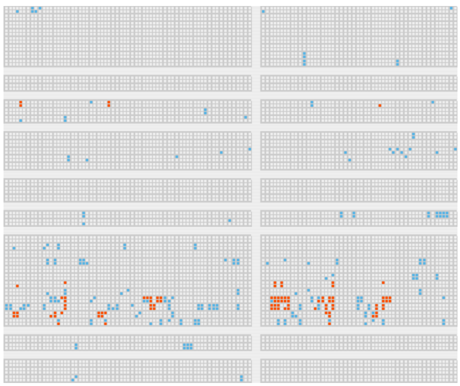
Figure 5.13: Visualisation of the innocuous emergent kitchen door behaviour on the second day of the first week of the retiree data set shown on a two minute transaction interval. The historical data sets suggests that the kitchen door would normally remain in its open or closed state for longer periods of time. Triggered events are coloured in blue while events matching the emergent IARs are coloured in red.



(a)



(b)



(c)

Figure 5.14: Visualisation of the kitchen related activity on (a) days one to three, (b) days four to six and (c) days seven and eight of the first week of the retiree data set. The presence of this emergent behaviour appears to be caused by a lack of historical support. The behaviour is shown using a two minute transaction interval. Triggered events are coloured in blue while events matching the emergent IARs are coloured in red.

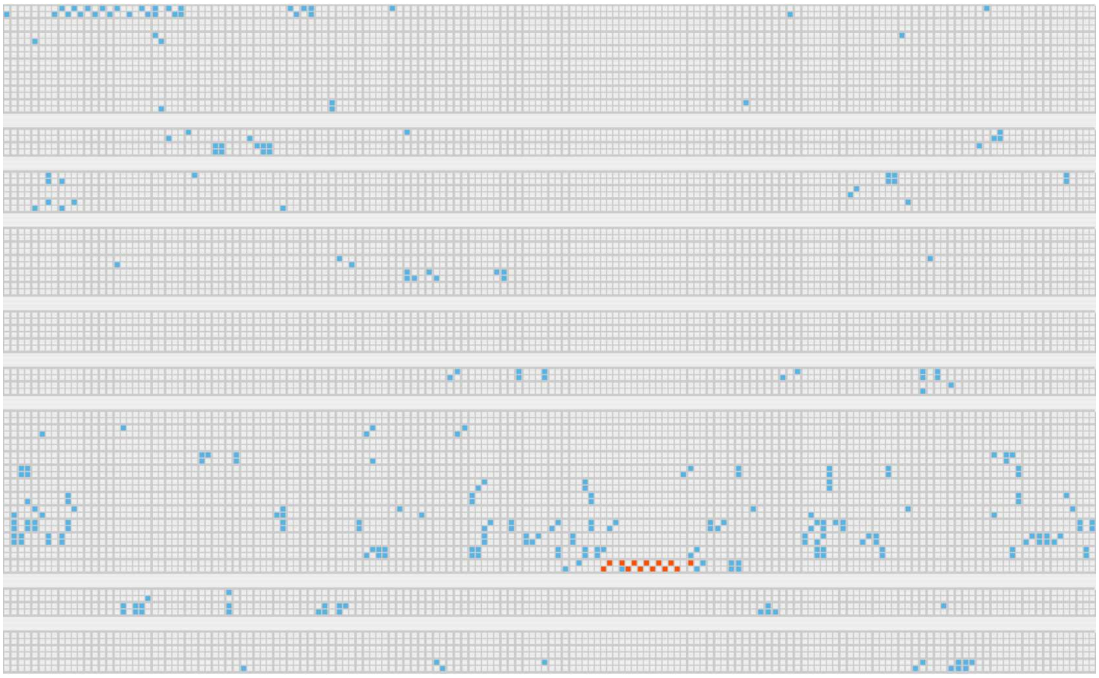


Figure 5.15: Visualisation of the repetitive toaster usage on the first day of the first week of the retiree data set shown using a one minute transaction interval. Triggered events are coloured in blue while events matching the emergent IARs are coloured in red.

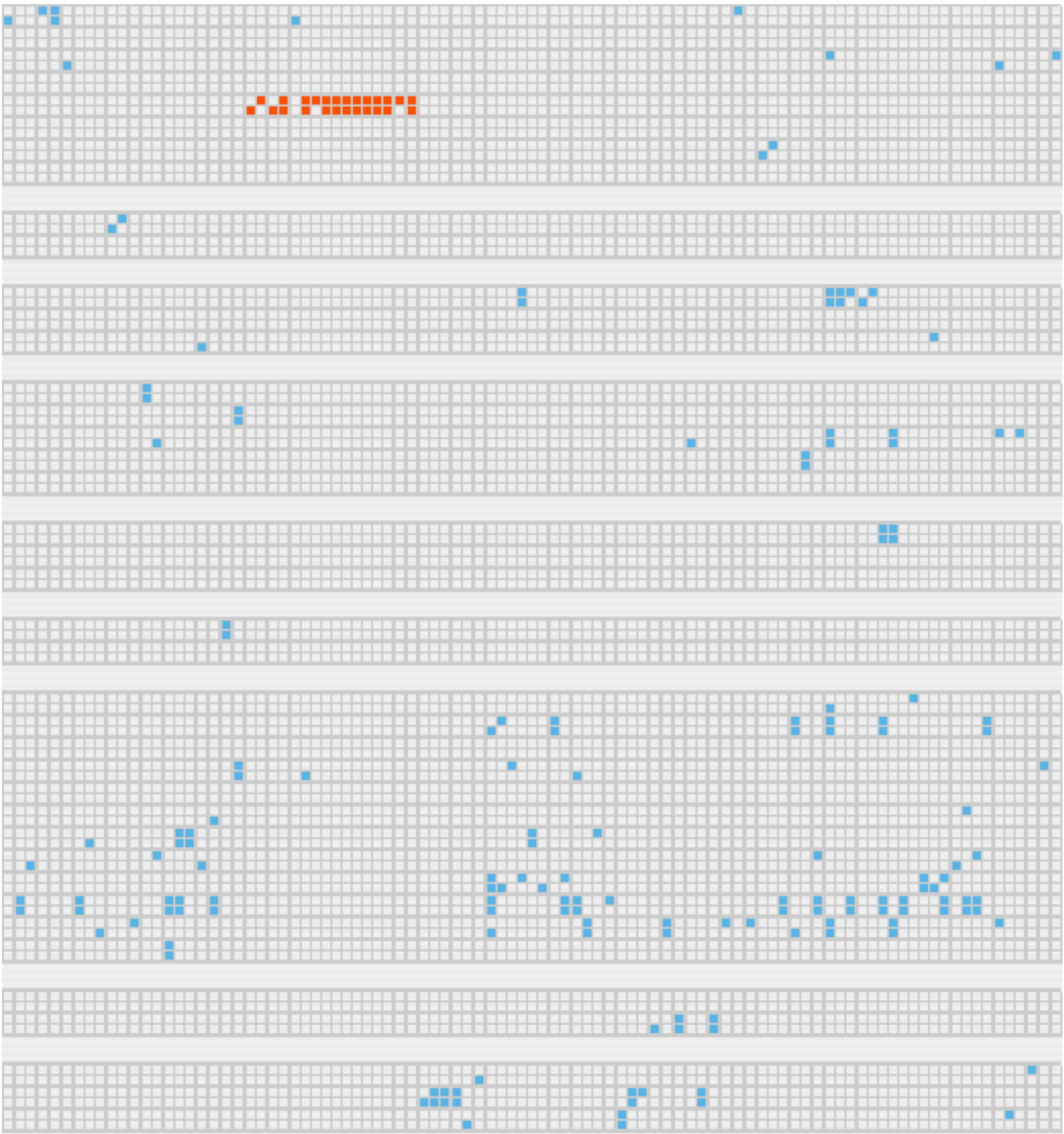


Figure 5.16: Visualisation of the malfunctioning shower faucet sensors on the first day of the second week of the retiree data set using a two minute transaction interval. Continuous toggling of the state of these sensors and their notable absence in the remainder of the data set reinforces the suspicion that a significant fault has been detected here. Triggered events are coloured in blue while events matching the emergent IARs are coloured in red.

over a forty minute period. Figure 5.17 shows the light in the pantry being switched on for a period of up to two minutes before being switched off again, a pattern that is repeated every one or two intervals. The use of the pantry in this way is unusual; the expected behaviour for this person is to have the pantry light on for a few minutes at a time and then to not return again for several hours. We do not believe that this new behaviour is abnormal, however.

The last emergent behaviour worth noting is an increase in the use of the drawer in the subject's home office. The drawer is used several times in short succession on days one and three of this week yet it is only used once in the previous week.

The last two sets of rules concerning the kitchen cabinet previously discussed in the second week of this subject in Section 4.3.3 are readily dismissible as kitchen activity related noise when viewed in their original context.

5.3.3 Automatic Selection of the Intertransaction Sliding Window Size Revisited

The heuristic previously introduced in Section 4.3.5 for automatically determining a suitable intertransaction sliding window size will, in this section, be revisited to investigate whether its application to the five intervals used in the visual examination of the emergent behaviour remains tenable. The number of rules discovered in both weeks of data from the working professional and retiree subject will be investigated as the window size is increased using the raw minimum support thresholds of $\alpha = 6$ and $\alpha = 8$.

The hypothesis remains that it is possible to determine an optimal sliding window size given an arbitrary data set, minimum support threshold and interval size by inspecting the number of new rules being mined as the window size is increased and weighing the benefit of any additional rules discovered with the additional computational cost of mining.

Working Professional Data Set

The number of rules mined in the first week of the working professional data set using the support threshold $\alpha = 6$ is presented in Figure 5.18. Here we see that the number of rules being returned using the one minute transaction interval tends to taper off at around $w = 35$. Although they have not completely tapered off within the period

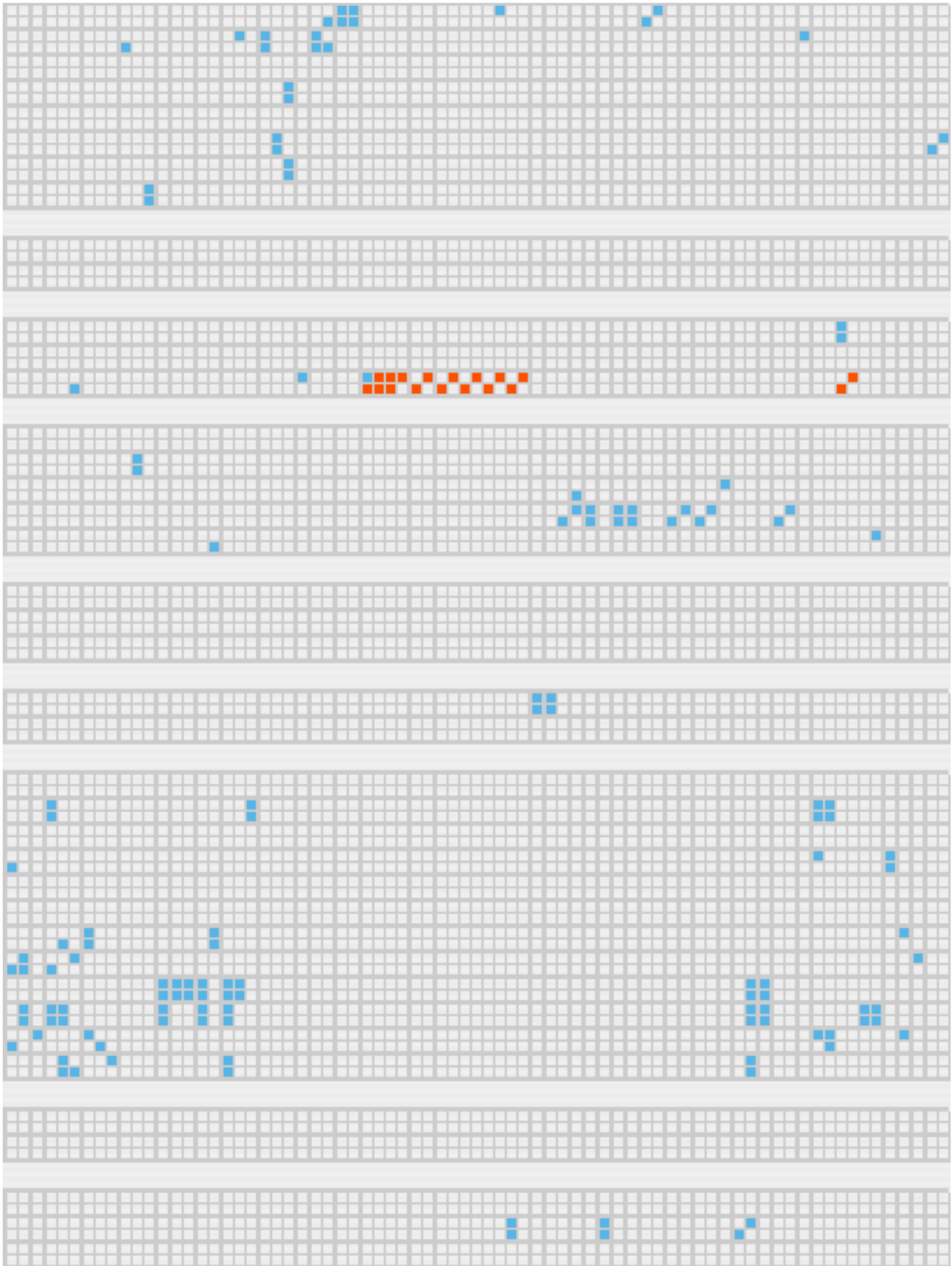


Figure 5.17: Visualisation of the unusual access of the butler's pantry on the seventh day of the second week of the retiree data set using a two minute transaction interval. Triggered events are coloured in blue while events matching the emergent IARs are coloured in red.

plotted, the number of new rules being discovered using the two minute interval appears to be declining from $w = 35$ onwards. The majority of the rules discovered using the one and two minute intervals are hence found within sliding windows spanning thirty-five and seventy minutes respectively. The three minute interval sees the number of rules returned falling off at around $w = 30$, a period of ninety minutes. Similarly, the four minute interval appears to taper off at $w = 27$, or 108 minutes, while the five minute interval tapers off at $w = 16$, or eighty minutes.

The period of time covered by the rules discovered in the first week using the higher support threshold of $\alpha = 8$ in Figure 5.19 is slightly reduced when compared to the lower support threshold. Rule discovery, as with the $\alpha = 6$ threshold, for the one minute interval continues to taper off at around $w = 35$ whereas the two minute interval now begins to taper off at around $w = 23$, or forty-six minutes. The coverage afforded by the three and four minute transaction intervals has likewise been slightly reduced. Here, suitable window sizes for the three and four minute interval are $w = 22$, or sixty-six minutes, and $w = 17$, or sixty-eight minutes. The rules returned with the five minute interval appear to fall off at around $w = 12$, a period of sixty minutes.

The results from the second week of data using the $\alpha = 6$ support threshold are shown in Figure 5.20. In contrast to the results from the first week, suitable window sizes for the one and two interval sizes appear to be relatively short. Rule discovery tapers off quickly when a window size of $w = 7$ is reached using the one minute interval and at $w = 5$ with the two minute interval. A suitable window size for the three minute interval appears at circa $w = 33$, spanning a period of just under 100 minutes. Similarly, the four and five minute intervals cover periods of approximately ninety-two and ninety-five minutes with $w = 23$ and $w = 19$ respectively.

Mining the second week of data with $\alpha = 8$ see similar results, shown in Figure 5.21, for the one and two minute intervals. Here, suitable intertransaction window sizes appear to be $w = 7$ for the one minute interval and $w = 5$ for the two minute interval. The three minute interval appears to only cover a fifteen minute span and tapers off at $w = 5$. The four and five minute intervals, however, continue to cover longer periods of time. The number of rules retrieved using the four minute interval tapers off at around $w = 18$, or seventy-two minutes, while the five minute interval falls off at around $w = 20$, a window spanning a 100 minute period.

We see from these results that increases in the granularity of the transaction intervals leads to an increase in both the number of rules being discovered and in the heuristically selected sliding window size. Larger transaction intervals increase the likelihood

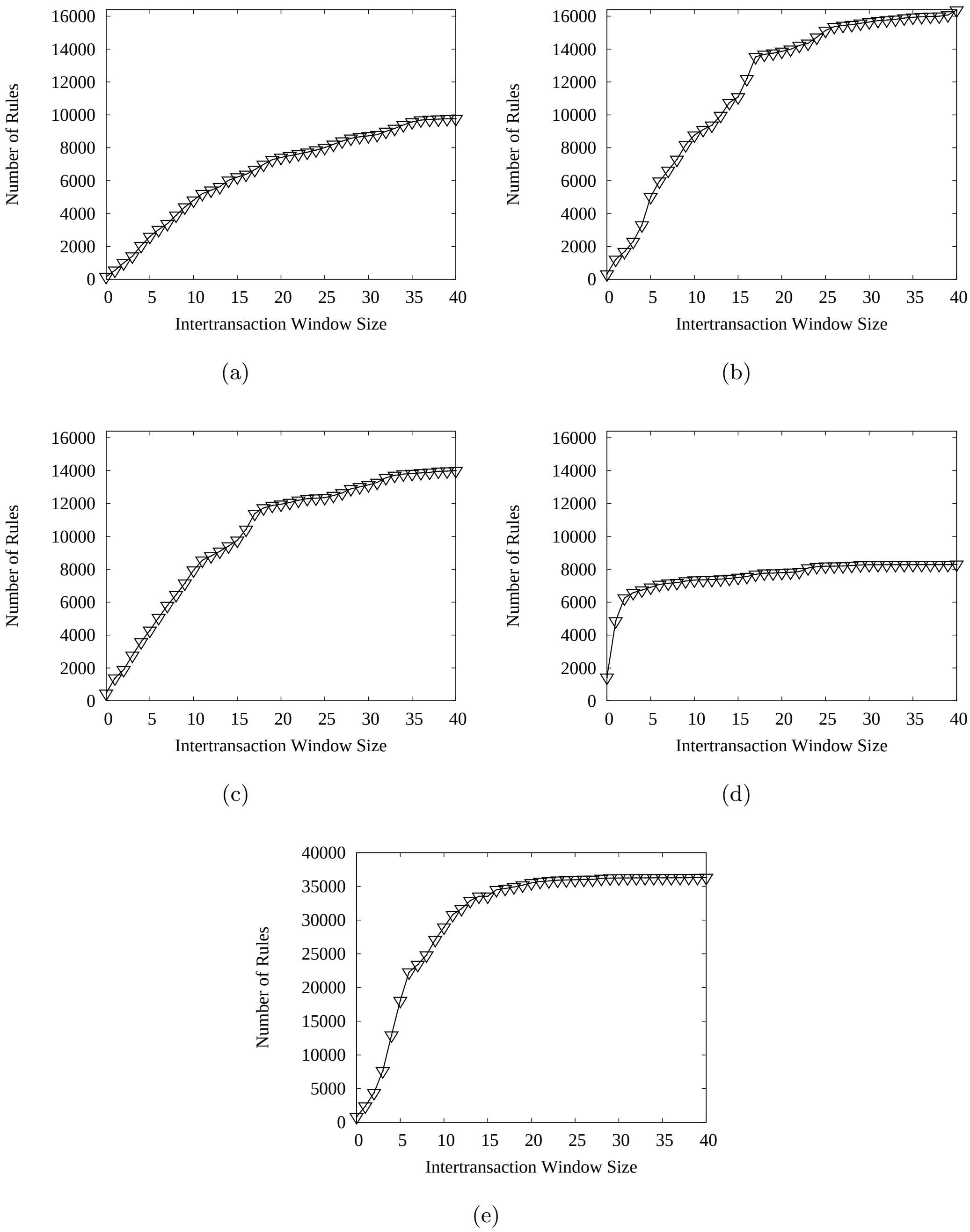
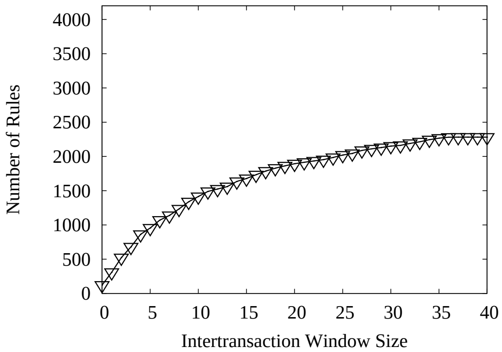
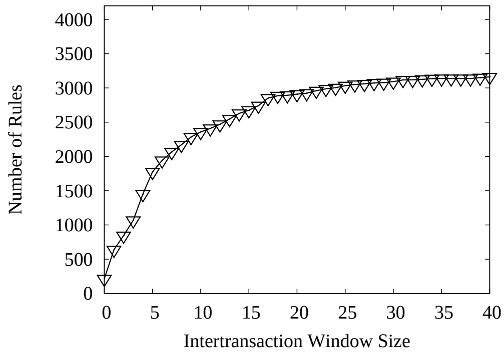


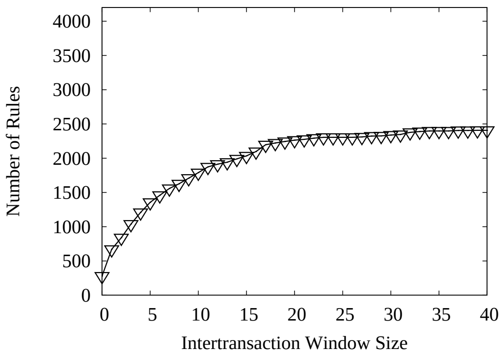
Figure 5.18: The number of rules discovered in the first week of the working professional data set using (a) one, (b) two, (c) three, (d) four and (e) five minute transaction intervals as the intertransaction sliding window size is increased. A raw minimum support threshold of $\alpha = 6$ was used.



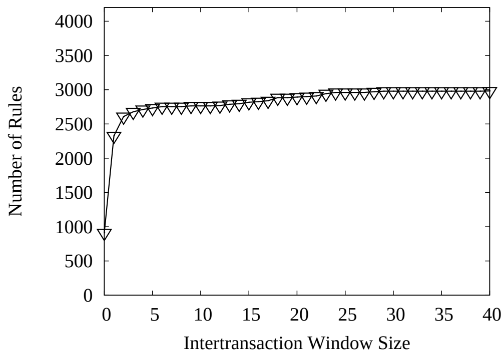
(a)



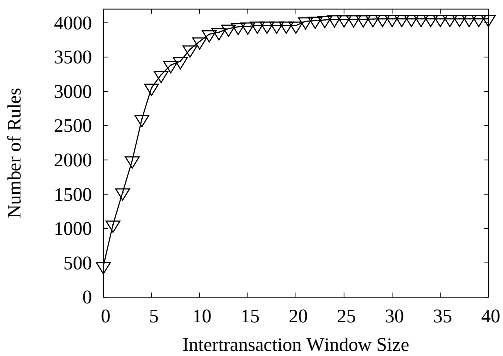
(b)



(c)



(d)



(e)

Figure 5.19: The number of rules discovered in the first week of the working professional data set using (a) one, (b) two, (c) three, (d) four and (e) five minute transaction intervals as the intertransaction sliding window size is increased. A raw minimum support threshold of $\alpha = 8$ was used.

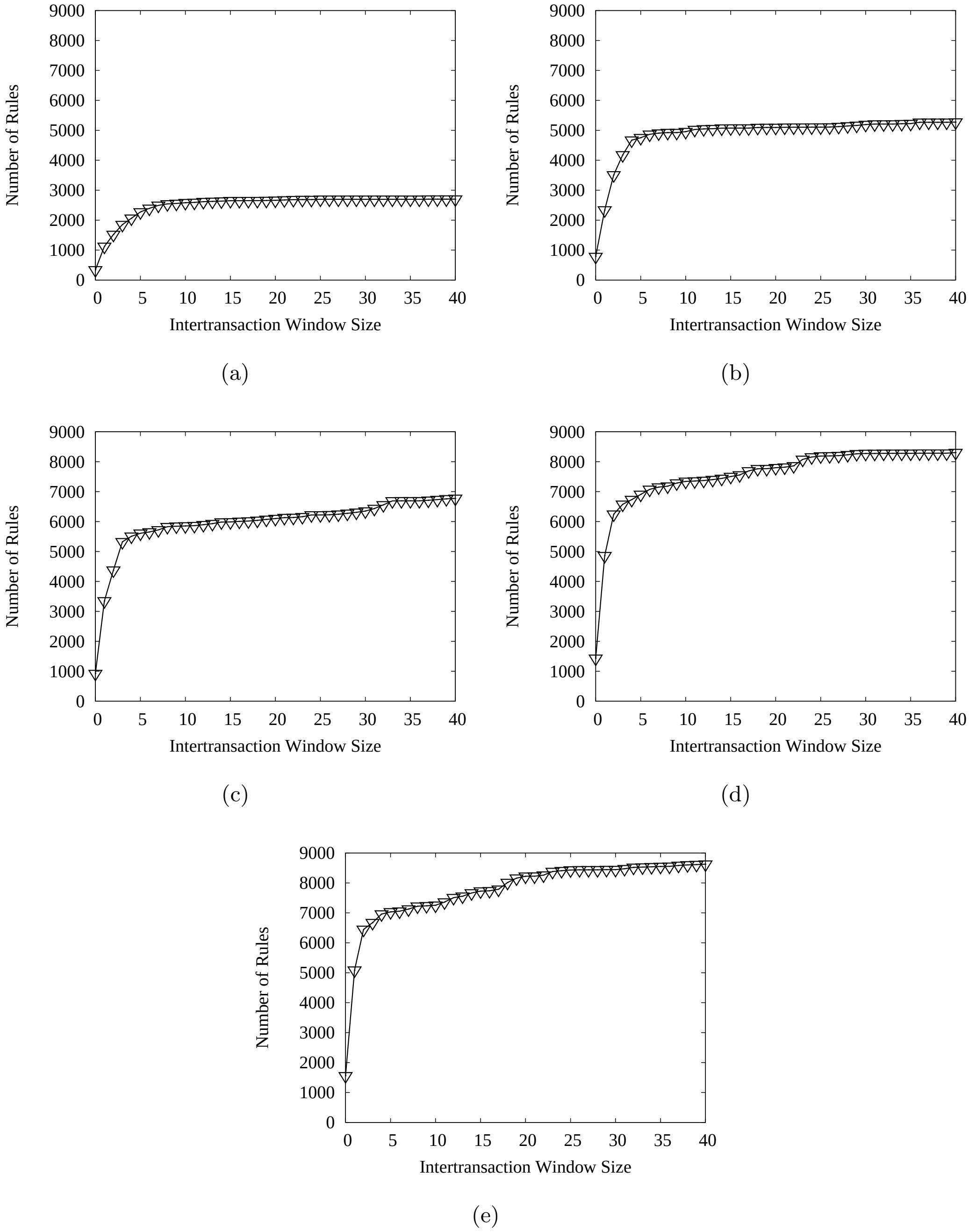


Figure 5.20: The number of rules discovered in the second week of the working professional data set using (a) one, (b) two, (c) three, (d) four and (e) five minute transaction intervals as the intertransaction sliding window size is increased. A raw minimum support threshold of $\alpha = 6$ was used.

of events, or items, being present in an interval and within the sliding window, directly resulting in an increase in the number of IARs that meet the minimum support threshold. The same effect is seen when the support level is reduced. The results also suggest that the heuristic remains a viable means of automating the selection of the length of the sliding window.

Retiree Data Set

Figure 5.22 shows the number of rules mined from the first week of the retiree data set as the sliding window size is increased using the $\alpha = 6$ support threshold. The number of new rules being discovered can be seen to clearly taper off for each of the five minute intervals. The one and two minute intervals are shown to taper off at $w = 8$ and $w = 12$ respectively while the coverage provided by the three minute interval is longer at $w = 7$, or twenty-one minutes. The four minute interval is shown to taper off at around $w = 10$ to cover a period of forty minutes. This same period of time is covered by the five minute interval which is seen to taper off at $w = 8$.

Mining the first week of data at the higher support threshold of $\alpha = 8$ again reveals an earlier tapering off than at the $\alpha = 6$ threshold. The heuristic continues to suggest a window size of around $w = 8$ for the one minute interval while the two minute interval tapers off after around fourteen minutes at $w = 7$. The period covered by the three, four and five minute intervals has been significantly reduced, however. The three minute interval now tapers off at circa $w = 5$, or fifteen minutes, while the number of new rules found using the four and five minute intervals falls off at circa $w = 6$, or eighteen minutes, and at $w = 5$, or twenty-five minutes, respectively.

Results from the second week of the retiree data set are presented in Figure 5.24. The appearance of an unusually steep jump in the number of rules being discovered at $w = 6$ for the one minute interval in Figure 5.24(a) is due to the many events recorded by the malfunctioning shower faucet sensors on the first day of data. The length and regularity with which these sensors are triggered causes all possible combinations of associations of these events within the bounds of the sliding window size to be mined. This effect is mitigated as the intervals become more granular and the number of possible combinations is reduced. Figure 5.25 shows the number of rules being mined from the second week of data with the malfunctioning sensors removed. The number of rules being discovered for the one minute interval in Figure 5.25(a) is seen to have returned to a steady decline in growth and the optimal window size has shifted from $w = 8$ to $w = 14$.

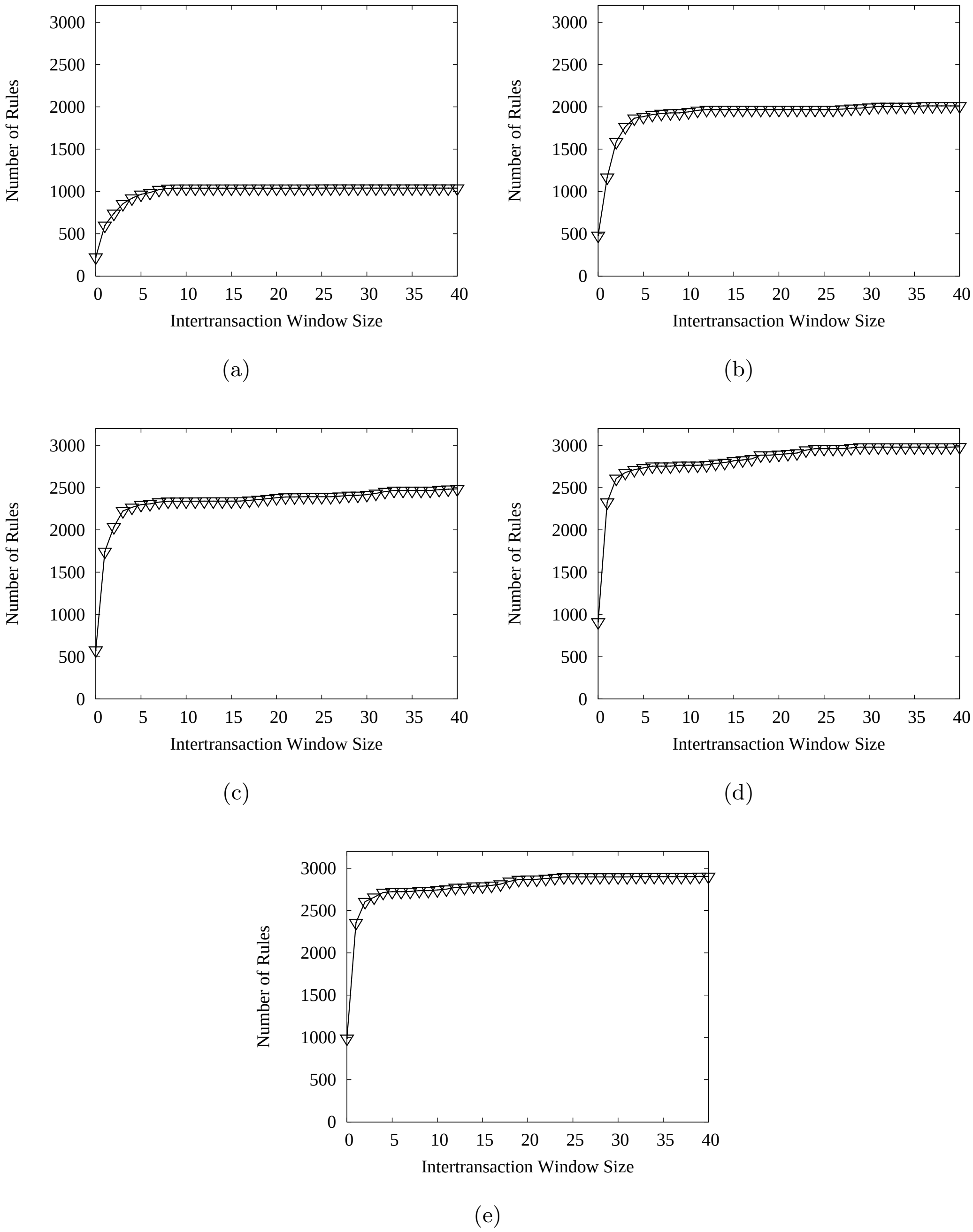
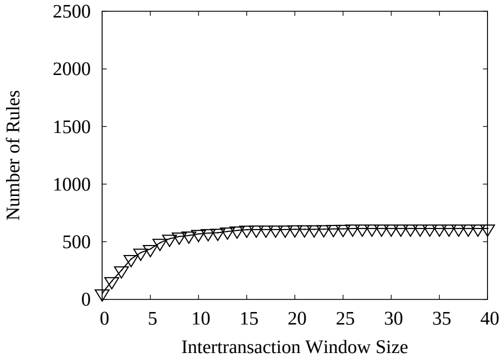
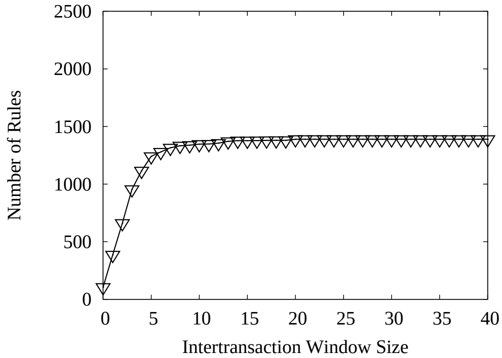


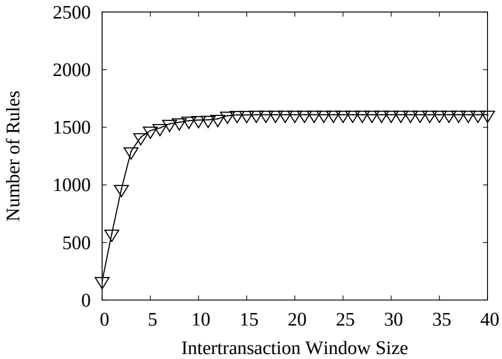
Figure 5.21: The number of rules discovered in the second week of the working professional data set using (a) one, (b) two, (c) three, (d) four and (e) five minute transaction intervals as the intertransaction sliding window size is increased. A raw minimum support threshold of $\alpha = 8$ was used.



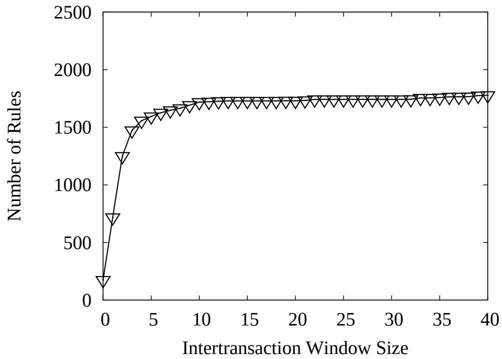
(a)



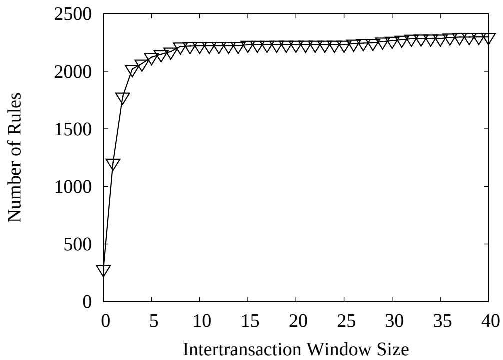
(b)



(c)



(d)



(e)

Figure 5.22: The number of rules discovered in the first week of the retiree data set using (a) one, (b) two, (c) three, (d) four and (e) five minute transaction intervals as the intertransaction sliding window size is increased. A raw minimum support threshold of $\alpha = 6$ was used.

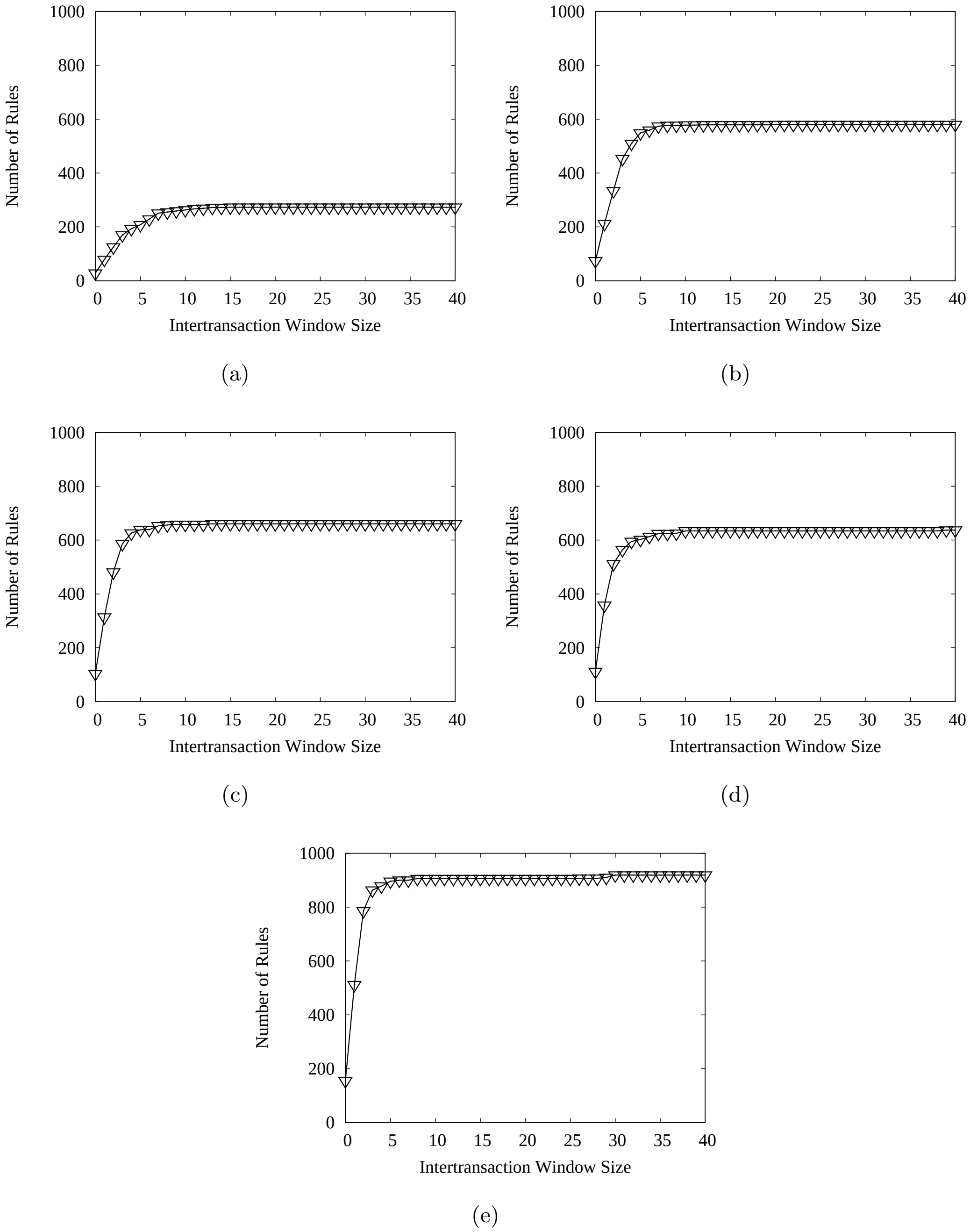


Figure 5.23: The number of rules discovered in the first week of the retiree data set using (a) one, (b) two, (c) three, (d) four and (e) five minute transaction intervals as the intertransaction sliding window size is increased. A raw minimum support threshold of $\alpha = 8$ was used.

The remainder of the intervals in both Figure 5.24 and in Figure 5.25 are seen to have similar curves, the data set with the malfunctioning sensor removed exhibiting only a marginal decrease in the number of rules found. The two minute interval is seen to taper off at $w = 16$ for a window that spans thirty-two minutes while the growth in the number of new rules found for the three, four and five minute intervals each appear to fall off at $w = 10$.

Similar results are seen with $\alpha = 8$ for the second week of data in Figure 5.26. Here also, the anomalous shower faucet sensors cause a significant jump in the number of rules mined at the one minute interval. The exponential growth seen in Figure 5.26(a) appears at $w = 4$ compared to $w = 6$ at the lower support threshold. Figure 5.27 shows the result of increasing the window length when the shower faucet sensors have been removed. The curve in Figure 5.27(a) is seen to behave normally. The number of new rules in Figure 5.27(a) is seen to taper off at $w = 10$ while those in Figure 5.26(a) taper off at $w = 7$.

The curves for the two, three, four and five minute intervals taper off at the same times in both Figure 5.26 and for Figure 5.27. Although the number of rules being found here are lower than with the $\alpha = 6$ support threshold, only the two minute interval has shown a significant decrease in the period covered by tapering off at $w = 6$. The remainder of the curves appear to fall off at $w = 10$, $w = 9$ and at $w = 8$ for each of the three, four and five minute intervals respectively.

The results from the retiree subject data set show that both an increase in the granularity of the transaction intervals and a reduction of the minimum support threshold continues to result in an increase in the length of the sliding window and in the number of rules being mined. These results also again confirm the viability of the heuristic as a technique for determining the size of the sliding window.

5.4 Conclusion

This chapter has introduced a novel visual data mining tool for exploring and analysing emergent intertransaction associations in their original data space. The proposed interface provides users with a grid representation showing the triggered sensors events on a time line view. Users may select whether the grid is presented in a compact form, where only intervals in which activity is recorded are displayed, or in its full form. Emergent IARs are displayed in a table that can be ordered by the rule support measure, rule growth or by rule similarity. Occurrences of rules that are selected in the table are

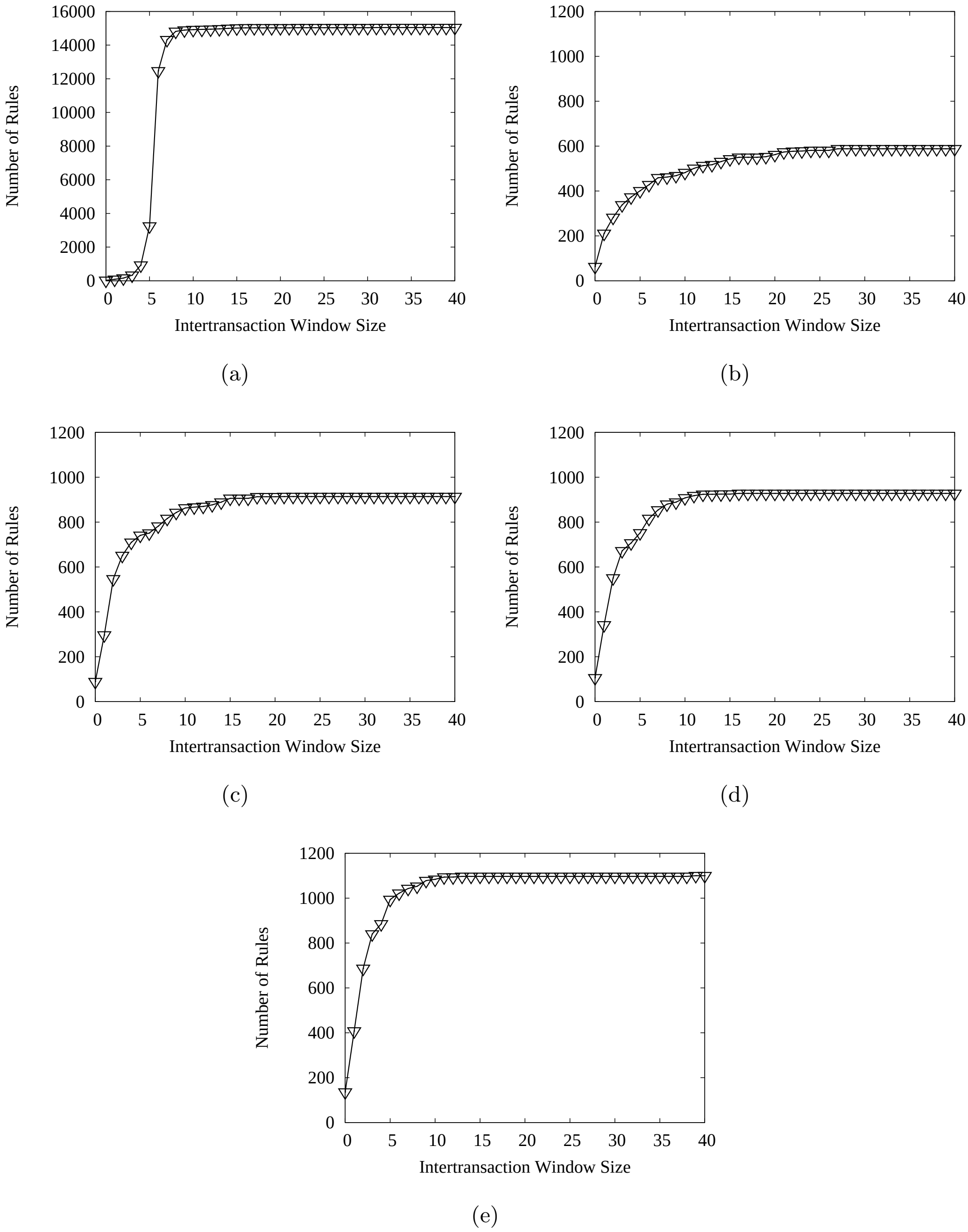
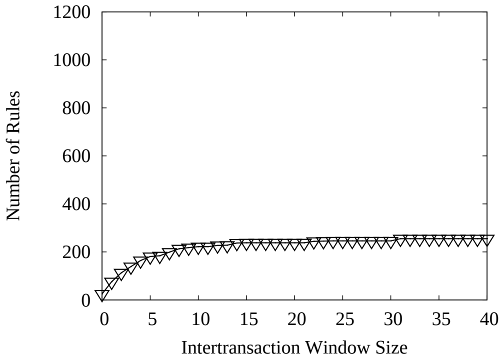
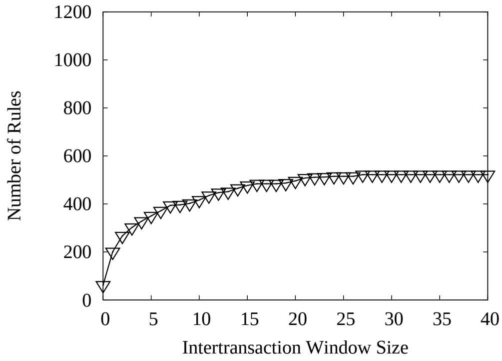


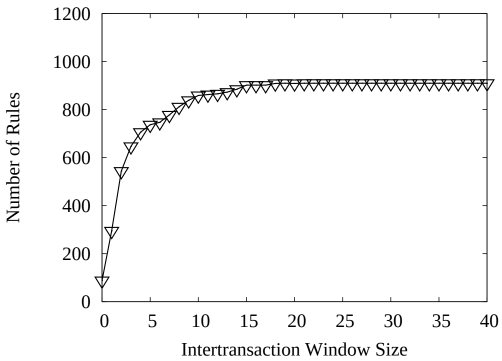
Figure 5.24: The number of rules discovered in the second week of the retiree data set using (a) one, (b) two, (c) three, (d) four and (e) five minute transaction intervals as the intertransaction sliding window size is increased. A raw minimum support threshold of $\alpha = 6$ was used.



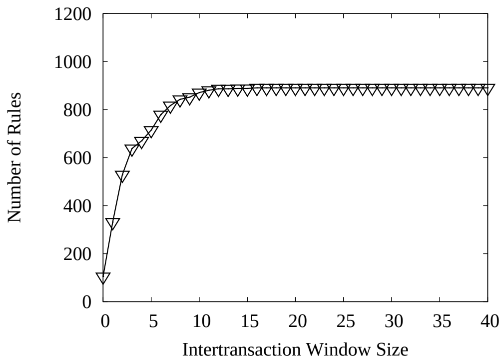
(a)



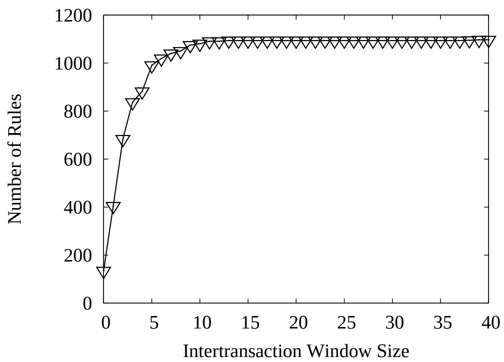
(b)



(c)



(d)



(e)

Figure 5.25: The number of rules discovered in the second week of the retiree data set with the malfunctioning shower faucet sensor events removed using (a) one, (b) two, (c) three, (d) four and (e) five minute transaction intervals as the intertransaction sliding window size is increased. A raw minimum support threshold of $\alpha = 6$ was used.

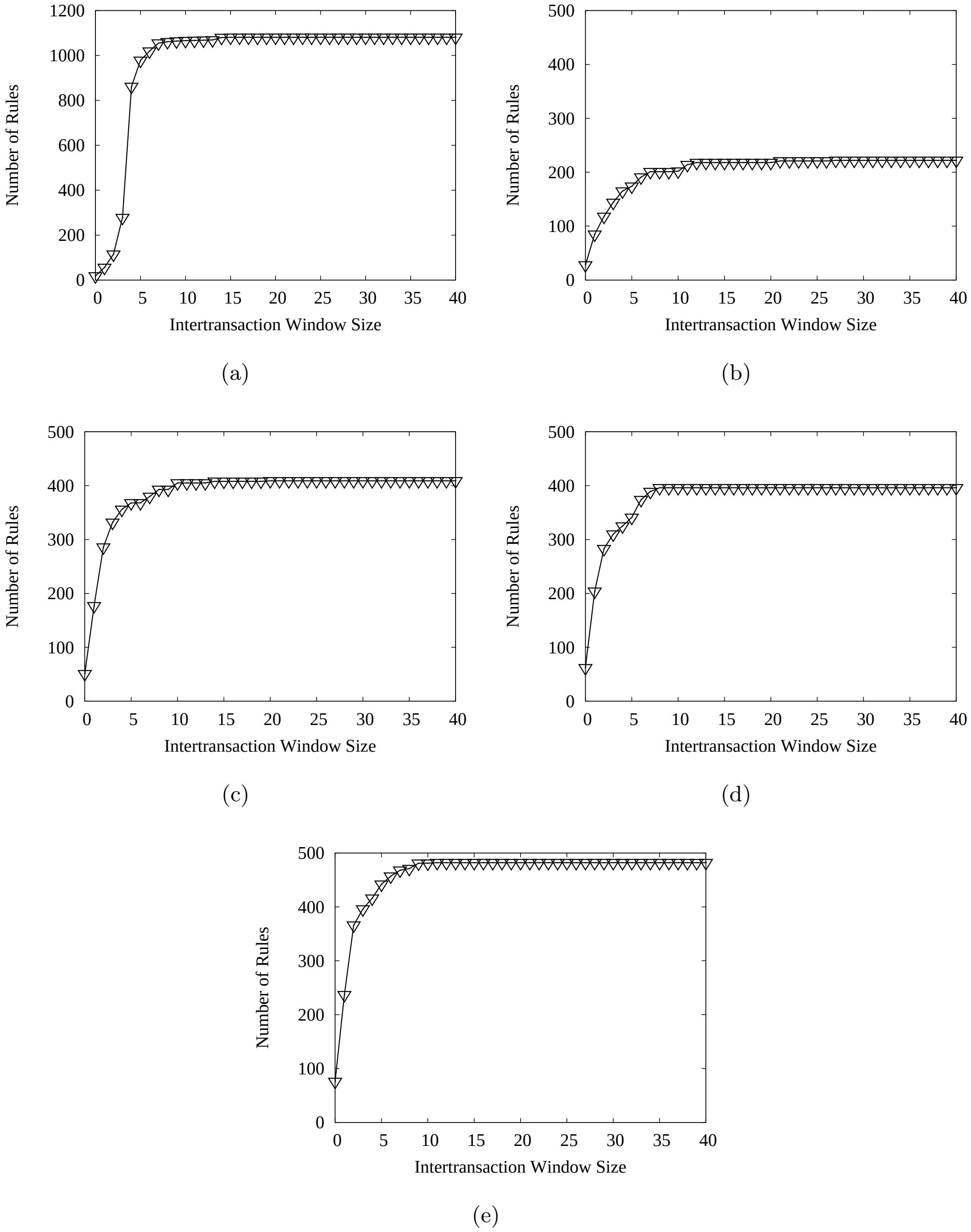


Figure 5.26: The number of rules discovered in the second week of the retiree data set using (a) one, (b) two, (c) three, (d) four and (e) five minute transaction intervals as the intertransaction sliding window size is increased. A raw minimum support threshold of $\alpha = 8$ was used.

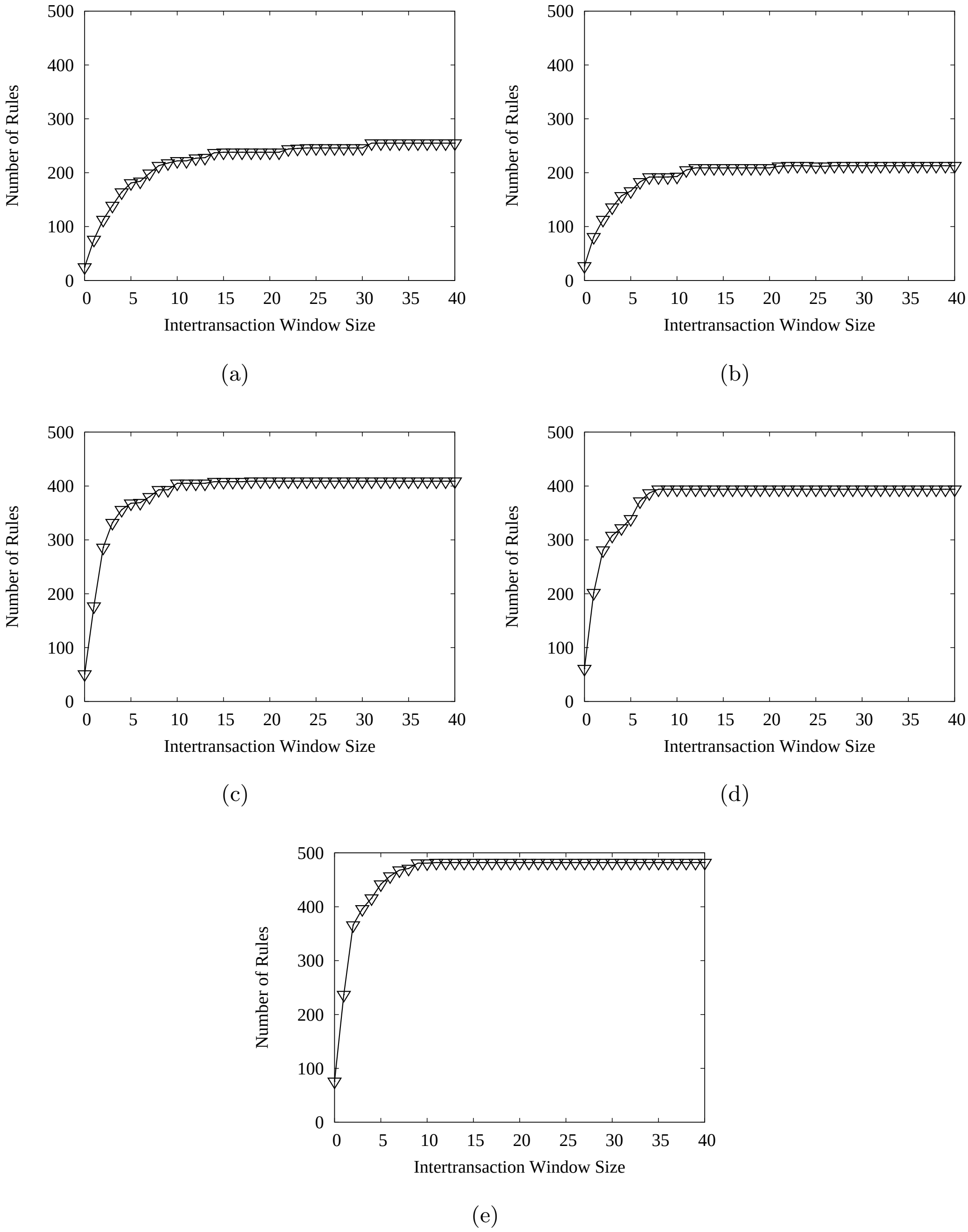


Figure 5.27: The number of rules discovered in the second week of the retiree data set with the malfunctioning shower sensor removed using (a) one, (b) two, (c) three, (d) four and (e) five minute transaction intervals as the intertransaction sliding window size is increased. A raw minimum support threshold of $\alpha = 8$ was used.

prominently highlighted on the grid view. It was shown that the design rationale is the logical conclusion to the problem of visualising intertransaction associations in the original data space.

Repeat experimentation on the real world data sets described in Chapter 4 revealed that use of the interface provides a more intuitive, efficient and user friendly approach to investigating emergent IARs than through manual inspection of the event log data. The majority of the observations made on the real world data via the interface were found to concur with the previous observations in Section 4.3.3 and Section 4.3.4 with two notable additions.

It was revealed that the full extent of an anomalous pattern on the first day of data from the first week of the retiree subject was previously not apparent. Through the interface, it could be seen that the frequency, duration and time of day of the elderly subject making short visits to the bathroom was highly abnormal and would warrant further investigation. The interface also provided additional insight by confirming that a suspected sensor malfunction in the second week of data from the retiree subject was indeed a technical glitch by revealing that the sensor remained inactive for the remainder of the week.

The optimal intertransaction sliding window size heuristic was revisited. Investigation revealed that the heuristic remains a feasible technique for automatically selecting a suitable sliding window size given an arbitrary data set, a minimum support threshold and interval size.

Chapter 6

Conclusion

This thesis has presented techniques for the detection of abnormality in human behaviour. Two different approaches originating from the areas of stochastic models and data mining were investigated.

The application of stochastic models for learning patterns of normality with which new observational data gathered by a visual tracking system could be examined was presented in Chapter 3. Investigation into the use of the Hierarchical Hidden Markov Model (HHMM) showed that the model could be applied to the training of multi-level models of human behaviour. Experimentation demonstrated that models trained on subpatterns of behaviour were able to learn state transitions whose structure closely resembles a hierarchical decomposition of the activities present in the training data. Further experimentation demonstrated that the HHMM could also be employed as a classifier of normal activities. The resulting models could, therefore, be reused in the training of higher level models that are able to encompass longer term human behaviours.

Next, an investigation into the importance of incorporating duration into models of human behaviour revealed that the Hidden Markov Model (HMM) and, by extension, the HHMM were unable to be used as reliable classifiers of normal behaviour or in the detection of abnormality when the ordering of activity within observation sequences remained the same yet differed in the duration of the activities. A comparison between the HMM, the left-right constrained HMM, the ESD-HMM and the left-right constrained ESD-HMM showed that only the latter model could be reliably applied as both a classifier of normal behaviour and be used to detect longer term durational abnormality. An extension to the ESD-HMM in which the state durations are known was then introduced. The new model, dubbed the Observed Time Indices ESD-HMM (OTI ESD-HMM), uses an observation signal that has been augmented with pressure mat sensor information to exploit known state transition times in model training and inferencing. Experimentation revealed that the classification and abnormality detection performance of the OTI ESD-HMM was comparable with that of the standard ESD-HMM with the added benefit of being able to encode sequences of activity such that the observation signal could be faithfully reproduced when a generative process is applied to the trained model.

Finally, a comparison between the HMM and the OTI ESD-HMM was made using data obtained by deploying the visual tracking system into the home of a volunteer subject. Similar results between the models were documented for both the classification and abnormality detection tasks. The OTI ESD-HMM demonstrated improved sensitivity to abnormality compared to the HMM. Its advantage was diminished, however, by increased susceptibility to noise; the harsh lighting conditions of the environment making the tracking of the subject difficult. It was demonstrated that the OTI ESD-HMM was able to reliably detect long term durational anomalies when the duration of an otherwise normal activity sequence was artificially increased. The HMM was unable to detect this type of abnormality.

The data mining component of the thesis began in Chapter 4. A novel application of Intertransaction Association Rule (IAR) mining that enables emergent behaviours to be discovered from state change sensor event logs was introduced. Emergent behaviours are those that are new or occur with unusual frequency given some historical data set. Extended Frequent Pattern Growth (EFP-Growth), a new algorithm for IAR mining, and its accompanying Extended Frequent Pattern Tree (EFP-Tree) data structure were introduced as extensions to the Frequent Pattern Growth (FP-Growth) algorithm and the Frequent Pattern Tree (FP-Tree) structure. EFP-Growth allows IARs to be efficiently mined by avoiding the computationally expensive candidate generation-then-test approach employed in existing IAR mining algorithms. Experimental results on both synthetic and real world data showed EFP-Growth outperforming the First Intra Then Inter (FITI) algorithm with an order of magnitude improvement in computational complexity. This was especially the case on the synthetic dense data sets where a high number of frequent items rendered FITI unable to discard unnecessary data early on in the mining process. Analysis of the emergent IARs found in the real world data sets from two test subjects (people) showed that new and changing behaviours could be detected with several examples of sensor malfunctions being identified.

Chapter 5 continued the work into analysis of emergent behaviours with the introduction of a visual data mining tool for exploring emergent IARs. The visual interface is unique in that it is the first such tool for IAR analysis in that it maps discovered IARs back onto the original data space. This allows users to visually investigate the significance and meaning of discovered rules in the context of the sensor event data without the need for laborious trawling through the event logs. Repeat experimentation showed that the behaviours identified in previous experimentation in Chapter 4 could be identified more easily and in less time than is possible through manual inspection of the sensor event logs. New insight was gained into the importance of one particular emergent IAR with the discovery of a behaviour that would warrant investigation into the health of an elderly subject by outside carers.

6.1 Future Directions

Several limiting factors were observed in the investigation of the suitability of the HHMM as a tool for learning hierarchical models of human behaviour. These were: cubic complexity with respect to the length of the observation sequence, numerical underflow constraints due to lack of scaling in implementations of the model and a lack of explicit duration modelling. The issue of numerical underflow has since been resolved in Phung (2005) while a special case of a two layer HHMM in which duration modelling is incorporated has been presented with the introduction of the Switching Hidden Semi-Markov Model (S-HSMM) (Duong et al., 2005). This invites future work into a general case HHMM in which duration is modelled at every layer in the hierarchy. The issue of computational complexity may be resolved by representing the resulting model as a Dynamic Bayesian Network (DBN) and employing the methodology used in Murphy and Paskin (2001) to trade a reduction in computational complexity with an increase in memory usage. Such work may lead to complex hierarchical models of human behaviour that suffer none of the limitations of the original HHMM while retaining the ability to properly accommodate duration.

Opportunity for further work also exists by building upon the work undertaken in emergent behaviour mining. To wit; the benefit of intertransaction association rules lies in their ability to express event associations within and between fixed time intervals. However, discrete time intervals impose an artificial segmentation of time. A more expressive semantic for representing relations based on the interval in which events, actions or activities occur may be offered by Allen's temporal interval logic (Allen, 1983). This allows relations such as "before", "during", "in between" and "after" to describe the temporal relationships between observed activities or states of objects.

Höppner and Klawonn (2002) have, for example, used this logic to extend frequent episode mining to discover more expressive rule relationships. The relative nature of such temporal relationships provides flexibility yet also implies imprecision; no quantitative feature is assigned to relations so a rule "turns light out in between getting into bed after showering" may be interpreted differently than "turns light out in between getting into bed approximately ten hours after showering". In the former, it is implied that a subject has gone to bed shortly after showering in the evening while the latter implies the subject has gone to bed in the evening after showering in the morning. A hybrid technique that incorporates Allen's temporal interval logic with intertransaction mining may be of use here.

The direction of such work is closely aligned to multigranular temporal mining which has seen an extension of frequent episode mining that introduced temporal constraints to relationship between events (Bettini et al., 1998a,b). The granularity of these constraints can be arbitrarily specified by a user given knowledge about the problem domain. Given an array of feature rich sensors, it may be possible to apply or extend these methods to discover rules which capture richer relationships between events and activities than is currently offered by IAR mining.

Appendix A

Hierarchical HMM State Transition and Emission Likelihoods

For completeness, this appendix lists the learned state transition likelihoods and emission probabilities of the HHMM models depicted in Section 3.1.2.

A.1 Kitchen Models

A.1.1 "Food Preparation First"

The state transition likelihoods and the emission probabilities for the "food preparation first" class model in Figure 3.3 are presented here in Tables A.1–A.11.

Table A.1: Initial state likelihoods stored at the root node in the "food preparation first" model.

Destination		
q_1^2	q_2^2	q_3^2
0	0	1

Table A.2: State transition likelihoods governing the second layer of the "food preparation first" model stored at the root node.

Source	Destination			
	q_1^2	q_2^2	q_3^2	q_{end}^2
q_1^2	0.0000	0.0000	1.0000	0.0000
q_2^2	1.0000	0.0000	0.0000	0.0000
q_3^2	0.0000	0.5000	0.0000	0.5000

Table A.3: Initial state likelihoods for the "cooking" submodel in the "food preparation first" model stored at state q_1^2.

Destination					
q_1^3	q_2^3	q_3^3	q_4^3	q_5^3	q_6^3
0.0889	0.0000	0.0000	0.0000	0.0000	0.9111

Table A.4: State transition likelihoods governing the production state children of the "cooking" submodel in the "food preparation first" model stored at state q_1^2.

Source	Destination						
	q_1^3	q_2^3	q_3^3	q_4^3	q_5^3	q_6^3	q_{end}^3
q_1^3	0.5342	0.0000	0.0000	0.4658	0.0000	0.0000	0.0000
q_2^3	0.0000	0.3374	0.6617	0.0001	0.0008	0.0000	0.0000
q_3^3	0.0000	0.1462	0.2316	0.0000	0.6402	0.0000	0.0000
q_4^3	0.0000	0.4654	0.0008	0.5338	0.0000	0.0000	0.0000
q_5^3	0.0000	0.0000	0.0000	0.0000	0.5342	0.0000	0.4658
q_6^3	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

Table A.5: Emission likelihoods of the production state children of the "cooking" submodel in the "food preparation first" model stored at the children states of q_1^2.

State	Observation					
	Stove	Food Prep	Sink	Undefined	Fridge	Door
q_1^3	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000
q_2^3	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000
q_3^3	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000
q_4^3	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000
q_5^3	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000
q_6^3	0.0000	0.0091	0.0000	0.9909	0.0000	0.0000

Table A.6: Initial state likelihoods for the "fridge & food prep" submodel in the "food preparation first" model stored at state q_2^2.

Destination					
q_1^3	q_2^3	q_3^3	q_4^3	q_5^3	q_6^3
0.0654	0.0000	0.0000	0.0000	0.9346	0.0000

Table A.7: State transition likelihoods governing the production state children of the "fridge & food prep" submodel in the "food preparation first" model stored at state q_2^2.

Source	Destination						
	q_1^3	q_2^3	q_3^3	q_4^3	q_5^3	q_6^3	q_{end}^3
q_1^3	0.7330	0.0000	0.0000	0.0000	0.2413	0.0257	0.0000
q_2^3	0.0619	0.0283	0.0000	0.0995	0.0000	0.0000	0.8104
q_3^3	0.0970	0.0000	0.2608	0.0318	0.0000	0.6104	0.0000
q_4^3	0.0000	0.0000	0.4476	0.5407	0.0000	0.0000	0.0116
q_5^3	0.0283	0.1419	0.0000	0.7579	0.0719	0.0000	0.0000
q_6^3	0.3568	0.1649	0.3001	0.0004	0.0000	0.0019	0.1758

Table A.8: Emission likelihoods of the production state children of the "fridge & food prep" submodel in the "food preparation first" model stored at the children states of q_2^2.

State	Observation					
	Stove	Food Prep	Sink	Undefined	Fridge	Door
q_1^3	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000
q_2^3	0.0000	0.0000	0.1544	0.8456	0.0000	0.0000
q_3^3	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000
q_4^3	0.0000	0.9996	0.0000	0.0004	0.0000	0.0000
q_5^3	0.0000	0.0000	0.0000	0.0300	0.9042	0.0658
q_6^3	0.0000	0.9023	0.0274	0.0000	0.0429	0.0274

Table A.9: Initial state likelihoods for the "enter/exit" submodel in the "food preparation first" model stored at state q_3^2.

Destination					
q_1^3	q_2^3	q_3^3	q_4^3	q_5^3	q_6^3
0.0000	0.0991	0.0641	0.0000	0.5566	0.2803

Table A.10: State transition likelihoods governing the production state children of the "enter/exit" submodel in the "food preparation first" model stored at state q_3^2.

Source	Destination						
	q_1^3	q_2^3	q_3^3	q_4^3	q_5^3	q_6^3	q_{end}^3
q_1^3	0.0618	0.0000	0.0000	0.9382	0.0000	0.0000	0.0000
q_2^3	0.0000	0.0000	0.5255	0.0000	0.4745	0.0000	0.0000
q_3^3	0.9555	0.0000	0.0445	0.0000	0.0000	0.0000	0.0000
q_4^3	0.0000	0.0000	0.0000	0.0303	0.0000	0.0000	0.9697
q_5^3	0.1058	0.0000	0.8911	0.0000	0.0031	0.0000	0.0000
q_6^3	0.0000	0.0000	0.5255	0.0000	0.4745	0.0000	0.0000

Table A.11: Emission likelihoods of the production state children of the "enter/exit" submodel in the "food preparation first" model stored at the children states of q_3^2.

State	Observation					
	Stove	Food Prep	Sink	Undefined	Fridge	Door
q_1^3	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000
q_2^3	0.0000	0.0000	0.0000	0.1090	0.8910	0.0000
q_3^3	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000
q_4^3	0.0692	0.0000	0.0000	0.0000	0.0000	0.9308
q_5^3	0.0000	0.0000	0.0000	0.2990	0.0000	0.9701
q_6^3	0.0000	0.0000	0.0000	0.1682	0.8318	0.0000

A.1.2 "Washing Dishes First"

The state transition likelihoods and the emission probabilities for the "washing dishes first" class model in Figure 3.4 are presented here in Tables A.12–A.22.

Table A.12: Initial state likelihoods stored at the root node in the "washing dishes first" model.

Destination		
q_1^2	q_2^2	q_3^2
0.0000	1.0000	0.0000

Table A.13: State transition likelihoods governing the second layer of the "washing dishes first" model stored at the root node.

Source	Destination			
	q_1^2	q_2^2	q_3^2	q_{end}^2
q_1^2	0.0000	0.0000	1.0000	0.0000
q_2^2	0.5000	0.0000	0.0000	0.5000
q_3^2	0.0000	1.0000	0.0000	0.0000

Table A.14: Initial state likelihoods for the "cooking" submodel in the "washing dishes first" model stored at the children states of q_1^2.

Destination					
q_1^3	q_2^3	q_3^3	q_4^3	q_5^3	q_6^3
0.0000	0.0000	1.0000	0.0000	0.0000	0.0000

Table A.15: State transition likelihoods governing the production state children of the "cooking" submodel in the "washing dishes first" model stored at state q_1^2.

Source	Destination						
	q_1^3	q_2^3	q_3^3	q_4^3	q_5^3	q_6^3	q_{end}^3
q_1^3	0.4651	0.3421	0.0000	0.1667	0.0261	0.0000	0.0000
q_2^3	0.0007	0.4743	0.0000	0.0000	0.5250	0.0000	0.0000
q_3^3	0.0000	0.0000	0.0713	0.0010	0.0000	0.9277	0.0000
q_4^3	0.4031	0.0320	0.0000	0.5649	0.0000	0.0000	0.0000
q_5^3	0.0000	0.1164	0.0000	0.0000	0.5708	0.0000	0.3128
q_6^3	0.0000	0.0000	0.0000	0.9536	0.0000	0.0464	0.0000

Table A.16: Emission likelihoods of the production state children of the "cooking" submodel in the "washing dishes first" model stored at the children states of q_1^2.

State	Observation					
	Stove	Food Prep	Sink	Undefined	Fridge	Door
q_1^3	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000
q_2^3	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000
q_3^3	0.0000	0.1000	0.0000	0.2632	0.4703	0.1663
q_4^3	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000
q_5^3	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000
q_6^3	0.0000	0.0000	0.3383	0.6617	0.0000	0.0000

Table A.17: Initial state likelihoods for the "wash dishes" submodel in the "washing dishes first" model stored at state q_2^2.

Destination					
q_1^3	q_2^3	q_3^3	q_4^3	q_5^3	q_6^3
0.0000	0.2500	0.7500	0.0000	0.0000	0.0000

Table A.18: State transition likelihoods governing the production state children of the "wash dishes" submodel in the "washing dishes first" model stored at state q_2^2.

Source	Destination						
	q_1^3	q_2^3	q_3^3	q_4^3	q_5^3	q_6^3	q_{end}^3
q_1^3	0.0045	0.0000	0.0000	0.0065	0.0000	0.0000	0.9890
q_2^3	0.0475	0.0000	0.7328	0.0000	0.1790	0.0407	0.0000
q_3^3	0.0001	0.0000	0.1613	0.0001	0.8362	0.0023	0.0000
q_4^3	0.4180	0.0000	0.0000	0.0429	0.0000	0.0000	0.5391
q_5^3	0.1459	0.0000	0.0000	0.6147	0.1879	0.0000	0.0515
q_6^3	0.0911	0.0000	0.0000	0.0227	0.7321	0.0777	0.0764

Table A.19: Emission likelihoods of the production state children of the "wash dishes" submodel in the "washing dishes first" model stored at the children states of q_2^2.

State	Observation					
	Stove	Food Prep	Sink	Undefined	Fridge	Door
q_1^3	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000
q_2^3	0.0000	0.0000	0.0000	0.2857	0.7143	0.0000
q_3^3	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000
q_4^3	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000
q_5^3	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000
q_6^3	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000

Table A.20: Initial state likelihoods for the "enter/exit" submodel in the "washing dishes first" model stored at state q_3^2.

Destination					
q_1^3	q_2^3	q_3^3	q_4^3	q_5^3	q_6^3
0.0000	1.0000	0.0000	0.0000	0.0000	0.0000

Table A.21: State transition likelihoods governing the production state children of the "enter/exit" submodel in the "washing dishes first" model stored at state q_3^2.

Source	Destination						
	q_1^3	q_2^3	q_3^3	q_4^3	q_5^3	q_6^3	q_{end}^3
q_1^3	0.6890	0.0000	0.1285	0.1824	0.0000	0.0000	0.0000
q_2^3	0.9435	0.0565	0.0000	0.0000	0.0000	0.0000	0.0000
q_3^3	0.0756	0.0000	0.4267	0.2485	0.2097	0.0395	0.0000
q_4^3	0.1425	0.0000	0.5377	0.1403	0.0264	0.1531	0.0000
q_5^3	0.0000	0.0000	0.0000	0.0000	0.6175	0.1840	0.1984
q_6^3	0.0000	0.0000	0.0002	0.0000	0.9622	0.0011	0.0366

Table A.22: Emission likelihoods of the production state children of the "enter/exit" submodel in the "washing dishes first" model stored at the children states of q_3^2.

State	Observation					
	Stove	Food Prep	Sink	Undefined	Fridge	Door
q_1^3	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000
q_2^3	0.0000	0.0000	0.3583	0.6417	0.0000	0.0000
q_3^3	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000
q_4^3	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000
q_5^3	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000
q_6^3	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000

A.1.3 "Washing Dishes and Preparing Food"

The state transition likelihoods and the emission probabilities for the "washing dishes and preparing food" class model in Figure 3.5 are presented here in Tables A.23–A.33.

Table A.23: Initial state likelihoods stored at the root node in the "washing dishes and preparing food" model.

Destination		
q_1^2	q_2^2	q_3^2
1.0000	0.0000	0.0000

Table A.24: State transition likelihoods governing the second layer of the "washing dishes and preparing food" model stored at the root node.

Source	Destination			
	q_1^2	q_2^2	q_3^2	q_{end}^2
q_1^2	0.0000	0.5119	0.0000	0.4881
q_2^2	0.0348	0.1267	0.8386	0.0000
q_3^2	1.0000	0.0000	0.0000	0.0000

Table A.25: Initial state likelihoods for the "enter/exit and wash dishes" submodel in the "washing dishes and preparing food" model stored at state q_1^2.

Destination					
q_1^3	q_2^3	q_3^3	q_4^3	q_5^3	q_6^3
0.0238	0.0000	0.0000	0.0000	0.9762	0.0000

Table A.26: State transition likelihoods governing the production state children of the "enter/exit and wash dishes" submodel in the "washing dishes and preparing food" model stored at state q_1^2.

Source	Destination						
	q_1^3	q_2^3	q_3^3	q_4^3	q_5^3	q_6^3	q_{end}^3
q_1^3	0.7046	0.0000	0.2954	0.0000	0.0000	0.0000	0.0000
q_2^3	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000
q_3^3	0.0000	0.0000	0.7046	0.0000	0.0000	0.0000	0.2954
q_4^3	0.0000	0.2512	0.0000	0.4976	0.0000	0.0000	0.2512
q_5^3	0.0000	0.0000	0.0000	0.6875	0.3125	0.0000	0.0000
q_6^3	0.9530	0.0000	0.0000	0.0000	0.0000	0.0470	0.0000

Table A.27: Emission likelihoods of the production state children of the "enter/exit and wash dishes" submodel in the "washing dishes and preparing food" model stored at the children states of q_1^2.

State	Observation					
	Stove	Food Prep	Sink	Undefined	Fridge	Door
q_1^3	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000
q_2^3	0.0000	0.0000	0.0000	0.3737	0.5000	0.1262
q_3^3	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000
q_4^3	0.0175	0.0000	0.0000	0.0000	0.0000	0.9825
q_5^3	0.0000	0.0000	0.0000	0.0363	0.1454	0.8183
q_6^3	0.0000	0.0000	0.0922	0.9078	0.0000	0.0000

Table A.28: Initial state likelihoods for the "food prep" submodel in the "washing dishes and preparing food" model stored at state q_2^2.

Destination					
q_1^3	q_2^3	q_3^3	q_4^3	q_5^3	q_6^3
0.4000	0.1492	0.0315	0.0000	0.0746	0.3447

Table A.29: State transition likelihoods governing the production state children of the "food prep" submodel in the "washing dishes and preparing food" model stored at state q_2^2.

Source	Destination						
	q_1^3	q_2^3	q_3^3	q_4^3	q_5^3	q_6^3	q_{end}^3
q_1^3	0.0000	0.2224	0.0553	0.0000	0.2145	0.0000	0.5077
q_2^3	0.0007	0.1362	0.4077	0.4212	0.0000	0.0342	0.0000
q_3^3	0.0433	0.0692	0.0874	0.7816	0.0004	0.0180	0.0001
q_4^3	0.0005	0.0177	0.0119	0.3829	0.2946	0.0280	0.2645
q_5^3	0.0000	0.0393	0.0523	0.0000	0.7775	0.1309	0.0000
q_6^3	0.0904	0.4819	0.1494	0.0218	0.0000	0.2504	0.0000

Table A.30: Emission likelihoods of the production state children of the "food prep" submodel in the "washing dishes and preparing food" model stored at the children states of q_2^2.

State	Observation					
	Stove	Food Prep	Sink	Undefined	Fridge	Door
q_1^3	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000
q_2^3	0.0000	0.0000	0.0000	0.3737	0.5000	0.1262
q_3^3	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000
q_4^3	0.1750	0.0000	0.0000	0.0000	0.0000	0.9825
q_5^3	0.0000	0.0000	0.0000	0.0363	0.1454	0.8183
q_6^3	0.0000	0.0000	0.0922	0.9078	0.0000	0.0000

Table A.31: Initial state likelihoods for the "cooking" submodel in the "washing dishes and preparing food" model stored at state q_3^2.

Destination					
q_1^3	q_2^3	q_3^3	q_4^3	q_5^3	q_6^3
0.0000	0.0000	0.0000	0.0000	1.0000	0.0000

Table A.32: State transition likelihoods governing the production state children of the "cooking" submodel in the "washing dishes and preparing food" model stored at state q_3^2.

Source	Destination						
	q_1^3	q_2^3	q_3^3	q_4^3	q_5^3	q_6^3	q_{end}^3
q_1^3	0.3658	0.0716	0.1500	0.2429	0.0000	0.1697	0.0000
q_2^3	0.2301	0.5512	0.2187	0.0000	0.0000	0.0000	0.0000
q_3^3	0.3718	0.0839	0.2516	0.1393	0.0000	0.1534	0.0000
q_4^3	0.0001	0.0000	0.0001	0.2488	0.0000	0.5138	0.2372
q_5^3	0.0000	0.9798	0.0000	0.0000	0.0202	0.0000	0.0000
q_6^3	0.0001	0.0000	0.0000	0.2636	0.0000	0.3655	0.3699

Table A.33: Emission likelihoods of the production state children of the "cooking" submodel in the "washing dishes and preparing food" model stored at the children states of q_3^2.

State	Observation					
	Stove	Food Prep	Sink	Undefined	Fridge	Door
q_1^3	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000
q_2^3	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000
q_3^3	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000
q_4^3	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000
q_5^3	0.0002	0.0717	0.0000	0.9281	0.0000	0.0000
q_6^3	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000

A.1.4 "Round Robin"

The state transition likelihoods and the emission probabilities for the "round robin" class model in Figure 3.6 are presented here in Tables A.34–A.44.

Table A.34: Initial state likelihoods stored at the root node in the "round robin" model.

Destination		
q_1^2	q_2^2	q_3^2
0.0000	0.0000	1.0000

Table A.35: State transition likelihoods governing the second layer of the "round robin" model stored at the root node.

	Destination			
Source	q_1^2	q_2^2	q_3^2	q_{end}^2
q_1^2	0.0000	0.0000	1.0000	0.0000
q_2^2	1.0000	0.0000	0.0000	0.0000
q_3^2	0.0000	0.5000	0.0000	0.5000

Table A.36: Initial state likelihoods for the "cooking" submodel in the "round robin" model stored at state q_1^2.

Destination					
q_1^3	q_2^3	q_3^3	q_4^3	q_5^3	q_6^3
0.0000	0.9964	0.0000	0.0000	0.0000	0.0036

Table A.37: State transition likelihoods governing the production state children of the "cooking" submodel in the "round robin" model stored at state q_1^2.

Source	Destination						
	q_1^3	q_2^3	q_3^3	q_4^3	q_5^3	q_6^3	q_{end}^3
q_1^3	0.5900	0.0000	0.3777	0.0000	0.0321	0.0000	0.0002
q_2^3	0.0000	0.0000	0.0000	0.0003	0.0000	0.9997	0.0000
q_3^3	0.0000	0.0000	0.1055	0.0000	0.0000	0.0000	0.8945
q_4^3	0.0000	0.0000	0.0000	0.6459	0.3541	0.0000	0.0000
q_5^3	0.4164	0.0000	0.0000	0.0000	0.5831	0.0000	0.0000
q_6^3	0.0000	0.0000	0.0000	0.8694	0.0000	0.1306	0.0000

Table A.38: Emission likelihoods of the production state children of the "cooking" submodel in the "round robin" model stored at the children states of q_1^2.

State	Observation					
	Stove	Food Prep	Sink	Undefined	Fridge	Door
q_1^3	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000
q_2^3	0.0732	0.0000	0.3098	0.6170	0.0000	0.0000
q_3^3	0.6277	0.0000	0.0000	0.1861	0.1861	0.0000
q_4^3	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000
q_5^3	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000
q_6^3	0.9135	0.0000	0.0000	0.0865	0.0000	0.0000

Table A.39: Initial state likelihoods for the "food prep" submodel in the "round robin" model stored at state q_2^2.

Destination					
q_1^3	q_2^3	q_3^3	q_4^3	q_5^3	q_6^3
0.0000	1.0000	0.0000	0.0000	0.0000	0.0000

Table A.40: State transition likelihoods governing the production state children of the "food prep" submodel in the "round robin" model stored at state q_2^2.

Source	Destination						
	q_1^3	q_2^3	q_3^3	q_4^3	q_5^3	q_6^3	q_{end}^3
q_1^3	0.8277	0.0000	0.0000	0.0000	0.0000	0.0000	0.1723
q_2^3	0.0000	0.7523	0.0000	0.2477	0.0000	0.0000	0.0000
q_3^3	0.8571	0.0714	0.0714	0.0000	0.0000	0.0000	0.0000
q_4^3	0.0000	0.0000	0.0000	0.1116	0.0000	0.8884	0.0000
q_5^3	0.2781	0.0360	0.3230	0.2919	0.0710	0.0000	0.0000
q_6^3	0.1245	0.0000	0.0000	0.0000	0.7640	0.1115	0.0000

Table A.41: Emission likelihoods of the production state children of the "food prep" submodel in the "round robin" model stored at the children states of q_2^2.

State	Observation					
	Stove	Food Prep	Sink	Undefined	Fridge	Door
q_1^3	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000
q_2^3	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000
q_3^3	0.0000	0.0000	0.0000	1.0000	0.0000	0.0000
q_4^3	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000
q_5^3	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000
q_6^3	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000

Table A.42: Initial state likelihoods for the "enter/exit via stove" submodel in the "round robin" model stored at state q_3^2.

Destination					
q_1^3	q_2^3	q_3^3	q_4^3	q_5^3	q_6^3
0.0000	0.0000	0.0000	1.0000	0.0000	0.0000

Table A.43: State transition likelihoods governing the production state children of the "enter/exit via stove" submodel in the "round robin" model stored at state q_3^2.

Source	Destination						
	q_1^3	q_2^3	q_3^3	q_4^3	q_5^3	q_6^3	q_{end}^3
q_1^3	0.1568	0.0000	0.8432	0.0000	0.0000	0.0000	0.0000
q_2^3	0.0000	0.3620	0.0000	0.0000	0.3190	0.0000	0.3190
q_3^3	0.0000	0.0000	0.1568	0.0000	0.0000	0.8432	0.0000
q_4^3	0.0000	0.5919	0.0000	0.4081	0.0000	0.0000	0.0000
q_5^3	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
q_6^3	0.0478	0.0000	0.0000	0.0000	0.0000	0.1350	0.8173

Table A.44: Emission likelihoods of the production state children of the "enter/exit via stove" submodel in the "round robin" model stored at the children states of q_3^2.

State	Observation					
	Stove	Food Prep	Sink	Undefined	Fridge	Door
q_1^3	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000
q_2^3	0.0000	0.0000	0.0000	0.0244	0.1348	0.8408
q_3^3	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000
q_4^3	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000
q_5^3	0.3636	0.0000	0.0000	0.4223	0.2141	0.0000
q_6^3	0.6802	0.0000	0.0000	0.3198	0.0000	0.0000

A.2 Lounge Room Models

A.2.1 "Watch Television"

The state transition likelihoods and the emission probabilities for the "watch television" class model in Figure 3.7 are presented here in Tables A.45–A.55.

Table A.45: Initial state likelihoods stored at the root node in the "watch television" model.

Destination		
q_1^2	q_2^2	q_3^2
0.0000	1.0000	0.0000

Table A.46: State transition likelihoods governing the second layer of the "watch television" model stored at the root node.

Source	Destination			
	q_1^2	q_2^2	q_3^2	q_{end}^2
q_1^2	0.0000	0.0000	0.5119	0.4881
q_2^2	1.0000	0.0000	0.0000	0.0000
q_3^2	0.8845	0.0000	0.1155	0.0000

Table A.47: Initial state likelihoods for the "walk across room / exit" sub-model in the "watch television" model stored at state q_1^2.

Destination					
q_1^3	q_2^3	q_3^3	q_4^3	q_5^3	q_6^3
0.0000	0.1787	0.0000	0.0000	0.0000	0.8213

Table A.48: State transition likelihoods governing the production state children of the "walk across room / exit" submodel in the "watch television" model stored at state q_1^2.

Source	Destination						
	q_1^3	q_2^3	q_3^3	q_4^3	q_5^3	q_6^3	q_{end}^3
q_1^3	0.6845	0.0000	0.0000	0.0000	0.0000	0.0000	0.3155
q_2^3	0.0000	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000
q_3^3	0.0000	0.0000	0.2895	0.0907	0.5730	0.0000	0.0468
q_4^3	0.0000	0.0000	0.7588	0.2442	0.0000	0.0000	0.0000
q_5^3	0.5214	0.0000	0.0000	0.0000	0.0000	0.0000	0.4786
q_6^3	0.0000	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000

Table A.49: Emission likelihoods of the production state children of the "walk across room / exit" submodel in the "watch television" model stored at the children states of q_1^2.

State	Observation					
	Stove	Food Prep	Sink	Undefined	Fridge	Door
q_1^3	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000
q_2^3	0.0103	0.0000	0.0000	0.9897	0.0000	0.0000
q_3^3	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000
q_4^3	0.9806	0.0000	0.0000	0.0194	0.0000	0.0000
q_5^3	0.4795	0.0000	0.0000	0.4090	0.1115	0.0000
q_6^3	0.4599	0.0000	0.0000	0.5401	0.0000	0.0000

Table A.50: Initial state likelihoods for the "enter" submodel in the "watch television" model stored at state q_2^2.

Destination					
q_1^3	q_2^3	q_3^3	q_4^3	q_5^3	q_6^3
0.0000	0.0000	1.0000	0.0000	0.0000	0.0000

Table A.51: State transition likelihoods governing the production state children of the “enter” submodel in the “watch television” model stored at state q_2^2.

Source	Destination						
	q_1^3	q_2^3	q_3^3	q_4^3	q_5^3	q_6^3	q_{end}^3
q_1^3	0.0000	0.7857	0.0000	0.0000	0.0000	0.0000	0.2143
q_2^3	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000
q_3^3	0.0000	0.0000	0.0000	0.0000	0.2206	0.7794	0.0000
q_4^3	0.0000	0.7875	0.0000	0.0000	0.0000	0.0000	0.2125
q_5^3	0.1241	0.1350	0.0000	0.7408	0.0000	0.0000	0.0000
q_6^3	0.7704	0.1350	0.0000	0.0946	0.0000	0.0000	0.0000

Table A.52: Emission likelihoods of the production state children of the “enter” submodel in the “watch television” model stored at the children states of q_2^2.

State	Observation					
	Stove	Food Prep	Sink	Undefined	Fridge	Door
q_1^3	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000
q_2^3	0.0000	0.0000	0.0000	0.0000	0.4205	0.5795
q_3^3	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000
q_4^3	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000
q_5^3	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000
q_6^3	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000

Table A.53: Initial state likelihoods for the “watch TV” submodel in the “watch television” model stored at state q_3^2.

Destination					
q_1^3	q_2^3	q_3^3	q_4^3	q_5^3	q_6^3
0.0000	0.0000	0.0075	0.0000	0.0000	0.9925

Table A.54: State transition likelihoods governing the production state children of the "watch TV" submodel in the "watch television" model stored at state q_3^2.

Source	Destination						
	q_1^3	q_2^3	q_3^3	q_4^3	q_5^3	q_6^3	q_{end}^3
q_1^3	0.5128	0.4368	0.0291	0.0000	0.0000	0.0118	0.0000
q_2^3	0.0089	0.6789	0.0000	0.0430	0.2691	0.0000	0.0000
q_3^3	0.3828	0.1232	0.1623	0.0000	0.0000	0.3316	0.0000
q_4^3	0.0002	0.0223	0.0000	0.1766	0.3659	0.0000	0.4349
q_5^3	0.0004	0.0164	0.0000	0.5972	0.3035	0.0000	0.0825
q_6^3	0.1102	0.0110	0.3200	0.0000	0.0000	0.5589	0.0000

Table A.55: Emission likelihoods of the production state children of the "watch TV" submodel in the "watch television" model stored at the children states of q_3^2.

State	Observation					
	Stove	Food Prep	Sink	Undefined	Fridge	Door
q_1^3	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000
q_2^3	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000
q_3^3	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000
q_4^3	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000
q_5^3	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000
q_6^3	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000

A.2.2 "Read Book on Couch"

The state transition likelihoods and the emission probabilities for the "read book on couch" class model in Figure 3.8 are presented here in Tables A.56–A.66.

Table A.56: Initial state likelihoods stored at the root node in the "read book on couch" model.

Destination		
q_1^2	q_2^2	q_3^2
0.0000	1.0000	0.0000

Table A.57: State transition likelihoods governing the second layer of the "read book on couch" model stored at the root node.

Source	Destination			
	q_1^2	q_2^2	q_3^2	q_{end}^2
q_1^2	0.0000	0.0357	0.9643	0.0000
q_2^2	1.0000	0.0000	0.0000	0.0000
q_3^2	0.0000	0.0000	0.0000	1.0000

Table A.58: Initial state likelihoods for the "read on couch" submodel in the "read book on couch" model stored at state q_1^2.

Destination					
q_1^3	q_2^3	q_3^3	q_4^3	q_5^3	q_6^3
0.0000	0.0000	0.9286	0.0000	0.0000	0.0714

Table A.59: State transition likelihoods governing the production state children of the "read on couch" submodel in the "read book on couch" model stored at state q_1^2.

Source	Destination						
	q_1^3	q_2^3	q_3^3	q_4^3	q_5^3	q_6^3	q_{end}^3
q_1^3	0.3944	0.3384	0.0000	0.1397	0.1274	0.0000	0.0000
q_2^3	0.0000	0.7723	0.0000	0.0000	0.0000	0.0000	0.2277
q_3^3	0.0000	0.0000	0.2143	0.0000	0.0006	0.7851	0.0000
q_4^3	0.2581	0.2317	0.0000	0.4009	0.1086	0.0007	0.0000
q_5^3	0.2478	0.0014	0.0000	0.3244	0.4127	0.0136	0.0000
q_6^3	0.0005	0.0000	0.0000	0.0091	0.2449	0.7455	0.0000

Table A.60: Emission likelihoods of the production state children of the "read on couch" submodel in the "read book on couch" model stored at the children states of q_1^2.

State	Observation					
	Stove	Food Prep	Sink	Undefined	Fridge	Door
q_1^3	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000
q_2^3	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000
q_3^3	0.0000	0.0000	0.0000	1.0000	0.0000	0.0000
q_4^3	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000
q_5^3	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000
q_6^3	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000

Table A.61: Initial state likelihoods for the "enter via bookcase" submodel in the "read book on couch" model stored at state q_2^2.

Destination					
q_1^3	q_2^3	q_3^3	q_4^3	q_5^3	q_6^3
0.0357	0.0000	0.0000	0.0000	0.0000	0.9643

Table A.62: State transition likelihoods governing the production state children of the "enter via bookcase" submodel in the "read book on couch" model stored at state q_2^2.

Source	Destination						
	q_1^3	q_2^3	q_3^3	q_4^3	q_5^3	q_6^3	q_{end}^3
q_1^3	0.0000	0.6512	0.0000	0.0000	0.0000	0.3488	0.0000
q_2^3	0.0000	0.6317	0.0000	0.3683	0.0000	0.0000	0.0000
q_3^3	0.0000	0.0000	0.6321	0.0000	0.0000	0.0000	0.3679
q_4^3	0.0000	0.0001	0.3682	0.6317	0.0000	0.0000	0.0000
q_5^3	0.0260	0.6381	0.0000	0.0000	0.3359	0.0000	0.0000
q_6^3	0.0000	0.1633	0.0000	0.0000	0.2600	0.5767	0.0000

Table A.63: Emission likelihoods of the production state children of the "enter via bookcase" submodel in the "read book on couch" model stored at the children states of q_2^2.

State	Observation					
	Stove	Food Prep	Sink	Undefined	Fridge	Door
q_1^3	0.0000	0.0000	0.0000	1.0000	0.0000	0.0000
q_2^3	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000
q_3^3	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000
q_4^3	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000
q_5^3	0.1252	0.0000	0.0000	0.0000	0.0000	0.8748
q_6^3	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000

Table A.64: Initial state likelihoods for the "exit via bookcase" submodel in the "read book on couch" model stored at state q_3^2.

Destination					
q_1^3	q_2^3	q_3^3	q_4^3	q_5^3	q_6^3
0.0000	0.0000	1.0000	0.0000	0.0000	0.0000

Table A.65: State transition likelihoods governing the production state children of the "exit via bookcase" submodel in the "read book on couch" model stored at state q_3^2.

Source	Destination						
	q_1^3	q_2^3	q_3^3	q_4^3	q_5^3	q_6^3	q_{end}^3
q_1^3	0.0304	0.7746	0.0000	0.0000	0.0000	0.1950	0.0000
q_2^3	0.1890	0.0090	0.0000	0.0000	0.1908	0.6112	0.0000
q_3^3	0.7381	0.0000	0.2619	0.0000	0.0000	0.0000	0.0000
q_4^3	0.0000	0.0000	0.0000	0.1096	0.0000	0.0000	0.8904
q_5^3	0.0000	0.0000	0.0000	0.8904	0.1096	0.0000	0.0000
q_6^3	0.0000	0.0000	0.0000	0.0000	0.9996	0.0004	0.0000

Table A.66: Emission likelihoods of the production state children of the "exit via bookcase" submodel in the "read book on couch" model stored at the children states of q_3^2.

State	Observation					
	Stove	Food Prep	Sink	Undefined	Fridge	Door
q_1^3	0.0444	0.0000	0.0000	0.0000	0.9556	0.0000
q_2^3	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000
q_3^3	0.0000	0.0000	0.0000	1.0000	0.0000	0.0000
q_4^3	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000
q_5^3	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000
q_6^3	0.0000	0.0000	0.0000	0.0000	0.2276	0.7724

A.2.3 "Eat Dinner"

The state transition likelihoods and the emission probabilities for the "eat dinner" class model in Figure 3.9 are presented here in Tables A.67–A.77.

Table A.67: Initial state likelihoods stored at the root node in the "eat dinner" model.

Destination		
q_1^2	q_2^2	q_3^2
1.0000	0.0000	0.0000

Table A.68: State transition likelihoods governing the second layer of the "eat dinner" model stored at the root node.

Source	Destination			
	q_1^2	q_2^2	q_3^2	q_{end}^2
q_1^2	0.0000	0.5000	0.0000	0.5000
q_2^2	0.0000	0.0000	1.0000	0.0000
q_3^2	0.9286	0.0714	0.0000	0.0000

Table A.69: Initial state likelihoods for the "enter/exit" submodel in the "eat dinner" model stored at state q_1^2.

Destination					
q_1^3	q_2^3	q_3^3	q_4^3	q_5^3	q_6^3
0.0000	0.0000	1.0000	0.0000	0.0000	0.0000

Table A.70: State transition likelihoods governing the production state children of the "enter/exit" submodel in the "eat dinner" model stored at state q_1^2.

Source	Destination						
	q_1^3	q_2^3	q_3^3	q_4^3	q_5^3	q_6^3	q_{end}^3
q_1^3	0.0001	0.0000	0.0000	0.0000	0.0000	0.1660	0.8339
q_2^3	0.6550	0.0031	0.0000	0.2335	0.0000	0.0000	0.1084
q_3^3	0.0001	0.9976	0.0006	0.0016	0.0000	0.0000	0.0000
q_4^3	0.0019	0.0000	0.0000	0.0000	0.0000	0.1661	0.8320
q_5^3	0.0000	0.6591	0.1976	0.0000	0.0000	0.0000	0.1433
q_6^3	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000

Table A.71: Emission likelihoods of the production state children of the "enter/exit" submodel in the "eat dinner" model stored at the children states of q_1^2.

State	Observation					
	Stove	Food Prep	Sink	Undefined	Fridge	Door
q_1^3	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000
q_2^3	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000
q_3^3	0.0000	0.0000	0.0000	0.0000	0.0409	0.9591
q_4^3	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000
q_5^3	0.0000	0.0000	0.0000	0.0000	0.9999	0.0001
q_6^3	0.2244	0.0000	0.0000	0.0000	0.0000	0.7756

Table A.72: Initial state likelihoods for the first "eat dinner" submodel in the "eat dinner" model stored at state q_2^2.

Destination					
q_1^3	q_2^3	q_3^3	q_4^3	q_5^3	q_6^3
0.0000	0.0284	0.0000	0.9659	0.0057	0.0000

Table A.73: State transition likelihoods governing the production state children of the first "eat dinner" submodel in the "eat dinner" model stored at state q_2^2.

Source	Destination						
	q_1^3	q_2^3	q_3^3	q_4^3	q_5^3	q_6^3	q_{end}^3
q_1^3	0.7785	0.0000	0.0000	0.0000	0.0000	0.2215	0.0000
q_2^3	0.7183	0.0000	0.2817	0.0000	0.0000	0.0000	0.0000
q_3^3	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
q_4^3	0.2099	0.7001	0.0000	0.0000	0.0900	0.0000	0.0000
q_5^3	0.7137	0.0000	0.2863	0.0000	0.0000	0.0000	0.0000
q_6^3	0.0000	0.0000	0.0000	0.0000	0.0000	0.7785	0.2214

Table A.74: Emission likelihoods of the production state children of the first "eat dinner" submodel in the "eat dinner" model stored at the children states of q_2^2.

State	Observation					
	Stove	Food Prep	Sink	Undefined	Fridge	Door
q_1^3	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000
q_2^3	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000
q_3^3	0.0000	0.0000	0.0000	0.2523	0.7477	0.0000
q_4^3	0.0000	0.0000	0.0000	0.0000	0.4016	0.5984
q_5^3	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000
q_6^3	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000

Table A.75: Initial state likelihoods for the second "eat dinner" submodel in the "eat dinner" model stored at state q_3^2.

Destination					
q_1^3	q_2^3	q_3^3	q_4^3	q_5^3	q_6^3
0.0000	0.0000	0.0000	0.0000	0.0013	0.9987

Table A.76: State transition likelihoods governing the production state children of the second "eat dinner" submodel in the "eat dinner" model stored at state q_3^2.

Source	Destination						
	q_1^3	q_2^3	q_3^3	q_4^3	q_5^3	q_6^3	q_{end}^3
q_1^3	0.4336	0.0040	0.2300	0.2622	0.0569	0.0132	0.0000
q_2^3	0.0000	0.1294	0.0000	0.0000	0.0000	0.0000	0.8706
q_3^3	0.1050	0.1051	0.1031	0.6667	0.0139	0.0062	0.0000
q_4^3	0.0571	0.3543	0.1983	0.3559	0.0322	0.0023	0.0000
q_5^3	0.4591	0.0000	0.2415	0.0266	0.0610	0.2118	0.0000
q_6^3	0.1416	0.0000	0.0022	0.0000	0.1952	0.6611	0.0000

Table A.77: Emission likelihoods of the production state children of the second "eat dinner" submodel in the "eat dinner" model stored at the children states of q_3^2.

State	Observation					
	Stove	Food Prep	Sink	Undefined	Fridge	Door
q_1^3	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000
q_2^3	0.0000	0.0000	0.0000	0.0000	0.9433	0.0567
q_3^3	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000
q_4^3	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000
q_5^3	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000
q_6^3	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000

A.2.4 "Eat Dinner While Watching TV"

The state transition likelihoods and the emission probabilities for the "eat dinner while watching TV' class model in Figure 3.10 are presented here in Tables A.78–A.88.

Table A.78: Initial state likelihoods stored at the root node in the "eat dinner while watching TV" model.

Destination		
q_1^2	q_2^2	q_3^2
0.0000	1.0000	0.0000

Table A.79: State transition likelihoods governing the second layer of the "eat dinner while watching TV" model stored at the root node.

Source	Destination			
	q_1^2	q_2^2	q_3^2	q_{end}^2
q_1^2	0.0000	0.0000	1.0000	0.0000
q_2^2	0.0000	0.0000	0.5000	0.5000
q_3^2	0.5357	0.4643	0.0000	0.0000

Table A.80: Initial state likelihoods for the "eat dinner" submodel in the "eat dinner while watching TV" model stored at state q_1^2.

Destination					
q_1^3	q_2^3	q_3^3	q_4^3	q_5^3	q_6^3
0.0000	0.0000	0.0956	0.0000	0.0000	0.9044

Table A.81: State transition likelihoods governing the production state children of the "eat dinner" submodel in the "eat dinner while watching TV" model stored at state q_1^2.

Source	Destination						
	q_1^3	q_2^3	q_3^3	q_4^3	q_5^3	q_6^3	q_{end}^3
q_1^3	0.7958	0.0000	0.0000	0.0000	0.0000	0.0000	0.2042
q_2^3	0.0266	0.5070	0.0000	0.0464	0.4199	0.0000	0.0000
q_3^3	0.0000	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000
q_4^3	0.0000	0.4586	0.0000	0.5124	0.0039	0.0000	0.0250
q_5^3	0.3868	0.1833	0.0000	0.1089	0.3211	0.0000	0.0000
q_6^3	0.0000	0.0000	0.9933	0.0067	0.0000	0.0000	0.0000

Table A.82: Emission likelihoods of the production state children of the "eat dinner" submodel in the "eat dinner while watching TV" model stored at the children states of q_1^2.

State	Observation					
	Stove	Food Prep	Sink	Undefined	Fridge	Door
q_1^3	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000
q_2^3	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000
q_3^3	0.0000	0.7510	0.0000	0.2490	0.0000	0.0000
q_4^3	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000
q_5^3	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000
q_6^3	0.0000	0.0000	0.0000	0.8720	0.1280	0.0000

Table A.83: Initial state likelihoods for the "enter/exit" submodel in the "eat dinner while watching TV" model stored at state q_2^2.

Destination					
q_1^3	q_2^3	q_3^3	q_4^3	q_5^3	q_6^3
0.2265	0.0365	0.7369	0.0000	0.0000	0.0000

Table A.84: State transition likelihoods governing the production state children of the "enter/exit" submodel in the "eat dinner while watching TV" model stored at state q_2^2.

Source	Destination						
	q_1^3	q_2^3	q_3^3	q_4^3	q_5^3	q_6^3	q_{end}^3
q_1^3	0.0000	0.4647	0.0000	0.0000	0.4968	0.0385	0.0000
q_2^3	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000
q_3^3	0.7103	0.2713	0.0000	0.0000	0.0184	0.0000	0.0000
q_4^3	0.6967	0.2990	0.0000	0.0000	0.0043	0.0000	0.0000
q_5^3	0.0000	0.0000	0.0000	0.0000	0.0000	0.9727	0.0273
q_6^3	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000

Table A.85: Emission likelihoods of the production state children of the "enter/exit" submodel in the "eat dinner while watching TV" model stored at the children states of q_2^2.

State	Observation					
	Stove	Food Prep	Sink	Undefined	Fridge	Door
q_1^3	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000
q_2^3	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000
q_3^3	0.0000	0.0000	0.0000	0.0000	0.4586	0.5414
q_4^3	0.0000	0.0000	0.0000	0.0000	0.8457	0.1543
q_5^3	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000
q_6^3	0.0000	0.0000	0.0000	0.0590	0.0000	0.9410

Table A.86: Initial state likelihoods for the "walk by TV" submodel in the "eat dinner while watching TV" model stored at state q_3^2.

Destination					
q_1^3	q_2^3	q_3^3	q_4^3	q_5^3	q_6^3
0.0000	0.2990	0.0700	0.0374	0.0000	0.5935

Table A.87: State transition likelihoods governing the production state children of the "walk by TV" submodel in the "eat dinner while watching TV" model stored at state q_3^2.

Source	Destination						
	q_1^3	q_2^3	q_3^3	q_4^3	q_5^3	q_6^3	q_{end}^3
q_1^3	0.1372	0.0000	0.0000	0.0547	0.0000	0.0000	0.8081
q_2^3	0.0775	0.0697	0.2680	0.0583	0.5264	0.0000	0.0000
q_3^3	0.0177	0.0000	0.0000	0.0000	0.9823	0.0000	0.0000
q_4^3	0.8672	0.0000	0.0000	0.1305	0.0024	0.0000	0.0000
q_5^3	0.0000	0.0000	0.0000	0.8140	0.1706	0.0000	0.0154
q_6^3	0.0000	0.0000	0.8579	0.0655	0.0004	0.0000	0.0762

Table A.88: Emission likelihoods of the production state children of the "walk by TV" submodel in the "eat dinner while watching TV" model stored at the children states of q_3^2.

State	Observation					
	Stove	Food Prep	Sink	Undefined	Fridge	Door
q_1^3	0.9673	0.0000	0.0000	0.0327	0.0000	0.0000
q_2^3	0.0018	0.0000	0.0000	0.9982	0.0000	0.0000
q_3^3	0.8063	0.0000	0.0000	0.1937	0.0000	0.0000
q_4^3	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000
q_5^3	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000
q_6^3	0.0562	0.0000	0.0000	0.5573	0.3864	0.0000

A.2.5 "There is Nothing Good on TV – Read a Book Instead"

The state transition likelihoods and the emission probabilities for the "there is nothing good on TV" class model in Figure 3.11 are presented here in Tables A.89–A.99.

Table A.89: Initial state likelihoods stored at the root node in the "there is nothing good on TV" model.

Destination		
q_1^2	q_2^2	q_3^2
1.0000	0.0000	0.0000

Table A.90: State transition likelihoods governing the second layer of the "there is nothing good on TV" model stored at the root node.

Source	Destination			
	q_1^2	q_2^2	q_3^2	q_{end}^2
q_1^2	0.0000	0.0000	1.0000	0.0000
q_2^2	0.0000	0.0000	0.0000	1.0000
q_3^2	0.4643	0.5357	0.0000	0.0000

Table A.91: Initial state likelihoods for the "enter via TV" submodel in the "there is nothing good on TV" model stored at state q_1^2.

Destination					
q_1^3	q_2^3	q_3^3	q_4^3	q_5^3	q_6^3
0.3842	0.0000	0.0436	0.0000	0.0365	0.5357

Table A.92: State transition likelihoods governing the production state children of the "enter via TV" submodel in the "there is nothing good on TV" model stored at state q_1^2.

Source	Destination						
	q_1^3	q_2^3	q_3^3	q_4^3	q_5^3	q_6^3	q_{end}^3
q_1^3	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000	0.0000
q_2^3	0.0000	0.2615	0.0223	0.0000	0.0896	0.0000	0.6267
q_3^3	0.0000	0.9264	0.0158	0.0000	0.0057	0.0000	0.0520
q_4^3	0.8520	0.0000	0.0000	0.1480	0.0000	0.0000	0.0000
q_5^3	0.0000	0.3858	0.2986	0.0000	0.3156	0.0000	0.0000
q_6^3	0.0000	0.0000	0.0000	0.3632	0.0000	0.6368	0.0000

Table A.93: Emission likelihoods of the production state children of the "enter via TV" submodel in the "there is nothing good on TV" model stored at the children states of q_1^2.

State	Observation					
	Stove	Food Prep	Sink	Undefined	Fridge	Door
q_1^3	0.2116	0.0000	0.0000	0.7884	0.0000	0.0000
q_2^3	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000
q_3^3	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000
q_4^3	0.0000	0.0000	0.0000	0.0000	0.5088	0.4912
q_5^3	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000
q_6^3	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000

Table A.94: Initial state likelihoods for the "exit via bookcase" submodel in the "there is nothing good on TV" model stored at state q_2^2.

Destination					
q_1^3	q_2^3	q_3^3	q_4^3	q_5^3	q_6^3
0.0000	0.0000	0.0000	0.0000	0.0000	1.0000

Table A.95: State transition likelihoods governing the production state children of the "exit via bookcase" submodel in the "there is nothing good on TV" model stored at state q_2^2.

Source	Destination						
	q_1^3	q_2^3	q_3^3	q_4^3	q_5^3	q_6^3	q_{end}^3
q_1^3	0.0000	0.2136	0.7864	0.0000	0.0000	0.0000	0.0000
q_2^3	0.0000	0.0356	0.1133	0.8512	0.0000	0.0000	0.0000
q_3^3	0.0000	0.0000	0.0607	0.9393	0.0000	0.0000	0.0000
q_4^3	0.0000	0.0000	0.0000	0.0927	0.9073	0.0000	0.0000
q_5^3	0.0000	0.0000	0.0000	0.0000	0.0000	0.0767	0.9233
q_6^3	0.99997	0.0000	0.0000	0.0000	0.0000	0.0003	0.0000

Table A.96: Emission likelihoods of the production state children of the "exit via bookcase" submodel in the "there is nothing good on TV" model stored at the children states of q_2^2.

State	Observation					
	Stove	Food Prep	Sink	Undefined	Fridge	Door
q_1^3	0.0714	0.0000	0.0000	0.5711	0.2860	0.0715
q_2^3	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000
q_3^3	0.0000	0.0000	0.0000	0.0000	0.0010	0.9990
q_4^3	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000
q_5^3	0.0646	0.0000	0.0000	0.0000	0.0000	0.9354
q_6^3	0.0000	0.0000	0.0000	0.9286	0.0714	0.0000

Table A.97: Initial state likelihoods for the "find book / use couch" submodel in the "there is nothing good on TV" model stored at state q_3^2.

Destination					
q_1^3	q_2^3	q_3^3	q_4^3	q_5^3	q_6^3
0.5804	0.0010	0.2463	0.0000	0.0000	0.1723

Table A.98: State transition likelihoods governing the production state children of the "find book / use couch" submodel in the "there is nothing good on TV" model stored at state q_3^2.

Source	Destination						
	q_1^3	q_2^3	q_3^3	q_4^3	q_5^3	q_6^3	q_{end}^3
q_1^3	0.0000	0.3984	0.4949	0.0000	0.1067	0.0000	0.0000
q_2^3	0.0000	0.6520	0.0000	0.3465	0.0000	0.0015	0.0000
q_3^3	0.0000	0.0000	0.8389	0.0000	0.1492	0.0119	0.0000
q_4^3	0.0000	0.0213	0.0086	0.6758	0.0000	0.0001	0.2942
q_5^3	0.0000	0.0000	0.0000	0.0000	0.0952	0.9048	0.0000
q_6^3	0.0000	0.1437	0.0000	0.0000	0.0000	0.8563	0.0000

Table A.99: Emission likelihoods of the production state children of the "find book / use couch" submodel in the "there is nothing good on TV" model stored at the children states of q_3^2.

State	Observation					
	Stove	Food Prep	Sink	Undefined	Fridge	Door
q_1^3	0.0146	0.0000	0.0000	0.9854	0.0000	0.0000
q_2^3	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000
q_3^3	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000
q_4^3	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000
q_5^3	0.0000	0.0000	0.0000	1.0000	0.0000	0.0000
q_6^3	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000

Bibliography

R. C. Agarwal, C. C. Aggarwal, and V. V. V. Prasad. A tree projection algorithm for generation of frequent item sets. *Journal of Parallel and Distributed Computing*, 61 (3):350–371, 2001.

J. K. Aggarwal and Q. Cai. Human motion analysis: A review. *Computer Vision and Image Understanding*, 73(3):428–440, 1999.

R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In J. B. Bocca, M. Jarke, and C. Zaniolo, editors, *Proc. 20th Int'l Conf. Very Large Data Bases*, pages 487–499, Santiago, Chile, September 1994. Morgan Kaufmann.

R. Agrawal and R. Srikant. Mining sequential patterns. In P. S. Yu and A. L. P. Chen, editors, *Proc. Eleventh Int'l Conf. Data Engineering*, pages 3–14, Taipei, Taiwan, March 1995. IEEE Computer Society.

R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. In P. Buneman and S. Jajodia, editors, *Proc. 1993 ACM SIGMOD Int'l Conf. Management of Data*, volume 22 of *SIGMOD Record*, pages 207–216. ACM Press, 1993.

R. Agrawal, M. Mehta, J. Shafer, R. Srikant, A. Arning, and T. Bollinger. The quest data mining system. In E. Simoudis, J. Han, and U. M. Fayyad, editors, *Proc. Second Int'l Conf. Knowledge Discovery and Data Mining*, pages 244–249, Portland, Oregon, USA, August 1996. AAAI Press.

J. F. Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11):832–843, 1983.

L. E. Baum, T. Petrie, G. Soules, and N. Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *The Annals of Mathematical Statistics*, 41(1):164–171, 1970.

C. Berberidis, L. Angelis, and I. Vlahavas. PREVENT: An algorithm for mining inter-transactional patterns for the prediction of rare events. In E. Onaindia and S. Staab, editors, *Proc. Second Starting AI Researchers' Symposium*, volume 109 of *Frontiers in Artificial Intelligence and Applications*, Valencia, Spain, August 2004. IOS Press.

C. Bettini, S. Wang, S. Jajodia, and J.-L. Lin. Discovering frequent event patterns with multiple granularities in time sequences. *IEEE Trans. Knowledge and Data Engineering*, 10(2):222–237, March/April 1998a.

C. Bettini, X. S. Wang, and S. Jajodia. A general framework for time granularity and its application to temporal reasoning. *Annals of Mathematics and Artificial Intelligence*, 22(1–2):29–58, 1998b.

J. Blanchard, F. Guillet, and H. Briand. Exploratory visualization for association rule rummaging. In *Proc. KDD 2003 Workshop on Multimedia Data Mining*, pages 107–114, Washington, District of Columbia, USA, 2003.

M. Brand and V. Kettnaker. Discovery and segmentation of activities in video. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 22(8):844–851, August 2000.

M. Brand, N. Oliver, and A. Pentland. Coupled hidden Markov models for complex action recognition. In *Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition*, pages 994–999, San Juan, Puerto Rico, June 1997. IEEE Computer Society.

C. Brunk, J. Kelly, and R. Kohavi. MineSet: An integrated system for data mining. In D. Heckerman, H. Mannila, and D. Pregibon, editors, *Proc. Third Int'l Conf. Knowledge Discovery and Data Mining*, pages 135–138, Newport Beach, California, USA, August 1997. AAAI Press.

D. Bruzzese and C. Davino. Visual post analysis of association rules. *Journal of Visual Languages and Computing*, 14(6):621–635, December 2003.

H. H. Bui, S. Venkatesh, and G. West. Tracking and surveillance in wide-area spatial environments using the abstract hidden Markov model. *International Journal of Pattern Recognition and Artificial Intelligence*, 15(1):177–196, February 2001.

H. H. Bui, S. Venkatesh, and G. West. Policy recognition in the abstract hidden Markov model. *Journal of Artificial Intelligence Research*, 17:451–499, 2002.

O. Cappé. Ten years of HMMs. viewed 12 February 2006, URL http://www.tsi.enst.fr/ cappe/docs/hmmbib.html, March 2001.

D. J. Cook, M. Youngblood, E. O. Heierman, III, K. Gopalratnam, S. Rao, A. Litvin, and F. Khawaja. MavHome: An agent-based smart home. In *Proc. First IEEE Int'l Conf. Pervasive Computing and Communications*, pages 521–524, Fort Worth, Texas, USA, March 2003. IEEE Computer Society.

R. Cucchiara, C. Grana, M. Piccardi, and A. Prati. Detecting moving objects, ghosts and shadows in video streams. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 25(10):1337–1342, October 2003.

G. Dong and J. Li. Efficient mining of emerging patterns: Discovering trends and differences. In *Proc. Fifth ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining*, pages 43–52, San Diego, California, USA, August 1999. ACM Press.

T. V. Duong, H. H. Bui, D. Q. Phung, and S. Venkatesh. Activity recognition and abnormality detection with the switching hidden semi-Markov model. In *IEEE Computer Society Conf. Computer Vision and Pattern Recognition*, volume 1, pages 838–845, San Diego, California, USA, 2005. IEEE Computer Society.

J. D. Ferguson. Variable duration models for speech. In *Proc. Symp. Application of Hidden Markov Models to Text and Speech*, pages 143–179, 1980.

M. C. Ferreira de Oliveira and H. Levkowitz. From visual data exploration to visual data mining: A survey. *IEEE Trans. Visualization and Computer Graphics*, 9(3): 378–394, July/September 2003.

S. Fine, Y. Singer, and N. Tishby. The hierarchical hidden Markov model: Analysis and applications. *Machine Learning*, 32(1):41–62, July 1998.

G. D. Forney, Jr. The Viterbi algorithm. *Proc. IEEE*, 61(3):268–278, 1973.

B. Goethals. Survey on frequent pattern mining. Technical report, HIIT Basic Research Unit, Department of Computing, University of Helsinki, 2003.

S. Greenhill, S. Venkatesh, and G. West. Adaptive model for foreground extraction in adverse lighting conditions. In C. Zhang, H. W. Guesgen, and W. K. Yeap, editors, *Proc. 8th Pacific Rim Int'l Conf. Artificial Intelligence*, volume 3157 of *Lecture Notes in Computer Science*, pages 805–811, Auckland, New Zealand, August 2004. Springer Berlin / Heidelberg.

J. Han, J. Pei, and Y. Yin. Mining frequent patterns without candidate generation. In W. Chen, J. F. Naughton, and P. A. Bernstein, editors, *Proc. 2000 ACM SIGMOD Int'l Conf. Management of Data*, pages 1–12, Dallas, Texas, USA., May 2000. ACM.

J. Han, J. Pei, Y. Yin, and R. Mao. Mining frequent patterns without candidate generation: A frequent-pattern tree approach. *Data Mining and Knowledge Discovery*, 8(1):53–87, January 2004.

A. G. Hauptmann, J. Gao, R. Yan, Y. Qi, J. Yang, and H. D. Wactlar. Automated analysis of nursing home observations. *IEEE Pervasive Computing*, 3(2): 15–21, April/June 2004.

H. Hofmann, A. P. J. M. Siebes, and A. F. X. Wilhelm. Visualizing association rules with interactive mosaic plots. In *Proc. 6th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining*, pages 227–235, Boston, Massachusetts, USA, August 2000. ACM.

J. D. Holt and S. M. Chung. Mining association rules using inverted hashing and pruning. *Information Processing Letters*, 83(4):211–220, August 2002.

F. Höppner and F. Klawonn. Finding informative rules in interval sequences. *Intelligent Data Analysis*, 6(3):237–255, 2002.

S. S. Intille, K. Larson, J. S. Beaudin, J. Nawyn, E. M. Tapia, and P. Kaushik. A living laboratory for the design and evaluation of ubiquitous computing technologies. In *Conf. Human Factors in Computing Systems*, pages 1941–1944, Portland, Oregon, USA, April 2005. ACM Press.

Y. A. Ivanov and A. F. Bobick. Recognition of visual activities and interactions by stochastic parsing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):852–872, August 2000.

R. E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–45, 1960.

C. D. Kidd, R. Orr, G. D. Abowd, C. G. Atkeson, I. A. Essa, B. MacIntyre, E. D. Mynatt, T. Starner, and W. Newstetter. The aware home: A living laboratory for ubiquitous computing research. In N. A. Streitz, J. Siegel, V. Hartkopf, and S. Konomi, editors, *Second Int'l Workshop Cooperative Buildings. Integrating Information, Organizations, and Architecture*, volume 1670 of *Lecture Notes in Computer Science*, pages 191–198, Pittsburgh, USA, October 1999. Springer.

S. E. Levinson. Continously variable duration hidden Markov models for speech analysis. *Computer Speech and Language*, 1(1):29–45, March 1986.

G. Liu, H. Lu, Y. Xu, and J. X. Yu. Ascending frequency ordered prefix-tree: Efficient mining of frequent patterns. In *Proc. Eighth Int'l Conf. Database Systems for Advanced Applications*, pages 65–72, Kyoto, Japan, March 2003. IEEE Computer Society.

G. Liu, H. Lu, W. Lu, Y. Xu, and J. X. Yu. Efficient mining of frequent patterns using ascending frequency ordered prefix-tree. *Data Mining and Knowledge Discovery*, 9 (3):249–274, November 2004.

J. Liu, Y. Pan, K. Wang, and J. Han. Mining frequent item sets by opportunistic projection. In *Proc. Eighth ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining*, pages 229–238, Edmonton, Alberta, Canada, July 2002. ACM.

H. Lu, J. Han, and L. Feng. Stock movement prediction and n-dimensional inter-transaction association rules. In *Proc. 1998 SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery*, pages 12:1–12:7, Seattle, Washington, USA, June 1998.

H. Lu, L. Feng, and J. Han. Beyond intra-transaction association analysis: Mining multi-dimensional inter-transaction association rules. *ACM Trans. Information Systems*, 18(4):423–454, October 2000.

H. Mannila and H. Toivonen. Discovering generalized episodes using minimal occurrences. In E. Simoudis, J. Han, and U. M. Fayyad, editors, *Proc. Second Int'l Conf. Knowledge Discovery and Data Mining*, pages 146–151, Portland, Oregon, USA, 1996. AAAI Press.

H. Mannila, H. Toivonen, and A. I. Verkamo. Efficient algorithms for discovering association rules. In U. M. Fayyad and R. Uthurusamy, editors, *Proc. AAAI Workshop on Knowledge Discovery in Databases*, pages 181–192, Seattle, Washington, USA, 1994. AAAI Press.

H. Mannila, H. Toivonen, and A. I. Verkamo. Discovering frequent episodes in sequences. In *Proc. First Int'l Conf. Knowledge Discovery and Data Mining*, pages 210–215, Montréal, Canada, August 1995. AAAI Press.

H. Mannila, H. Toivonen, and A. I. Verkamo. Discovery of frequent episodes in event sequences. *Data Mining and Knowledge Discovery*, 1(3):259–289, September 1997.

C. D. Mitchell and L. H. Jamieson. Modelling duration in a hidden Markov model with the exponential family. In *IEEE Int'l Conf. Acoustics, Speech, and Signal Processing*, volume 2, pages 331–334, Minneapolis, Minnesota, USA, April 1993.

M. C. Mozer. Lessons from an adaptive house. In D. Cook and S. Das, editors, *Smart Environments: Technology, Protocols and Applications*, pages 271–294. John Wiley & Sons, 2004.

K. P. Murphy and M. A. Paskin. Linear time inference in hierarchical HMMs. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Proc. 2001 Neural Information Processing Systems (NIPS) Conf.*, volume 14 of *Advances in Neural Information Processing Systems*, pages 833–840, Vancouver, British Columbia, Canada, 2001. MIT Press.

E. D. Mynatt, I. Essa, and W. Rogers. Increasing the opportunities for aging in place. In *ACM Conf. Universal Usability*, pages 65–71, Arlington, Virginia, USA, November 2000. ACM.

N. T. Nguyen, S. Venkatesh, G. West, and H. H. Bui. Hierarchical monitoring of people's behaviours in complex environments using multiple cameras. In *Proc. Int'l Conf. Pattern Recognition*, volume 1, pages 13–16, Québec, Canada, August 2002.

M. Okabe and K. Ito. How to make figures and presentations that are friendly to color blind people, November 2002. Retrieved October 2005 from `http://jfly.iam.u-tokyo.ac.jp/color/`.

N. Oliver, E. Horvitz, and A. Garg. Layered representations for human activity recognition. In *Fourth IEEE Int'l Conf. Multimodal Interfaces*, pages 3–8, Pittsburgh, Pennsylvania, USA, October 2002.

N. M. Oliver, B. Rosario, and A. P. Pentland. A Bayesian computer vision system for modeling human interactions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):831–843, August 2000.

K. Ong, K.-L. Ong, W.-K. Ng, and E.-P. Lim. CrystalClear: Active visualization of association rules. In *ICDM 2002 Workshop on Active Mining*, 2002.

J. S. Park, M.-S. Chen, and P. S. Yu. Using a hash-based method with transaction trimming for mining association rules. *IEEE Trans. Knowledge and Data Engineering*, 9 (5):813–825, September/October 1997.

Y. K. Park, C. K. Un, and O. W. Kwon. Modeling acoustic transitions in speech by modified hidden Markov models with state duration and state duration-dependent observation probabilities. *IEEE Trans. Speech and Audio Processing*, 4(5):389–392, September 1996.

J. Pei, J. Han, H. Lu, S. Nishio, S. Tang, and D. Yang. H-Mine: Hyper-structure mining of frequent patterns in large databases. In N. Cercone, T. Y. Lin, and X. Wu, editors, *Proc. 2001 IEEE Int'l Conf. Data Mining*, pages 441–448, San Jose, California, USA, November 2001. IEEE Computer Society.

A. Pentland and A. Liu. Towards augmented control systems. In *IEEE Intelligent Vehicles Symposium*, pages 350–355, Detroit, Michigan, USA, September 1995.

D. Q. Phung. *Probabilistic and Film Grammar Based Methods for Content Understanding*. PhD thesis, Department of Computer Science, Curtin University of Technology, January 2005.

D. Q. Phung, H. H. Bui, and S. Venkatesh. Content structure discovery in educational videos using shared structures in the hierarchical hidden markov models. In A. L. N. Fred, T. Caelli, R. P. W. Duin, A. C. Campilho, and D. de Ridder, editors, *Joint IAR Int'l Workshops on Syntactical and Structural Pattern Recognition, and Statistical Pattern Recognition*, volume 3138 of *Lecture Notes in Computer Science*, pages 1155–1163, Lisbon, Portugal, August 2004. Springer-Verlag.

L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, February 1989.

M. J. Russell and R. K. Moore. Explicit modelling of state occupancy in hidden Markov models for automatic speech recognition. In *Int'l Conf. Acoustics Speech and Signal Processing*, pages 5–8, Tampa, Florida, USA, March 1985.

A. Savasere, E. Omiecinski, and S. Navathe. An efficient algorithm for mining association rules in large databases. In U. Dayal, P. M. D. Gray, and S. Nishio, editors, *Proc. 21th In'tl Conf. Very Large Data Bases*, pages 432–444, Zurich, Switzerland, September 1995. Morgan Kaufmann.

B. Sin and J. H. Kim. Nonstationary hidden Markov model. *Signal Processing*, 46(1): 31–46, September 1995.

A. Sixsmith and N. Johnson. A smart sensor to detect the falls of the elderly. *IEEE Pervasive Computing*, 3(2):42–47, April/June 2004.

M. Skounakis, M. Craven, and S. Ray. Hierarchical hidden Markov models for information extraction. In G. Gottlob and T. Walsh, editors, *Proc. the Eighteenth Int'l Joint Conf. Artificial Intelligence*, pages 427–433, Acapulco, Mexico, August 2003. Morgan Kaufmann.

R. Srikant and R. Agrawal. Mining sequential patterns: Generalizations and performance improvements. In P. M. G. Apers, M. Bouzeghoub, and G. Gardarin, editors, *Proc. 5th Int'l Conf. Extending Database Technology: Advances in Database Technology*, volume 1057 of *Lecture Notes In Computer Science*, pages 3–17. Springer, 1996.

R. Srikant, Q. Vu, and R. Agrawal. Mining association rules with item constraints. In D. Heckerman, H. Mannila, and D. Pregibon, editors, *Proc. Third Int'l Conf. Knowledge Discovery and Data Mining*, pages 67–73, Newport Beach, California, USA, August 1997. AAAI Press.

C. Stauffer and W. E. L. Grimson. Learning patterns of activity using real-time tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):747–757, August 2000.

P.-N. Tan, V. Kumar, and J. Srivastava. Selecting the right interestingness measure for association patterns. In *Proc. Eighth ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining*, pages 32–41. ACM, 2002.

E. M. Tapia, S. S. Intille, and K. Larson. Activity recognition in the home using simple and ubiquitous sensors. In A. Ferscha and F. Mattern, editors, *Proc. Second Int'l Conf. Pervasive Computing*, volume 3001 of *Lecture Notes in Computer Science*, pages 158–175, Linz/Vienna, Austria, April 2004. Springer Berlin / Heidelberg.

G. Theocharous. *Hierarchical Learning and Planning in Partially Observable Markov Decision Processes*. PhD thesis, Department of Computer Science and Engineering, 2002.

H. Toivonen. Sampling large databases for association rules. In *Proc. 22nd Int'l Conf. Very Large Data Bases*, pages 134–145, San Francisco, California, USA, 1996. Morgan Kaufmann Publishers Inc.

A. K. H. Tung, H. Lu, J. Han, and L. Feng. Breaking the barrier of transactions: Mining inter-transaction association rules. In *Proc. Fifth ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining*, pages 297–301. ACM, August 1999.

A. K. H. Tung, H. Lu, J. Han, and L. Feng. Efficient mining of intertransaction association rules. *IEEE Trans. Knowledge and Data Engineering*, 15(1):43–56, January/February 2003.

United Nations Population Division. *World Population Ageing 1950-2050.* United Nations, New York, 2002.

S. V. Vaseghi. State duration modelling in hidden Markov models. *Signal Processing*, 41(1):31–41, January 1995.

P. C. Wong, P. Whitney, and J. Thomas. Visualizing association rules for text mining. In *IEEE Sym. Information Visualization*, pages 120–123, San Francisco, California, USA, October 1999. IEEE Computer Society.

L. Xie, S.-F. Chang, A. Divakaran, and H. Sun. Unsupervised discovery of multilevel statistical video structures using hierarchical hidden Markov models. In *Proc. 2003 Int'l Conf. Multimedia & Expo*, volume 3, pages 29–32, July 2003.

J. Yamato, J. Ohya, and K. Ishii. Recognizing human action in time-sequential images using hidden Markov model. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 379–385, Champaign, Illinois, USA, June 1992.

L. Yang. Pruning and visualizing generalized association rules in parallel coordinates. *IEEE Trans. Knowledge and Data Engineering*, 17(1):60–70, January 2005.

J. M. Zacks and B. Tversky. Event structure in perception and conception. *Psychological Bulletin*, 127(1):3–21, January 2001.

M. J. Zaki. Scalable algorithms for association mining. *IEEE Trans. Knowledge and Data Engineering*, 12(3):372–390, May/June 2000.

M. J. Zaki, S. Parthasarathy, M. Ogihara, and W. Li. New algorithms for fast discovery of association rules. In D. Heckerman, H. Mannila, D. Pregibon, and R. Uthurusamy, editors, *Int'l Conf. Knowledge Discovery and Data Mining*, pages 283–296, Newport Beach, California, USA, August 1997. AAAI Press.