Practical Probabilistic Inference

By

Quaid Donald Jozef Morris

Submitted to the Department of Brain and Cognitive Sciences
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Computational Neuroscience

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
[February 2003]
November 2002

Author..
Department of Brain and Cognitive Sciences
November 18, 2002

Certified by..
Whitman Richards
Professor of Cognitive Science
Thesis Supervisor

Accepted by..
Earl K. Miller
Professor of Neuroscience
Chairman, Department Graduate Committee

Practical Probabilistic Inference

by

Quaid Donald Jozef Morris

Submitted to the Department of Brain and Cognitive Sciences
on November 18, 2002, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Computational Neuroscience

Abstract

The design and use of expert systems for medical diagnosis remains an attractive goal. One such system, the Quick Medical Reference, Decision Theoretic (QMR-DT), is based on a Bayesian network. This very large-scale network models the appearance and manifestation of disease and has approximately 600 unobservable nodes and 4000 observable nodes that represent, respectively, the presence and measurable manifestation of disease in a patient. Exact inference of posterior distributions over the disease nodes is extremely intractable using generic algorithms. Inference can be made much more efficient by exploiting the QMR-DT's unique structure. Indeed, tailor-made inference algorithms for the QMR-DT efficiently generate exact disease posterior marginals for some diagnostic problems and accurate approximate posteriors for others.

In this thesis, I identify a risk with using the QMR-DT disease posteriors for medical diagnosis. Specifically, I show that patients and physicians conspire to preferentially report findings that suggest the presence of disease. Because the QMR-DT does not contain an explicit model of this reporting bias, its disease posteriors may not be useful for diagnosis. Correcting these posteriors requires augmenting the QMR-DT with additional variables and dependencies that model the diagnostic procedure.

I introduce the diagnostic QMR-DT (dQMR-DT), a Bayesian network containing both the QMR-DT and a simple model of the diagnostic procedure. Using diagnostic problems sampled from the dQMR-DT, I show the danger of doing diagnosis using disease posteriors from the unaugmented QMR-DT. I introduce a new class of approximate inference methods, based on feed-forward neural networks, for both the QMR-DT and the dQMR-DT. I show that these methods, recognition models, generate accurate approximate posteriors on the QMR-DT, on the dQMR-DT, and on a version of the dQMR-DT specified only indirectly through a set of presolved diagnostic problems.

Thesis Supervisor: Peter Dayan
Title: Professor, University College London

Acknowledgments

I would like to thank my advisor Peter Dayan for his support and guidance. I cannot hope to list all of Peter's many and valuable contributions. I am especially indebted to Peter for his rigour, precision, and patience. Peter is, and will continue to be, an inspiration.

I would also like to thank the members of my graduate committee: Tommi Jaakkola, Whitman Richards, and Sebastian Seung. Whitman has been invaluable in helping to bring this thesis to completion.

I would like to thank Geoffrey Hinton who played a very crucial role. This work began as a project with him. He also provided me with valuable encouragement.

I spent three wonderful years at the Gatsby Unit. I would like to thank the members of the Gatsby for intellectual stimulation and great distraction. In particular, I would like to thank Yee Whye Teh who gave me valuable insight and read and commented on parts of my thesis. Sam Roweis has also borne much of the burden of my enthusiasm.

Brendan Frey graciously provided me with an office and a laptop to finish my thesis. denise heintze provided valuable administrative support.

I would like to thank some friends who have supported me and made life more fun: the Gatsby Unit, the Greater Clinton Manor (especially Raja Bhattacharyya), Onil Bhattacharyya, Chua Siok Peng, Devon Curtis, Janet Papadakos, and Timothy Wu.

Finally, I would like to thank my family, especially my mother, for their support and love.

While a graduate student, I was supported by the Department of Brain and Cognitive Sciences at MIT and by the Gatsby Computational Neuroscience Unit.

Contents

1 Introduction **23**

1.1 Motivation . . . 23

1.2 Summary . . . 25

2 Foundations **29**

2.1 Introduction . . . 29

2.2 Probabilistic Models . . . 30

2.3 Probabilistic Inference . . . 30

2.4 Graphical Models . . . 32

2.5 Exact Inference . . . 33

2.6 Approximate Inference . . . 33

2.6.1 Stochastic Methods . . . 34

2.6.2 Deterministic Methods . . . 35

2.7 Recognition Models . . . 36

2.8 Observation Processes . . . 38

3 The QMR-DT **41**

3.1 Introduction . . . 41

3.2 Structure and Parameterisation . . . 43

3.3 BN2O Network Inference Simplifications . . . 45

3.4 Sparsity in the QMR-DT . . . 53

3.5 Quickscore . . . 54

3.5.1 Introduction . . . 54

3.5.2 Recursive Quickscore 56

3.6 Structural Quickscore 58

3.6.1 Introduction 58

3.6.2 Factoring in Disconnected BN2O networks 59

3.6.3 Description of the Algorithm 60

3.7 Evaluating Structural Quickscore 64

3.7.1 Introduction 64

3.7.2 Evaluation Techniques 65

3.7.3 Experiments 66

3.8 Approximate Inference in the QMR-DT 70

3.8.1 Introduction 70

3.9 Summary 72

4 Recognition Models for Approximate Inference 73

4.1 Introduction 73

4.2 General Framework for Recognition Models 74

4.3 Examples of Recognition Models 76

4.3.1 Multilayer Perceptrons 76

4.3.2 Mixtures of Experts 77

4.4 Concise Input Encoding 78

4.5 Training 80

4.5.1 Introduction 80

4.5.2 Stochastic Meta-Descent 81

4.5.3 Logistic Regression Networks 82

4.5.4 Multilayer Perceptrons 86

4.6 Experiments 87

4.6.1 Benchmark Inference Algorithm 87

4.6.2 Evaluation Techniques 89

4.6.3 Results 95

4.7 Discussion 97

5 Recognition Models for Medical Diagnosis 99

5.1 Introduction . . . 99

5.2 Observation Processes for Medical Diagnosis . . . 101

5.2.1 Introduction . . . 101

5.2.2 Diagnostic Procedure . . . 102

5.2.3 Evidence for Non-ignorability . . . 105

5.3 The Diagnostic QMR-DT . . . 106

5.3.1 Introduction . . . 106

5.3.2 General Framework . . . 107

5.3.3 Parameterisation of $P(\Phi)$ . . . 108

5.4 Evaluation Techniques . . . 110

5.4.1 Introduction . . . 110

5.4.2 Diagnoses Lists . . . 110

5.4.3 Scoring Diagnoses Lists . . . 111

5.4.4 Generating Diagnoses Lists . . . 112

5.4.5 Posterior Mass Ratio . . . 113

5.4.6 Test Sets . . . 113

5.4.7 Discussion . . . 114

5.5 Known observation processes . . . 114

5.5.1 Introduction . . . 114

5.5.2 Preliminaries . . . 115

5.5.3 Results . . . 118

5.5.4 Discussion . . . 124

5.6 Weakly-specified observation processes . . . 126

5.6.1 Introduction . . . 126

5.6.2 Problem Definition . . . 127

5.6.3 Training . . . 128

5.6.4 Intuition . . . 129

5.6.5 Experiments . . . 130

5.6.6 Discussion . . . 135

5.7 Discussion 136

6 Discussion **139**

A Creating the aQMR-DT **145**

A.1 Introduction 145

A.2 The Anonymised QMR Knowledge Base 145

A.3 Calculating Θ 147

A.4 Generating p_k 147

A.5 Calculating $\boldsymbol{\theta}_0$ 148

A.6 Conclusions 150

B Information Loss in the Concise Encoding **151**

B.1 Introduction 151

B.2 Indistinguishable Evidence Vectors 151

B.3 Evaluating Information Loss 153

B.4 Discussion 156

List of Figures

3-1 Encoding of non-binary manifestations in the QMR-DT. A) Example encoding of a non-binary valued categorical manifestation. B) Example encoding of a continuous valued manifestation. 44

3-2 Structure of the QMR-DT. 44

3-3 Graph transformations in QMR-DT inference. I use different shades of outlines to indicate categorical differences between the disease nodes. Disease nodes with differently shaded outlines have priors that incorporate different pieces of evidence. 46

3-4 Empirical distribution of number of active diseases. The bar graph shows two distributions. The black bars are the empirical distribution generated using 10^6 samples from $P(\boldsymbol{d})$. No sample contained more than nine active diseases. The grey bars show, for comparison, a Poisson distribution with the same mean. 52

3-5 Empirical distributions of number of positive manifestations. A) Number of positive manifestations. Empirical distribution was generated using 10^6 samples. B) Number of positive manifestations conditioned on number of active diseases. Each curve is marked with the number of active diseases conditioned on. Each distribution was generated using 10^5 samples. 52

3-6 Recursion tree of a call to Quickscore. Only the elimination and reduction steps are shown. The shaded disease node borders have the same interpretation as in figure 3-3. Note that the positive manifestations are eliminated in the same order along each branch of the tree and that the graph becomes disconnected after the first elimination. Section 3.6 describes an algorithm that exploits disconnectedness to speed-up Quickscore. 55

3-7 Pseudocode for the second stage of structural Quickscore 62

3-8 Comparison of elimination orderings. Figure shows results of elimination ordering comparisons on each of the four test sets. Results are shown with box plots. The center line in each box shows the median. The upper and lower lines are the upper and lower quartiles. The whiskers are the extent of the rest of the data up to a maximum distance away from the median. Points more than 1.5 times the interquartile distance away from the median are displayed with the square symbol. Each column represents a single inference method. QS, #P, GA, GB stand for Quickscore, *most parents*, *greedy A*, and *greedy B* respectively. 67

3-9 Savings using greedy B. A) The savings on all four datasets resulting from using greedy B. The savings is the difference between the number of positive manifestations and the equivalent number of positive manifestations under greedy B. B) Boxplot of the savings on each of the test sets. The interpretation of boxplots is described in figure 3-8. . . 68

4-1 Architecture of the multilayer perceptron-based recognition model. The arrows between the layers indicate a weight matrix that fully connects the layers. The input layer contains an additional bias unit that is always on. The hidden layer non-linearity is the tanh function. The output layer non-linearity is the logistic function. 76

4-2 Architecture of the mixtures-of-expert-based recognition model. The arrows between the layers indicate a weight matrix that fully connects the layers. The input layer contains an additional bias unit that is always on. The $\boldsymbol{\pi}$ layer units are connected together, indicating that their values are tied via the multinomial logistic regression. 77

4-3 Structure of the LR network recognition models for the QMR-DT. Arrows connecting boxes indicate fully connectivity between the units in the boxes. The input vector $\boldsymbol{x}$ contains the *sparse*, *non-sparse* and *bias* input units. The bias weight vector $\boldsymbol{w}_0$ is optimised using the same parameter settings as the non-sparse weights. 83

4-4 Architecture of the MLP trained on the QMR-DT. Arrows connecting boxes indicate fully connectivity between the units in the boxes. The input vector $\boldsymbol{x}$ contains the *sparse*, *non-sparse* and *bias* input units. The lightly shaded weight matrices W^s and W^n are fixed during training. Optimisation of the bias weight vector $\boldsymbol{w}_0$ copied from the LR network restarts using initial SMD parameter settings $(\mu_u, P_u^{\text{init}})$. Another bias unit (not shown) provides an additional input to the hidden layer. These bias weights use the same SMD parameter settings as the V weight matrix. 85

4-5 Performance comparison of LR and MLP-based recognition models against the variational method (V) adapted from [21] and an adaptive importance sampling method (AIS-BN) [5] on two test sets. The MLP and LR50 models were trained on partially visible samples, the LR100 model was trained on fully visible evidence vectors. All recognition models treat negative findings as being unobserved. In (A) the state of each manifestation in the test set was hidden with probability 0.5. In (B) posterior marginals for AIS-BN were unavailable. Bars in (A) and (B) show the mean bits of cross entropy (averaged over 1000 samples) between the reference diagnoses and the approximate posteriors generated by each method. The error bars show the upper bound of the symmetric 95% confidence interval of the mean. All differences in the graphs, except those between LR50 and MLP, are significant. . . 91

4-6 Performance comparison of LR network and MLP-based recognition models against V on two test sets. The LR-X models were trained on samples with $\phi = X/100$. The MLPX models were trained on samples with $\phi = 0.5$ and had X hidden units. Each subfigure represents a different test set of 3000 samples, the value of ϕ used to generate the test sets is given in the title bar. Bars in each subfigure show the mean bits of cross entropy averaged over the 3000 samples, between each sample's disease node configuration and the approximate posteriors generated given the sample's observed manifestations. The error bars show the upper bound of the symmetric 95% confidence interval of the mean. 92

4-7 Performance comparison of LR network and MLP-based recognition models against V on two test sets. See caption of figure 4-6 for the interpretation of the figure. 93

4-8 Performance comparison of LR network and MLP-based recognition models on $\phi = 0.5$ test set. Figure shows cross entropy of each method averaged across 10^5 samples from the QMR-DT with $\phi = 0.5$ The error bars show the 95% confidence interval of the mean. Note that the 1000 hidden unit network is still far from convergence. The cross-entropy of this network should decrease below that of the 100 unit network given more training time. 94

5-1 Diagnostic QMR-DT specification. A) Graphical models of the diagnostic QMR-DT (dQMR-DT). Squares indicate discrete-valued variables, circles are continuous valued variables. All variables shown in (A) are multivariate. The distributions $P(\boldsymbol{D})$ and $P(\boldsymbol{F}|\boldsymbol{D})$ are prespecified by the QMR-DT. The observation process is conditioned on $\boldsymbol{\Phi}$ which has density $P(\boldsymbol{\Phi})$. B) A close-up of the observation process. The variables $\{E_1, E_2, \ldots, E_I\}$ and $\{F_1, F_2, \ldots, F_I\}$ are the elements of the random vectors $\boldsymbol{E}$ and $\boldsymbol{F}$ respectively. The dependence of $\{F_1, F_2, \ldots, F_I\}$ on $\boldsymbol{D}$ is not shown. C) The conditional probability table (CPT) of each finding. The dependence of $P(F_i|E_i, \Phi)$ on Φ is through the parameters (ϕ_i^+, ϕ_i^-) of the CPT. 107

5-2 Locations of eight simple observation processes. Each point in the (q, r)-space maps into a value of $\boldsymbol{\phi}(q, r)$. The letters (B-E, H-K) label the eight observation processes used to train the LR networks and evaluate the inference algorithms. 116

5-3 Performance of various inference methods on observation processes B,C,D,E. For each approximate inference method and each observation process, I compare three different ways of selecting 20 diagnoses out of the algorithm's basic 1000 diagnoses list. Further description is provided in the text. The title of each of the eight subfigures is the approximate inference algorithm used to generate the diagnoses lists analysed in the figure. Each row of subfigure matrix is labelled with the observation process analysed in the row. 119

5-4 Performance of the LR networks and V on observation processes H,I,J,K. See figure 5-3 and the text for a description of this figure. 120

5-5 Comparison of the expected number of diseases under the dQMR-DT, QMR-DT, and LR network posteriors. The title of each subfigure identifies the observation process analysed. Each subfigure shows the expected number of active diseases under the three approximate posteriors over the 100 sample test sets described in section 5.5.2 for the given observation process. Note that the average probability that a positive manifestation will be observed increases from the lower left corner of the figure (H) to the upper right hand corner (K). For each sample, the dQMR and the QMR posteriors were derived from the scored diagnoses pool (where diagnoses were scored using the dQMR-DT and the QMR-DT respectively) using equation (5.6). Note that because of the sparse disease prior, observation processes which produce fewer findings (e.g. H) have an expected number of diseases that is less than five, despite the five active diseases in the reference diagnoses. 121

5-6 Comparison of the ranking of disease posterior marginals under the dQMR-DT and QMR-DT. The title of each subfigure identifies observation process analysed. Each subfigures compares the average rank among the QMR-DT posterior marginals of the 10 diseases with the highest posterior marginals under the dQMR-DT on the 100 sample test sets described in section 5.5.2. Specifically, the false positive value corresponding to true positive value N is $N' - N$ where N' is the average size of the list of the highest ranking diseases under the QMR-DT that contains all of the N highest ranking diseases under the dQMR-DT. The QMR-DT and dQMR-DT disease posterior marginals were calculated using the posteriors implied by the appropriate scored diagnoses pools. These posterior marginals are approximations to the true posterior marginals. 122

5-7 Learning curves for gating networks. Each plot shows the curves for the training and validation sets. Since the two sets are disjoint, the average error, measured in cross-entropy bits per example, is different in the two sets. A) The learning curves for the input-dependent gating network. B) The learning curves for the fixed mixture. In both (A) and (B), an arrow marks the final network chosen. 131

5-8 Grid points used in test set. Each point in the (q, r)-space maps into a set of dQMR-DT parameters $\boldsymbol{\phi}(q, r)$. The basic test set contains 1,000 samples from $P(\boldsymbol{d}, \boldsymbol{e}|\boldsymbol{\phi}(q, r))$ for each of the 81 grid points, 81,000 $(\boldsymbol{d}, \boldsymbol{e})$ pairs in all. The challenging test set contains 100 samples from $P(\boldsymbol{d}, \boldsymbol{e}|\boldsymbol{\phi}(q, r), \sum_k d_k = 5)$ for each of the 81 grid points. The letters (B-E, H-K) label the locations of the parameter settings for each of the eight LR networks. The points {C, E, H, J} shown in italics correspond to the provided values of $\boldsymbol{\phi}^m$ and the LR networks whose outputs are combined by the gating network. 132

5-9 Bit loss images. The ten images in this figure show the performance of the gating network (GATE), the learned mixture (MIX) and the individual experts (B-E, H-K). Each image contains 81 pixels, each pixel corresponding to one of the 81 grid points in figure 5-8. The spatial arrangement of the pixels matches that of the grid points, i.e. the pixel in the upper left hand corner of each image shows the performance on samples from the observation process $P(\boldsymbol{e}|\boldsymbol{f}, \boldsymbol{\phi}(0.05, 0.45)$. Each pixel shows the bit loss of the expert (or gating network) on a test set drawn from the corresponding observation process. The labels of the four experts combined by the gating network, $\{C, E, H, J\}$, are shown in italics. A method's bit loss is the difference between its mean cross-entropy on the test set and the cross-entropy of the benchmark method for that test set. The benchmark method for a grid point is the expert with the lowest mean cross-entropy on that grid point's test set. 133

5-10 Distributions of posterior mass ratios. The plots compare the benchmark to the gating network, the fixed mixture and the individual expert, C, among the four with the best performance. Each plot shows the distribution of the mean posterior mass ratios across the 81 grid points. A) Box plot of the distribution of mean PMRs. The center line in each box is the median of the 81 means. The upper and lower lines show the upper and lower quartiles. The whiskers show the extent of the rest of the data up to a maximum distance away from the median. Points more than 1.5 times the interquartile distance away from the median are displayed with the square symbol. Note that though the median is high for expert C, it has a number of mean PMRs less than 0.5. B) Histogram of mean PMRs. Each row in this plot is composed of two histograms, one pointing upward and the other pointing downward. The upward pointing histogram shows the distribution of posterior mass ratios for the top 20 diagnoses in each list. The downward pointing histogram shows the distribution for the whole list of 100 diagnoses. Each histogram contains ten equally spaced bins. . . 134

A-1 The distribution of the values used to generate the random priors. . 146

A-2 Evoking strengths and upper bounds. A) The distribution of upper bounds, u_{ik} on estimates of the evoking strengths, $\hat{e}_{ik}$. Notice the spike at $u_{ik} = 1$, these are cases where F_i only has a single parent. B) The distribution of evoking strengths e_{ik} in the aQMR KB. 146

B-1 Indistinguishable observations under the concise encoding. A) A simple BN2O network. B) Concise encodings of three different evidence vectors. See section 4.4 for a description of the two types of inputs. Note that the final two evidence vectors are indistinguishable under the concise encoding. However, since $\phi_1^- \neq \phi_2^-$, disease posteriors given each of the vectors are different. 152

B-2 Comparison of the lossless versus the concise input encoding for recognition models. Figures compare learning curves from the training of the lossless versus the concise recognition models on a dQMR-DT where $P(\Phi = \phi') = \delta(\phi' - \phi(0.2, 0.2))$. Curves are exponential traces of an estimate of the cross-entropy error for each model on untrained data. A) Cross-entropy versus number of training examples. B) Cross-entropy versus CPU time. C; D) Close-up of the tails of A and B respectively. The x-axis is scaled differently for A; B versus C; D. . . 154

List of Tables

4.1 Table comparing the lossless and concise encodings. The number of parameters in the LR networks is proportional to the total number of inputs. Inputs representing positive manifestations are guaranteed to be sparse. Inputs representing negative manifestation or log odds are not. Notice that the concise encoding has less than half as many parameters as the lossless encoding and about an eighth as many non-sparse inputs. 79

4.2 Information on the training of the LR networks. This table gives the SMD parameter settings, the training times, and number of mini-batches used to train each of the LR networks. All network but LR-10 used mini-batches with 500 samples and had unbounded gains. The LR-10 network used smaller mini-batches (300 samples) and bounded gains. CPU times are measured on a 2.4 GHz Pentitum-4 processor with 512K cache. 90

4.3 Information on the training of the MLPs. This table gives the SMD parameter settings, the training times, and the number of mini-batches used to train each of the multilayer perceptrons. All meta-gain step size parameters, μ, were set to 0.3. All MLPs used mini-batches with 500 samples and had bounded gains. CPU times are measured on a 1.4GHz Pentitum-III processor with 512K cache. 90

5.1 Table comparing realistic diagnostic problems to fully observed samples from the anonymised QMR-DT. The entries in the table are empirical means. The displayed margins of error are 95% confidence intervals of the mean. Note that I have analytically calculated the expected number of positive findings in a sample from the QMR-DT. SAM and CPC diagnoses contained exactly one disease and between one and six diseases respectively. The expected number of diseases under the QMR-DT prior is 1.04. MSH91, MPM82, SC91, and J97 stand for [34, 37, 63, 21] respectively. 105

5.2 Parameter settings for LR network SMD training algorithm. All networks were trained on 1.1×10^8 training examples using mini-batches containing 500 examples. These parameters were chosen to be similar to those used to train the LR networks in section 4.5.3. 115

Chapter 1

Introduction

1.1 Motivation

It is now feasible to build large-scale expert systems based on probabilistic domain models. Graphical models provide an intuitive representational language for building these models, allowing the easy extraction and representation of knowledge from non-technical experts. However, real-world systems require industrial strength inference algorithms. This thesis is about the design and evaluation of practical, domain-specific inference algorithms for these large-scale expert systems.

Two major issues arise when doing practical probabilistic inference. First, there are severe requirements on practical inference algorithms. These algorithms need to produce highly accurate answers in real time. In general, however, high accuracy only comes at the expense of substantial computation time. Though online computation time is scarce, ample offline computation time is available both before the expert system is deployed and between uses of the system. This resource is rarely exploited by existing inference algorithms. The second issue is that the probabilistic domain models of the expert systems are often incomplete. These models concentrate on representing knowledge about how unobservable causes, like diseases, give rise to observable effects, like symptoms. However, they often ignore the process that selects which subset of the observable effects are actually observed. It is perilous to ignore this observation process since, if a selection bias exists in the choice of observables, then predictions made using the incomplete domain model can be highly inaccurate. Often, however, augmenting an existing large-scale domain model with an observation process model makes an already computationally-demanding inference problem much more difficult.

This thesis presents strategies for dealing with both these issues. First it suggests using recognition models to perform inference in large-scale expert systems. Recognition model methods have a number of advantages over competing methods: they make substantial use of offline computations to enable fast, accurate inference online and unlike methods that need to be designed to exploit specific properties of the probability distributions in question, a recognition model may be generated for any probability distribution which supports efficient sampling.

In this thesis, I expand the framework of recognition models to include models of arbitrary complexity, thus allowing arbitrarily accurate inference. While these more complex models may take much longer to train, the online computation time is only marginally greater. I further demonstrate the importance of modelling the observation process by showing examples where failing to model this process has disastrous effects on inferential accuracy. Finally, I introduce and evaluate strategies for incorporating an observation process model into inference.

The Quick Medical Reference, Decision Theoretic (QMR-DT) belief network provides an ideal test bed for the strategies and algorithms discussed. This network is a widely-studied example of a large-scale expert system designed for practical use. Furthermore, as I will show, there is a significant selection bias in the present in the diagnostic procedure, and incorporating knowledge about this bias can significantly improve the accuracy of diagnostic inference done using QMR-DT. An anonymised version of this belief network has recently been made available, allowing other researchers to compare new inference algorithms against results presented in this thesis.

1.2 Summary

The QMR-DT plays a dual role in the literature. Primarily, it has become a benchmark problem for inference algorithms. One of the reasons that it has become a benchmark is due to its real-world application. Its intended real-world application is as a graphical model for medical diagnosis. In this thesis, I treat these two roles as distinct.

In its role as a benchmark, I use the QMR-DT to evaluate recognition models. This evaluation demonstrates that recognition models are feasible approximate inference methods for expert systems on the scale of the QMR-DT. Furthermore, it shows that increasing the complexity of the recognition models does indeed increase the accuracy of recognition model inference.

In its role as a diagnostic aid, I show that the QMR-DT itself is insufficient for medical diagnosis. In particular, the diagnostic procedure contains selection bias in the type of information that it reveals about the patient. I identify some of the properties of this bias and introduce a probabilistic observation process model with some of the same properties. I use this model in two ways: to demonstrate the types of errors that will occur when the observation process is ignored, and to evaluate techniques for doing inference in the presence of such a model.

It is, however, infeasible to assume that a fully-specified observation process model will be made available for the medical domain. Such a model would need to capture variation among patients, doctors, hospitals, and diagnostic protocols. This fully-specified observation model process would likely be harder to build than the Quick Medical Reference knowledge base (QMR KB), which itself took 25 years to bring to its present form. In the final part of the thesis, I introduce and evaluate techniques for diagnostic inference when the observation process model is only partially-specified, using information that is more likely to be available.

The thesis contains six chapters and two appendices.

Chapter 2 presents the background in probabilistic models and inference necessary to understand the contributions of this thesis. Readers already familiar with this material can skip chapter 2, except for section 2.7 which presents some background material that is less widely known. Section 2.1 describes generative models which are types of probabilistic model which model observable effects of unobservable causes. Section 2.2 introduces probabilistic inference as a method to extract information about the unobservable causes given the observed effects. Section 2.3 introduces the graphical model formalism, an intuitive graphical language for building probabilistic models. Algorithms for exact inference that make use of a graphical model are described in section 2.4. Section 2.5 describes stochastic and deterministic approximate inference algorithms for intractable probability models. A different type of approximate inference method, based on optimising a function approximator on samples from a generative model, is described in section 2.6. This method, called a recognition model, forms the basis of all the approximate inference algorithms described in this thesis. Section 2.7 provides necessary background on observation processes. It is critical to understand the distinction between an ignorable observation process, for which the selection of which observables are made available carries no information, and an non-ignorable one, for which this choice is informative.

Chapter 3 describes the QMR-DT and various inference algorithms associated with it. Section 3.2 describes the basic structure and parameterisation of the QMR-DT network. This same structure and parameterisation is common to all networks in a class called the binary two-layer noisy-OR networks (BN2O networks). Section 3.3 describes some techniques to simplify inference in BN2O networks that exploit their particular properties. Section 3.4 describes some properties particular to the QMR-DT, namely its concentration of probability mass on sparse configurations of variables. Section 3.5 describes the Quickscore algorithm, a surprisingly efficient exact inference algorithm that exploits both sparsity and BN2O network properties. Unfortunately, however, even with Quickscore, inference is still intractable for typical samples from the QMR-DT. Section 3.6 introduces a new algorithm, structural Quickscore, that extends Quickscore by exploiting the sparse connectivity of the QMR-DT. Section 3.7 evaluates the extent to which the new algorithm can enable exact inference in the QMR-DT. Unfortunately, though the new algorithm is orders of magnitude faster than Quickscore, it still does not make exact inference reliably tractable. Section 3.8 describes some basic techniques that have been used to approximate inference in the QMR-DT. The chapter is summarised in section 3.9.

Chapter 4 introduces and evaluates recognition models for approximate inference in the QMR-DT. Section 4.2 describes a general framework for recognition models and section 4.3 introduces two new types of recognition models that fit within this framework. One, described in section 4.3.1, is used in chapter 4, the other, described in section 4.3.2, is used in chapter 5. Section 4.4 describes a concise encoding of the observed manifestations that support faster optimisation of the recognition models on the QMR-DT. Section 4.5 details the training of the recognition models. Section 4.6 evaluates the accuracy of the approximate posteriors produced by the recognition models. These approximate posteriors are compared against another benchmark inference method. Section 4.7 summarises and discusses the results of the chapter.

Chapter 5 extends the recognition models introduced in chapter 4 to allow their use for medical diagnosis. Section 5.2 discusses observation processes that are appropriate for probabilistic models of the medical diagnostic domain. This section demonstrates that the diagnostic procedure embodies a non-ignorable observation process. This process, among other things, preferentially reveals positive manifestations of disease. Section 5.3 introduces the diagnostic QMR-DT. This probability model contains both the QMR-DT, which models the manifestation of disease, and an observation process model, which models the procedure by which manifestations are revealed. Section 5.4 presents a new set of metrics for medically-relevant evaluations of approximate inference algorithms. Section 5.5 uses these metrics to evaluate recognition models trained on samples from the dQMR-DT when the observation process is fully specified. Additionally, this section compares posteriors implied by the QMR-DT to the true posteriors, elucidating some of the inaccuracies resulting from ignoring the observation process. Section 5.6 considers diagnostic inference using recognition models when the observation process is only partially specified. In this section, the new recognition model described in section 4.3.2, is used to combine the predictions of other recognition models.

Chapter 6 summarises and discusses the results of this thesis.

Appendix A details how the QMR-DT used in this thesis was constructed from the anonymised QMR knowledge base and appendix B evaluates the information loss arising from using the concise encoding (described in section 4.4) for recognition models trained in chapter 5 on the non-ignorable observation processes.

Chapter 2

Foundations

2.1 Introduction

This literature survey provides both a review of previous work in the field and a brief introduction to probabilistic inference in general. Section 2.2 introduces probabilistic generative models and section 2.3 describes some issues relating to probabilistic inference in these models. One major issue is that in the worst case, probabilistic inference can be intractable. Section 2.4 introduces graphical models, a language for representing and building probabilistic models. Many exact and approximate inference algorithms make use of graphical models. In particular, section 2.5 briefly describes a prototypical exact inference algorithm that requires a graphical model. Exact inference, however, is often intractable for large multiply-connected graphical models, thus necessitating approximate inference. Section 2.6 gives a general overview of approximate inference methods. These methods may be broadly divided into stochastic methods, described in section 2.6.1, and deterministic methods, described in section 2.6.2. Section 2.7 describes recognition models, an alternative approximation method and the inference method on which this thesis is based. Finally, section 2.8 describes observation processes and presents techniques for determining when a selection bias must be taken into account.

2.2 Probabilistic Models

Knowledge about a domain can often usefully be represented using a probabilistic model. Domains typically include a set of observable effects and a set of unobservable (or hidden) causes. A probabilistic model of a domain specifies a joint probability distribution, $P(\boldsymbol{D}, \boldsymbol{F})$, over a random vector, $\boldsymbol{D}$, containing variables representing the causes and a random vector, $\boldsymbol{F}$, containing variables representing the effects, some or all of which are observed. In the medical domain, for example, $\boldsymbol{D}$ could contain variables specifying the presence or absence of particular diseases and $\boldsymbol{F}$ could contain variables representing the findings a physician may possibly make about a patient.

Often the joint distribution is most easily specified in the causal or generative direction, i.e. from causes to effects. A generative model for $\boldsymbol{D}$ and $\boldsymbol{F}$ requires specifying a prior distribution $P(\boldsymbol{D})$ over the causes and a likelihood function, $P(\boldsymbol{F} = \boldsymbol{f}|\boldsymbol{D})$, over configurations of $\boldsymbol{D}$ for each possible observation $\boldsymbol{F} = \boldsymbol{f}$.[1] Generative models of this form have been widely used both in unsupervised learning [9, 18, 41] and in neural modelling [45].

2.3 Probabilistic Inference

The process of probabilistic inference involves computing statistical information about the hidden variables implied by a particular observation, $\boldsymbol{F} = \boldsymbol{f}$. One obvious endpoint of inference is a posterior distribution over $\boldsymbol{D}$, $P(\boldsymbol{D}|\boldsymbol{F} = \boldsymbol{f})$. Another popular endpoint is the set of single-variable marginals of the posterior $\{P(D_k|\boldsymbol{F} = \boldsymbol{f})\}_{k=1}^{K}$ (hereafter called the *posterior marginals*). Finally, one may seek individual (or sets of) configurations that are optimal with respect to some loss function.

In the general case, if there are many variables in $\boldsymbol{D}$ then probabilistic inference is computationally intractable since all endpoints involve a summation over all possible configurations of the variables in $\boldsymbol{D}$. For example, the posterior probability, $P(\boldsymbol{d}|\boldsymbol{f})$[2]

[1] In this thesis, I will assume that all random variables under consideration are discrete-valued.

[2] To simplify notation, I will represent the event that a random variable (or vector) takes a particular value, e.g. $\boldsymbol{F} = \boldsymbol{f}$, by the value itself, e.g. $\boldsymbol{f}$. For example $P(\boldsymbol{d}|\boldsymbol{f})$ means $P(\boldsymbol{D} = \boldsymbol{d}|\boldsymbol{F} =$

can be written

$$P(\boldsymbol{d}|\boldsymbol{f}) = \frac{P(\boldsymbol{d})P(\boldsymbol{f}|\boldsymbol{d})}{\sum_{\boldsymbol{d}'} P(\boldsymbol{d}', \boldsymbol{f})}, \tag{2.1}$$

the summation in the denominator is over all possible configurations of $\boldsymbol{D}$. If $\boldsymbol{D}$ contains K binary random variables then the summation in equation (2.1) contains 2^K terms.

Finding an optimal configuration, $\boldsymbol{d}^*$, can be even harder. Given $\boldsymbol{f}$, the loss, $L(\boldsymbol{d})$, of a single configuration, $\boldsymbol{d}$, is

$$L(\boldsymbol{d}) = \sum_{\boldsymbol{d}'} P(\boldsymbol{d}', \boldsymbol{f}) C(\boldsymbol{d}', \boldsymbol{d}), \tag{2.2}$$

where $C(\boldsymbol{d}', \boldsymbol{d})$ measures the cost of predicting $\boldsymbol{d}'$ when the true answer is $\boldsymbol{d}$. Calculating this loss has at least the same worse-case time complexity as computing equation (2.1). In fact, finding $\boldsymbol{d}^* = \mathrm{argmin}_{\boldsymbol{d}} L(\boldsymbol{d})$ may require computing equation (2.2) several times.

Sometimes systematic patterns of marginal and conditional independence in $P(\boldsymbol{D}, \boldsymbol{F})$ can be exploited to make inference more efficient. If, for example, D_1 through D_K are mutually independent and the likelihood function for each variable, F_k, depends only upon D_k, then $P(\boldsymbol{d}, \boldsymbol{f})$ can be written

$$P(\boldsymbol{d}, \boldsymbol{f}) = \prod_k P(f_k|d_k)P(d_k).$$

This pattern of independence supports efficient inference. For example, the posterior marginal $P(d_k|\boldsymbol{f})$ may be written

$$P(d_k|\boldsymbol{f}) = \frac{P(f_k|d_k)P(d_k)}{\sum_s P(f_k|D_k = s)P(D_k = s)}. \tag{2.3}$$

Algorithms have been developed to save computation by exploiting patterns of independence relationships. Many of these algorithms depend ancillarily on a *graphical model* of the set of conditional independence relationships that hold in $P(\boldsymbol{D}, \boldsymbol{F})$.

$\boldsymbol{f}$), i.e. the posterior probability under P that $\boldsymbol{D} = \boldsymbol{d}$ given $\boldsymbol{F} = \boldsymbol{f}$.

2.4 Graphical Models

Graphical models are a representational language for probability distributions. A graphical model consists of a graph which describes the qualitative structure of the probability distribution and a set of functions that specify the quantitative structure. There is a one-to-one correspondence between the random variables of the distribution and the nodes in this graph. The connections between the nodes are chosen so that the graph structure represents a subset of the conditional independence relationships that hold in the distribution. In particular, graphical separation in the graph implies conditional independence in the distribution. Connections can be either directed or undirected. Most expert systems involve directed graphical models with discrete (often binary) random variables and it is on this class that this thesis concentrates.

In a directed graphical model, functions associated with each node specify the quantitative properties of the represented distribution. These function are conditional probability functions that specify a distribution over the node given a configuration of the node's parent. Often the functions take the form of tables in which each row, indexed by a configuration of the parents, contains a distribution. However since the number of rows in a table is exponential in the number of parents of the node, these tables are unwieldly in graphs containing nodes with many parents. The conditional probability function may instead be specified by a low dimensional function of the states of the parents. For binary (0/1) parents and children, the noisy-OR or the sigmoid function are often used.

Dual to the structure of a graphical model are computational procedures that perform exact or approximate inference on the underlying probability distribution [26]. Graph theoretic properties typically determine the nature and efficiency of these procedures. Many of these procedures involve propagating information between nodes via connections either in the original graphical model or in a graphical model derived from the original one.

2.5 Exact Inference

The prototypical exact inference algorithm based on the graphical model is junction tree propagation. This involves constructing a structure called a junction tree from the graphical model, on which probabilistic information can be locally propagated [30, 62]. Building a junction tree requires transforming the directed graph into an undirected graph (which usually contains additional implied dependencies) whose cliques possess the graph theoretic property called the running intersection property. This transformation includes two processes: triangulation and moralisation. Please consult [25] for further information on junction tree propagation. See [48] for other algorithms for exact inference on arbitrary graphical models.

The time complexity of junction tree propagation is exponential in the size of the largest clique in the moralised, triangulated graph. In sparsely connected graphs, the size of the largest clique may be much smaller than the number of nodes in the entire graph. Nonetheless, in many cases, the size of the largest clique is sufficient to render inference using junction tree propagation intractable. More efficient exact inference algorithms exist for graphs having a particular qualitative structure (e.g. [28, 56]) or having both a particular qualitative and quantitative structure (e.g. [15, 14, 52]) Often, however, tractability can only be achieved by approximating inference.

2.6 Approximate Inference

Stochastic and deterministic approximate inference algorithms have been developed both for general and specific graphical models. Stochastic algorithms, described in section 2.6.1, use averages over samples of configurations of the hidden nodes to approximate inference. In the large sample limit, many of these approximations become exact. Deterministic methods, described in section 2.6.2, approximate inference by parameterising the inferential outcome for each particular observation and optimising the values of the parameters according to some criterion of accuracy. For instance, one set of variational methods finds the member of a parameterised family of distributions over the unobserved variables that maximises a measure of goodness of probabilistic fit to the true posterior.

2.6.1 Stochastic Methods

Stochastic approximate inference use Monte Carlo methods to approximate expectations. Many inference outcomes can be written as expectations over the unobserved variables. For example, the denominator in equation (2.1), can be rewritten as an expectation over $P(\boldsymbol{d})$,

$$P(\boldsymbol{f}) = \langle P(\boldsymbol{f}|\boldsymbol{d})\rangle_{P(\boldsymbol{d})} . \tag{2.4}$$

The Monte Carlo approximation

$$\hat{P}_{\mathrm{lw}}(\boldsymbol{f}) = N^{-1}\sum_{n=1}^{N} P(\boldsymbol{f}|\boldsymbol{d}^{(n)}) \approx \langle P(\boldsymbol{f}|\boldsymbol{d})\rangle_{P(\boldsymbol{d})} , \tag{2.5}$$

where $\{\boldsymbol{d}^{(n)}\}$ are samples from $P(\boldsymbol{d})$, is an unbiased estimator of $P(\boldsymbol{f})$ that becomes exact in the limit of large N. Estimating $P(\boldsymbol{f})$ using equation (2.5) is called *likelihood weighting* [12, 61]. Posterior marginals can be estimated, using the same (or a different) set of samples, by computing

$$\hat{P}_{\mathrm{lw}}(D_k = s, \boldsymbol{f}) = N^{-1}\sum_{n=1}^{N} P(\boldsymbol{f}|\boldsymbol{d}^{(n)})\delta(s, d_k^{(n)}), \tag{2.6}$$

where $\delta(s, d_k^{(n)})$ is the Kronecker delta, and estimating $P(D_k = s|\boldsymbol{f})$ by

$$\hat{P}_{\mathrm{lw}}(D_k = s|\boldsymbol{f}) = \frac{\hat{P}_{\mathrm{lw}}(D_k = s, \boldsymbol{f})}{\hat{P}_{\mathrm{lw}}(\boldsymbol{f})}$$

If the likelihood is heavily peaked, $\hat{P}_{\mathrm{lw}}$ can have extremely high variance and thus high error. In this circumstance, it may be possible to reduce the estimator variance by sampling from a distribution, $Q(\boldsymbol{d})$, which is more similar to the likelihood than $P(\boldsymbol{d})$ is. For an arbitrary distribution over disease configurations $Q(\boldsymbol{d})$, we can write the denominator in equation (2.1) as an expectation over $Q(\boldsymbol{d})$

$$P(\boldsymbol{f}) = \sum_{\boldsymbol{d}} Q(\boldsymbol{d}) \frac{P(\boldsymbol{d})P(\boldsymbol{f}|\boldsymbol{d})}{Q(\boldsymbol{d})} = \left\langle \frac{P(\boldsymbol{d})P(\boldsymbol{f}|\boldsymbol{d})}{Q(\boldsymbol{d})} \right\rangle_{Q(\boldsymbol{d})}$$

which we can approximate by

$$\hat{P}_{\text{is}}(\boldsymbol{f}) = N^{-1} \sum_{n=1}^{N} \frac{P(\boldsymbol{d}^{(n)})P(\boldsymbol{f}|\boldsymbol{d}^{(n)})}{Q(\boldsymbol{d}^{(n)})} \approx \left\langle \frac{P(\boldsymbol{d})P(\boldsymbol{f}|\boldsymbol{d})}{Q(\boldsymbol{d})} \right\rangle_{Q(\boldsymbol{d})}$$

where $\{\boldsymbol{d}^{(n)}\}$ are samples from $Q(\boldsymbol{d})$. This technique is called *importance sampling* (see for example [54]) and $Q(\boldsymbol{d})$ is called the *proposal distribution.* The proposal distribution which gives the lowest variance estimator for $P(\boldsymbol{f})$ is $P(\boldsymbol{d}|\boldsymbol{f})$ [5], so choosing the best proposal distribution is as hard as the original inference problem.

Algorithms have been developed to select a good proposal distribution by bootstrapping. These algorithms use samples to adapt the proposal distribution either before or while doing approximate inference. Self-importance sampling (SIS) [61] and AIS-BN [5] are two examples of this approach. A modified version of the former (described in [63]) is a standard inference algorithms used on the QMR-DT, however AIS-BN has recently been shown by Cheng and Druzdzel [2000] to be significantly more accurate than SIS on the CPCS network, a medical belief network derived from the same knowledge base as the QMR-DT [47, 50].

2.6.2 Deterministic Methods

Deterministic approximate inference typically requires choosing a parameter structure (I use $\boldsymbol{\xi}$ for generic purposes), which is itself determined based on the graphical structure of the underlying probability distribution. Then, for a given observation, $\boldsymbol{f}^*$, the best parameters, $\boldsymbol{\xi}^*$ are determined by some computational procedure (which often is itself tied to the graphical model). Methods vary as to the parameter structure and how it is used in inference. Often $\boldsymbol{\xi}$ specifies a member of a family of distributions, $Q(\boldsymbol{d}; \boldsymbol{\xi})$, however $\boldsymbol{\xi}$ can also represent moments of the posterior distribution (e.g. see [68]) or specify likelihood functions $Q(\boldsymbol{f}^*|\boldsymbol{d}; \boldsymbol{\xi})$ (e.g. [22]). The parameters are usually determined by non-linear optimisation of a goodness of fit measure (e.g. [69] but see also loopy belief propagation [67]). Often, but not always (e.g. [22]), this goodness of fit measure is a KL-divergence involving the implied approximate posterior $Q(\boldsymbol{d}; \boldsymbol{\xi}^*)$ and the true posterior $P(\boldsymbol{d}|\boldsymbol{f}^*)$, These implied posteriors, Q, usually have a simpler form than P, and assume independencies that are not present in the true posterior. When KL-divergence cannot be optimised efficiently a tractably computable bound (e.g. [55]), or approximation (e.g. [11]) may be optimised instead.

Approximations made in deterministic methods often take advantage of properties specific to the probability distributions (e.g. [23]). For instance, in many distributions popular in probabilistic modelling, e.g. noisy-OR networks or sigmoid belief networks, the likelihood of an observation, $\boldsymbol{f}^*$, can be written as

$$P(\boldsymbol{f}^*|\boldsymbol{d}) = \prod_i g(\boldsymbol{\theta}_i^\top \boldsymbol{d}) \tag{2.7}$$

where $g(x)$ is a log concave function. Using Jensen's inequality and the variational transforms, both tractable lower and upper bounds on $P(\boldsymbol{f}^*|\boldsymbol{d})$ can be calculated [21]. These bounds have a product form with respect to the causes, allowing efficient computation of approximate expectations.

2.7 Recognition Models

Recognition models combine stochastic and deterministic approximations. Stochastic samples are used offline to tailor a recognition model to a particular probability distribution. This implies that these models are useful only if complete samples from the joint $P(\boldsymbol{d}, \boldsymbol{f})$ are easy to obtain. After this initial fitting, online inference is performed by evaluating a deterministic function of the observation.

Hinton et al [1995a] introduced a training algorithm to fit recognition models to a generative distribution, $P(\boldsymbol{d}, \boldsymbol{f})$, over two binary vectors $\boldsymbol{f}$ and $\boldsymbol{d}$. Their recognition model is a logistic regression network whose output layer contains a single unit z_k for each hidden variable D_k. For a particular binary observation vector $\boldsymbol{f}^*$, given recognition weights $\Omega = \{\{\boldsymbol{w}_k\}, \boldsymbol{b}\}$, the activity, z_k, of that output unit is given by

$$z_k = \sigma(\boldsymbol{w}_k^\top \boldsymbol{f}^* + b_k),$$

and is interpreted as an estimate of the posterior marginal probability that $D_k = 1$, i.e. $Q(D_k = 1; \boldsymbol{z}) = z_k$. The activity of the output layer as a whole implies a posterior distribution over disease configurations given by

$$Q(\boldsymbol{d}; \boldsymbol{z}) = \prod_k z_k^{d_k}(1 - z_k)^{(1-d_k)}$$

The estimates can be improved by setting the parameters, Ω, to minimise the total cross-entropy, $E(\Omega)$, on a set of samples $\{(\boldsymbol{d}^{(n)}, \boldsymbol{f}^{(n)})\}_{n=1}^N$ from $P(\boldsymbol{d}, \boldsymbol{f})$, where[3]

$$E(\Omega) = -\sum_n \log Q(\boldsymbol{d}^{(n)}; \boldsymbol{z}(\boldsymbol{f}^{(n)}, \Omega))$$

Hinton et al [1995a] use stochastic gradient descent to minimise $E(\Omega)$. When training a recognition model, unlike most other supervised learning problems, overfitting need never be a problem since new training data may be generated at will by sampling from $P(\boldsymbol{d}, \boldsymbol{f})$.

Hinton et al [1995a] and Dayan et al [1995]'s work used a recognition model as part of a biologically-motivated, self-supervised learning procedure that fits a stochastic generative model to data. Subsequent work on recognition models has also been in the context of learning generative models [42, 19, 20].

In this thesis, instead, I investigate using recognition models for inference when the generative model is prespecified. Whereas, previous attempts to improve inferential accuracy have concentrated on using recognition models that generate posteriors, Q, that model dependencies between variables [8, 7]. Here I improve accuracy both generating more complicated posteriors and by fitting more complicated recognition models, allowing more accurate prediction of the posterior marginals. The learning

[3] I have explicitly included the dependence on Ω here for clarity

tasks and models used in this thesis are orders of magnitude larger than those in previous work on recognition models [10].

2.8 Observation Processes

If the states of some of the observable variables may be unavailable (i.e. missing) doing inference, then one must consider the *observation process*[4] that determined which observable variables were available and which were not on any given observation. In some cases, knowledge about the observation process may be used to extract additional information about the unobserved variables, *above and beyond* any information contained in the states of the observed variables.

Rubin [1976] (see also [32]) specified the most general conditions under which an observation process is *ignorable* for a particular observation. If $\boldsymbol{R}$ is a vector of random indicator variables indicating the subset of the observables that are available (i.e. $R_i = 1$ indicates that the i-th observable is observed) then the observation process may be represented by a probability distribution over R that is conditioned on all the other variables. Rubin showed that for a particular observation, with indicator variables $\boldsymbol{r}$, the observation process is ignorable if and only if the probability assigned to $\boldsymbol{r}$ does not depend on the states of any of the unobserved variables. If this condition holds for all observations, then I will say that the observation process is, in general, ignorable.

If the observation process is non-ignorable then, in at least some cases, the fact that a variable is unobserved may be informative about its unobserved state (or that of another unobserved variable). Take for example, the medical diagnostic finding that a patient has severe abdominal pain. Common sense dictates that a conscious, sensate patient who does not complain of severe abdominal pain almost certainly does not have the pain. So if no information is recorded about whether or not the patient has the pain, we may assume then that the patient never complained about the pain

[4] Observation processes are more commonly referred to as *missing data mechanisms*, however the term observation process is more natural when, as in medical diagnosis, only a very small proportion of the observables are ever available

and therefore that the patient is likely not in pain. Note, however, that this deduction was based solely on the fact that no information was available, i.e. we know that if the patient did have the pain, then we would have observed that the patient had the pain.

Since the fact that a node is unobserved can be informative, we explicitly represent that fact. I introduce an additional set of variables, $\boldsymbol{E}$, called the *evidence variables.* Each variable E_i has the same state as the corresponding effect variable F_i when F_i is observed; when F_i is unobserved, $E_i = ?$. The conditional probability distribution of these evidence variables, $P(\boldsymbol{E}|\boldsymbol{F}, \boldsymbol{D})$ can depend upon the states of any of the variables in the probabilistic model. I call this distribution the observation process model.

Understanding the distinction between ignorable and non-ignorable observation processes is critical to understanding some of the contributions of this thesis. In particular, I show in section 5.2.3 that the medical diagnostic procedure is a non-ignorable observation process model. This non-ignorability means that disease posteriors calculated in the QMR-DT when observable but unobserved nodes are marginalised out are inaccurate. They are inaccurate because this marginalisation step implicitly assumes that the observation process model is ignorable. Overcoming this inaccuracy in the QMR-DT posteriors is one of the major themes of this thesis.

Chapter 3

The QMR-DT

3.1 Introduction

In this chapter, I describe the Quick Medical Reference, Decision Theoretic (QMR-DT), a Bayesian network designed for use in medical diagnosis. This network is widely used as a benchmark for comparing approximate inference algorithms and forms the basis of all graphical models used in this thesis.

There are a number of reasons for using the QMR-DT to evaluate the techniques introduced in this thesis. First, it is a very large graphical model, for which exact inference is often extremely intractable, so approximate inference is required. Second, it is a knowledge-rich graphical model designed to address a real-world problem. The QMR-DT is based on a knowledge base (the QMR/INTERNIST-1 KB [37, 36]) that incorporates more than 25 physician-years of effort. The builders of this knowledge base (KB) have made careful efforts to ensure its accuracy [13]. Accurate approximate inference algorithms for the QMR-DT would enable its use as a diagnostic support system [35]. Third, approximate inference in the QMR-DT is widely studied, (see for example [63, 22, 40, 43]). The main drawback is that all previous work has used either proprietary versions that are not widely available or randomly generated QMR-DT-like networks [11]. In this thesis, I use the anonymised QMR-DT which is derived[1] from the publicly available, anonymised QMR KB. The anonymised QMR KB is a noisy, label-free version of the proprietary QMR KB.[2] Since the KB is non-proprietary, my experiments can easily form the basis of a public evaluation paradigm. In summary, the QMR-DT presents a large-scale, real-world inference task for which practical approximate inference algorithms would be of great benefit.

This chapter describes both the QMR-DT and inference algorithms developed for the QMR-DT. Although the QMR-DT has been used as a prototypical inference problem, its unique structure is actually very conducive to particular reductions [15, 6] and approximations [17, 22, 11]. In particular, the QMR-DT is a member of a class of Bayesian networks, called binary noisy-OR two layer networks (BN2O networks). Structural properties of BN2O networks allow some types of inference to be done very efficiently. In addition, the QMR-DT itself has some properties that can be exploited to speed up inference. Specifically, the QMR-DT is sparsely connected and most of the probability mass under the QMR-DT is on sparse configurations of its variables. These structural and sparsity properties have been exploited by various exact and approximate inference algorithms specially designed for the QMR-DT. The most widely used of the exact inference algorithms is Quickscore [15]. This algorithm has played an important role in evaluating approximate inference algorithms because it can generate exact posterior marginals in reasonable time for some simple, though medically relevant, problems. These marginals have been used as a gold standard in the evaluations. However, many of the test problems used in this thesis lie beyond the upper limit of Quickscore's tractability. In an effort to preserve this gold standard, I introduce and evaluate an algorithm called structural Quickscore. Structural Quickscore exploits the sparsity connectivity of the QMR-DT, ignored by Quickscore, to speed up exact inference. This chapter also describes a number of approximate inference strategies that have been developed for the QMR-DT. Specific approximate inference algorithms used in this thesis will be described in later chapters.

[1] The derivation of the anonymised QMR-DT from the anonymised QMR KB is described in appendix A.

[2] The anonymised QMR KB is provided for research purposes by the University of Pittsburgh through the efforts of Frances Connell, Randolph A. Miller, and Gregory E. Cooper.

This chapter contains eight sections. Section 3.2 describes the structure and parameterisation of the QMR-DT. Inferential shortcuts licensed by this structure are detailed in section 3.3. The sparsity properties specific to the QMR-DT are described in section 3.4. Section 3.5 describes the Quickscore algorithm. Structural Quickscore is introduced in section 3.6 and evaluated in section 3.7. Section 3.8 describes some approximation strategies used with the QMR-DT. This chapter ends with a summary in section 3.9.

3.2 Structure and Parameterisation

The QMR-DT is a joint probability distribution over two binary random vectors: one representing the patient's unobservable disease state, $\boldsymbol{D}$, and another, $\boldsymbol{F}$, representing the observable manifestations of disease.[3] Each disease variable, D_k, encodes the presence or absence of one of the 570 represented diseases. These diseases cover the majority of the important diseases in internal medicine [36]. The 4075 manifestation variables represent symptoms, medical history, demographic data, physical signs, or laboratory test results [36] and can be positive or negative.[4] A positive manifestation typically represents an abnormal state of the represented quantity. Non-binary manifestations are encoded using multiple binary manifestation nodes. Under this encoding, the state of the manifestation is represented by activating only one of the multiple units. Figure 3-1 shows examples of the encoding of continuous-valued and non-binary discrete-valued manifestations.

The distribution, $P(\boldsymbol{d}, \boldsymbol{f})$, defined by the QMR-DT can be factored as

$$P(\boldsymbol{d}, \boldsymbol{f}) = \left[\prod_k P(d_k)\right] \prod_i P(f_i|\boldsymbol{d}), \tag{3.1}$$

[3] Though these variables are usually called findings, i.e. observations made by a physician or a patient, I use the term manifestations to emphasize the fact that the QMR-DT is only a model of the occurence and manifestation of disease. In section 5.3, I describe a model of the procedure by which findings are made.

[4] Strictly speaking, medical history and demographic data are not manifestations of disease. Treating these as such is an approximation made by the original designers of the QMR-DT to simplify inference.

A

Encoded manifestation: **Weakness of facial muscles**

Encoding:

□ Bilateral

□ Unilateral inc forehead

□ Unilateral, lower two-thirds only

Examples:

1 0 0	*Bilateral facial muscle weakness*
0 0 1	*Lower two-third left-side weakness*

B

Encoded manifestation: **Birth weight**

Encoding:

□ <1500 gm

□ 1501-2500 gm

□ 2501-4000 gm

□ >4000 gm

Examples:

0 0 0 1	*Birth weight: 4230 gm*
0 1 0 0	*Birth weight: 2100 gm*

Figure 3-1: Encoding of non-binary manifestations in the QMR-DT. A) Example encoding of a non-binary valued categorical manifestation. B) Example encoding of a continuous valued manifestation.

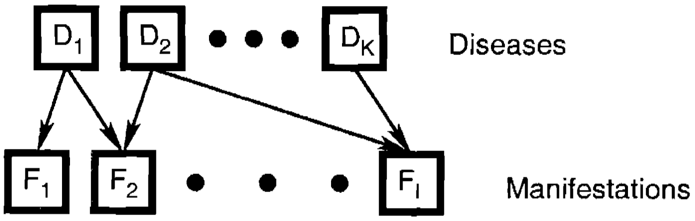


Figure 3-2: Structure of the QMR-DT.

and is defined by three sets of parameters: a vector of Bernoulli parameters, $\boldsymbol{p}$, one for each disease node, that specify the factorial disease prior, the vector $\boldsymbol{\theta}_0$ (with $\theta_{0i} \geq 0$) and matrix $\Theta = \{\theta_{ki}\}$ (also with $\theta_{ki} \geq 0$) which together specify the conditional probabilities $P(f_i|\boldsymbol{d})$. The conditional has a noisy-OR parameterisation, i.e.

$$P(F_i = -|\boldsymbol{d}) = e^{-\theta_{0i} - \sum_k \theta_{ki} d_k}. \tag{3.2}$$

Note that the conditional distribution of F_i depends only upon those diseases k for which $\theta_{ki} > 0$. These diseases are the parents of F_i in the directed graphical model of the QMR-DT shown in figure 3-2. I use Π_i to denote the set of indices of the parents of F_i and Λ_k to denote the set of indices of the children of D_k. When F_i has a single parent, I use π_i to denote its unique index. The $\boldsymbol{\theta}_0$ parameters are called *leak terms*. A *leak event* occurs when a manifestation is positive but none of its parents are active.

Bayesian networks having the same structure and parameterisation as the QMR-DT are called binary two-layer noisy-OR (BN2O) networks. The next section describes inference short-cuts that can be used in BN2O networks.

3.3 BN2O Network Inference Simplifications

This section describes techniques that can be used to simplify exact inference in any binary noisy-OR two-layer (BN2O) network. These techniques, licensed by properties of BN2O networks, allow many types of evidence to be efficiently incorporated into the disease prior. This allows the inference problem in the original BN2O network to be reduced into an inference problem in a similar BN2O network that nonetheless contains fewer manifestation nodes and less evidence. Specifically, the reduced BN2O network contains all of the disease nodes of the original network but only the multiparent positive manifestations. The conditional distributions over the remaining manifestations are the same as in the original distribution but the disease priors are updated to incorporate evidence from negative and single parent positive mani-

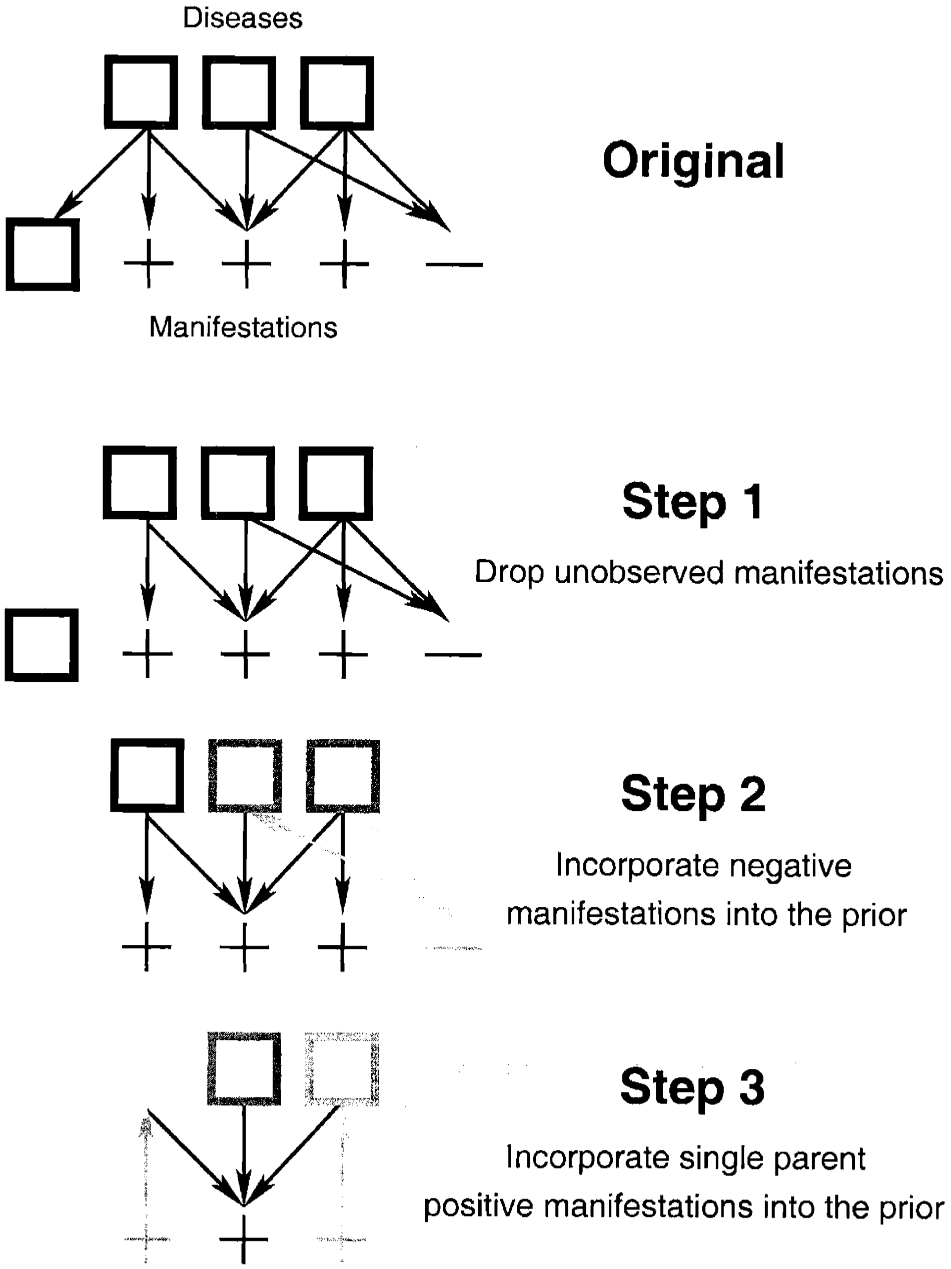

Diseases
Manifestations
Original
Step 1
Drop unobserved manifestations
Step 2
Incorporate negative manifestations into the prior
Step 3
Incorporate single parent positive manifestations into the prior

Figure 3-3: Graph transformations in QMR-DT inference. I use different shades of outlines to indicate categorical differences between the disease nodes. Disease nodes with differently shaded outlines have priors that incorporate different pieces of evidence.

festations. Disease posteriors calculated using the reduced network are the same as those in original network. However, the number of manifestation nodes can be much smaller, simplifying inference considerably.

I present the reduction in three steps. The output of each of the three steps is a new probability distribution whose disease posteriors are the same as the original distribution. In the first step, the original network is replaced with one that has the unobserved manifestation nodes removed. In the second step, the effect of the negative manifestations is incorporated into the prior and the negative nodes are pruned. In the final step, the effect of the single parent positive manifestations is incorporated into the prior and those nodes are pruned from the graph. Figure 3-3 shows these three steps graphically. The final graph contains only multiparent positive manifestations.

I represent an inference problem by the sets $\mathcal{I}^+$ and $\mathcal{I}^-$. These sets contain the indices of the observed positive and negative manifestations, respectively. The posterior probability of the disease vector $\boldsymbol{s}$ given $\mathcal{I}^+$ and $\mathcal{I}^-$ is

$$P(\boldsymbol{D} = \boldsymbol{s}|F^+, F^-) = \frac{P(\boldsymbol{D} = \boldsymbol{s}, F^+, F^-)}{\sum_{\boldsymbol{d}'} P(\boldsymbol{D} = \boldsymbol{d}', F^+, F^-)}, \tag{3.3}$$

where F^+ and F^- denote the events $\boldsymbol{F}_{\mathcal{I}^+} = +$ and $\boldsymbol{F}_{\mathcal{I}^-} = -$ respectively. Note that both the numerator and denominator in equation (3.3) are comprised of marginals of the form $P(\boldsymbol{d}, F^+, F^-)$. In each of the steps, I show how these marginals can be replaced with joint probabilities under the new distribution. Though these joints will be over a subset of the evidence, the disease posteriors under the new distribution will nonetheless be the same.

Step 1: Unobserved Manifestations

Computing $P(\boldsymbol{d}, F^+, F^-)$ requires marginalising over the unobserved manifestations. Here, I show, however, that the distribution P can be replaced with another distribution P' that contains all the disease nodes but only the observed manifestation nodes. This new distribution P' has the same disease posteriors as P given the observed evidence, i.e.

$$\forall \boldsymbol{d}, P'(\boldsymbol{d}|F^+, F^-) = P(\boldsymbol{d}|F^+, F^-). \tag{3.4}$$

Note, first of all, that due to the conditional independence of findings,

$$P(\boldsymbol{d}, F^+, F^-) = \sum_{\boldsymbol{f}|\boldsymbol{f}_{\mathcal{I}^+}=+, \boldsymbol{f}_{\mathcal{I}^-}=-} P(\boldsymbol{d}) \prod_i P(f_i|\boldsymbol{d}), \tag{3.5}$$

now, commuting the sum in equation (3.5) with the product gives:

$$P(\boldsymbol{d}, F^+, F^-) = P(\boldsymbol{d})P(F^+, F^-|\boldsymbol{d}) \prod_{i \in \mathcal{I} \setminus (\mathcal{I}^+ \cup \mathcal{I}^-)} P(F_i = +|\boldsymbol{d}) + P(F_i = -|\boldsymbol{d}). \tag{3.6}$$

Because each term in the product in equation (3.6) is one, the product as a whole is one and the marginalisation has no effect. If we define P' such that $P'(\boldsymbol{D}) \equiv P(\boldsymbol{D})$ and $P'(F_i|\boldsymbol{D}) \equiv P(F_i|\boldsymbol{D}), \forall i \in \mathcal{I}^+ \cup \mathcal{I}^-$, then for all disease vectors, $\boldsymbol{d}$,

$$P'(\boldsymbol{d}, F^+, F^-) = P(\boldsymbol{d}, F^+, F^-). \tag{3.7}$$

By substituting P' for P in equation (3.3), we have shown equation (3.4) is correct.

Step 2: Negative Manifestations

In this step, a new distribution is produced that has negative manifestations from the old distribution eliminated and incorporated into the new disease prior. Here, I show that the distribution that results from this step has the same posteriors as the original distribution. Note that, due to the bipartite structure of the QMR-DT and the conditional independence of manifestations given diseases, the joint distribution $P'(\boldsymbol{d}, F^+, F^-)$ decomposes as:

$$P'(\boldsymbol{d}, F^+, F^-) = P'(F^+|\boldsymbol{d})P'(\boldsymbol{d}|F^-)P'(F^-) \tag{3.8}$$

This decomposition is important since here I show that

$$P'(\boldsymbol{d}|F^-) = \prod_k P^\dagger(d_k) \tag{3.9}$$

for some factorial distribution $P^\dagger(\boldsymbol{d})$ over the diseases. To satisfy ourselves that equation (3.9) is correct, we note that because $P'(\boldsymbol{d}|F^-) \propto P'(\boldsymbol{d}, F^-)$ and

$$P'(\boldsymbol{d}, F^-) = \left(\prod_k P'(d_k)\right) \prod_{i \in \mathcal{I}^-} e^{-\theta_{i0} - \sum_k \theta_{ik} d_k},$$

that equation (3.9) is satisfied by

$$P^\dagger(d_k) \propto P'(d_k) e^{-d_k \sum_i \theta_{ik}}. \tag{3.10}$$

Defining $P^\dagger(F^+|\boldsymbol{D}) \equiv P'(F^+|\boldsymbol{D})$, and using equations (3.8) and (3.7), we can write

$$P(\boldsymbol{d}, F^+, F^-) = P^\dagger(F^+|\boldsymbol{d}) P^\dagger(\boldsymbol{d}) P'(F^-). \tag{3.11}$$

By substituting equation (3.11) into equation (3.3) and cancelling the $P'(F^-)$ term from the numerator and the denominator, we have shown that

$$P(\boldsymbol{d}|F^+, F^-) = P^\dagger(\boldsymbol{d}|F^+)$$

as required.

Step 3: Single Parent Positive Manifestations

In this step, the single parent positive manifestations from $P^\dagger$ are incorporated into a new prior and a new distribution containing only the multiparent positive manifestations is formed. This new distribution also has the same disease posteriors as the initial distribution.

Let $\mathcal{S}$ be the set of indices of the single parent manifestations, then $\mathcal{I}^+ \cap \mathcal{S}$ is the set of indices of the single parent positive manifestations and $\mathcal{I}^+ \setminus \mathcal{S}$ contains the indices of the multiparent positive manifestations. Again, due to the bipartite structure of the QMR-DT and the conditional independence of manifestations given diseases, the joint distribution $P^\dagger(\boldsymbol{d}, F^+)$ decomposes as:

$$P^\dagger(\boldsymbol{d}, F^+) = P^\dagger(F^+_{\text{mp}}|\boldsymbol{d})P^\dagger(\boldsymbol{d}|F^+_{\text{sp}})P^\dagger(F^+_{\text{sp}}) \tag{3.12}$$

where F^+_{sp} and F^+_{mp} denote the events $\boldsymbol{F}_{\mathcal{I}^+\cap\mathcal{S}} = +$ and $\boldsymbol{F}_{\mathcal{I}^+\setminus\mathcal{S}} = +$ respectively.

Here I show how to compute a factorial distribution $P^\ddagger$ over the diseases so that

$$P^\dagger(\boldsymbol{d}|F^+_{\text{sp}}) = \prod_k P^\ddagger(d_k). \tag{3.13}$$

In doing this, I follow a similar argument as the last step. Namely, because

$$P^\dagger(\boldsymbol{d}, F^+_{\text{sp}}) = \left(\prod_k P^\dagger(d_k)\right) \prod_{i\in\mathcal{I}^+\cap\mathcal{S}} P^\dagger(F_i = +|d_{\pi_i}),$$

equation (3.13) is satisfied by

$$P^\ddagger(d_k) \propto P^\dagger(d_k) \prod_{i\in\mathcal{I}^+\cap\mathcal{S}\cap\Lambda_k} \left(1 - e^{-\theta_{i0}-\theta_{i\pi_i}d_{\pi_i}}\right). \tag{3.14}$$

Now defining $P^\ddagger(F^+_{\text{mp}}|\boldsymbol{D}) \equiv P^\ddagger(F^+_{\text{mp}}|\boldsymbol{D})$, it can easily be shown that

$$P(\boldsymbol{d}, F^+, F^-) = P^\ddagger(F^+_{\text{mp}}|\boldsymbol{d})P^\ddagger(\boldsymbol{d})P^\dagger(F^+_{\text{sp}})P(F^-). \tag{3.15}$$

By substituting equation (3.15) into equation (3.3) we can show that

$$P(\boldsymbol{d}|F^+, F^-) = P^\ddagger(\boldsymbol{d}|F^+_{\text{mp}})$$

as required.

The model resulting from the third step, $P^\ddagger$, has a graph that is a subgraph of the original model P. This subgraph contains only the manifestation nodes that are both multiparent and positive in the original inference problem $(\mathcal{I}^+, \mathcal{I}^-)$. The manifestation conditional probabilities are the same for those nodes shared by the two models, however the priors are different.

Easy Computation of the Updated Prior

Here I show how to easily compute the prior resulting from applying the three reduction steps. Specifically, I show how to compute the new log prior odds, $l_k^\ddagger = \log\{p_k^\ddagger/(1 - p_k^\ddagger)\}$, (where $p_k^\ddagger = P^\ddagger(D_k = 1)$). The prior distribution can be computed from the log odds using the logistic function, i.e. $p_k^\ddagger = \sigma(l_k^\ddagger)$ where $\sigma(x) = (1 + \exp(-x))^{-1}$. These log odds have a very simple form, namely

$$l_k^\ddagger = l_k - \left[\sum_{i \in \mathcal{I}^-} \theta_{ik}\right] + \sum_{j \in \mathcal{I}^+ \cap \boldsymbol{S} \cap \Lambda_k} c_j, \tag{3.16}$$

where $l_k = \log\{p_k/(1 - p_k)\}$ and

$$c_j = \log \frac{1 - e^{-\theta_{j0} - \theta_{j\pi_j}}}{1 - e^{-\theta_{j0}}}.$$

Summary

In this section, I have shown how to simplify the computation of posteriors in BN2O networks by reducing the original network to a new network whose graph is a subgraph of the original with less evidence and no unobserved manifestations. The disease priors in the new graph are the posteriors in the original graph given the negative and single parent positive evidence. The disease posteriors in the new graph are identical to those in the old graph. Because this reduction exists, inference algorithms for the QMR-DT need only compute the effect of multiparent positive findings on the disease posteriors. In the following, unless otherwise stated, I will assume that this reduction has already been done and that inference problems only contain multiparent positive manifestations.

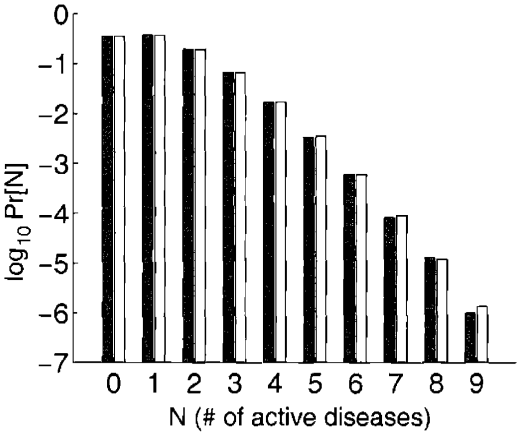


Figure 3-4: Empirical distribution of number of active diseases. The bar graph shows two distributions. The black bars are the empirical distribution generated using 10^6 samples from $P(\boldsymbol{d})$. No sample contained more than nine active diseases. The grey bars show, for comparison, a Poisson distribution with the same mean.

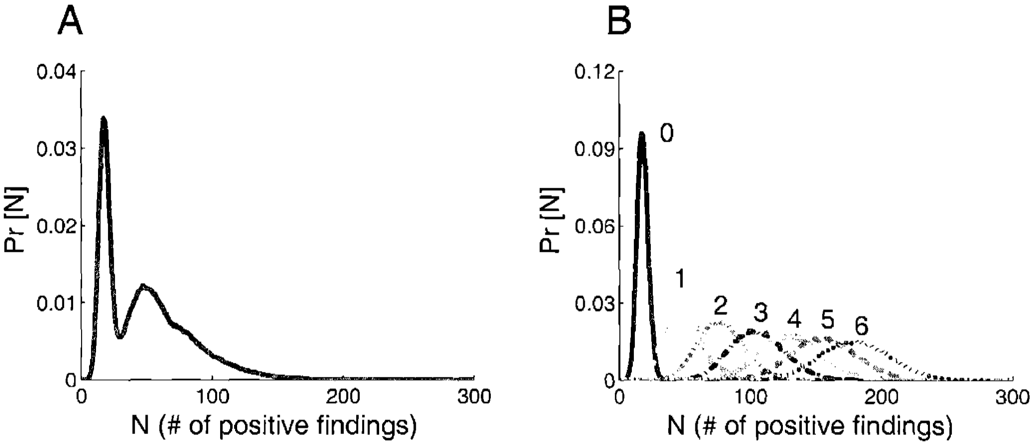


Figure 3-5: Empirical distributions of number of positive manifestations. A) Number of positive manifestations. Empirical distribution was generated using 10^6 samples. B) Number of positive manifestations conditioned on number of active diseases. Each curve is marked with the number of active diseases conditioned on. Each distribution was generated using 10^5 samples.

3.4 Sparsity in the QMR-DT

The parameter settings used in the QMR-DT produce sparsities which have relevance both to the design and evaluation of inference algorithms for the QMR-DT. Specifically, the QMR-DT puts most of its probability mass on sparse disease and manifestation vectors and the graphical model of the QMR-DT is sparsely connected. These sparsities have been exploited by various QMR-DT specific inference algorithms and bear on the evaluation of inference algorithms on the QMR-DT. In this section, I introduce each of the three sparsities to provide a reference for later discussion.

Sparsity of Disease Vectors

Most of the probability mass under the disease prior, $P(\boldsymbol{d})$, is concentrated on a small proportion of the possible configurations. Specifically, the entropy of $P(\boldsymbol{d})$ is 9.8 which is the same as that of a uniform distribution over 907 different choices instead of the 2^{570} possible combinations. Most of the mass in this highly peaked distribution is on sparse disease vectors, as figure 3-4 shows. Disease vectors with one or zero active diseases account for 72% of the prior mass. Due to this concentration of prior mass, diagnoses with large numbers of active diseases require significant evidential support. Note also that this concentration suggests approximate inference algorithms based on enumerating disease vectors in order of the number of active diseases.

It should be noted here, however, that the disease priors that I generated for the anonymised QMR-DT are not based on real medical data. The anonymised QMR KB does not presently contain disease priors and the lack of disease labels makes it difficult to assign priors using medical data. However, the priors that I generated have approximately the same expected number of diseases and a similar distribution of Bernoulli parameters as those associated with the QMR-DT described in [22]. Please see appendix A for further details.

Sparsity of Positive Manifestations

In samples from the QMR-DT, only a very small proportion of the manifestation nodes are positive. The expected number of positive manifestations is 49 out of a possible 4075. Figure 3-5 shows the distribution of numbers of positive manifestations. This sparsity of positive findings is exploited by the Quickscore algorithm, described in section 3-6. Also note that the lack of overlap between some of the conditional distributions in figure 3-5B allows the number of positive manifestations to be used as a rough indicator of the number of active diseases when all the manifestation nodes are observed and a rough lower bound on this number when only some of the nodes are observed.

Sparse Connectivity

The QMR-DT is sparsely connected. Specifically only 2% of the possible disease-finding links are present. This sparse connectivity may allow the QMR-DT to become easily disconnected when manifestations nodes are eliminated from its graphical model. Section 3.6 describes an algorithm that exploits this property to speed up exact inference in the QMR-DT. Section 3.7 gives experimental results showing the susceptibility of the QMR-DT to rapid disassembly.

3.5 Quickscore

3.5.1 Introduction

Quickscore is the standard exact inference algorithm for the QMR-DT. In its most basic form, it calculates the marginal probability of a set of positive manifestations, i.e.

$$P(F^+) = \sum_{\boldsymbol{d}} P(F^+, \boldsymbol{d}), \tag{3.17}$$

in BN2O networks. Because the posterior marginal probability of the presence of disease k, $P(D_k = 1|F^+)$, can written as a ratio of two of the marginal probabilities

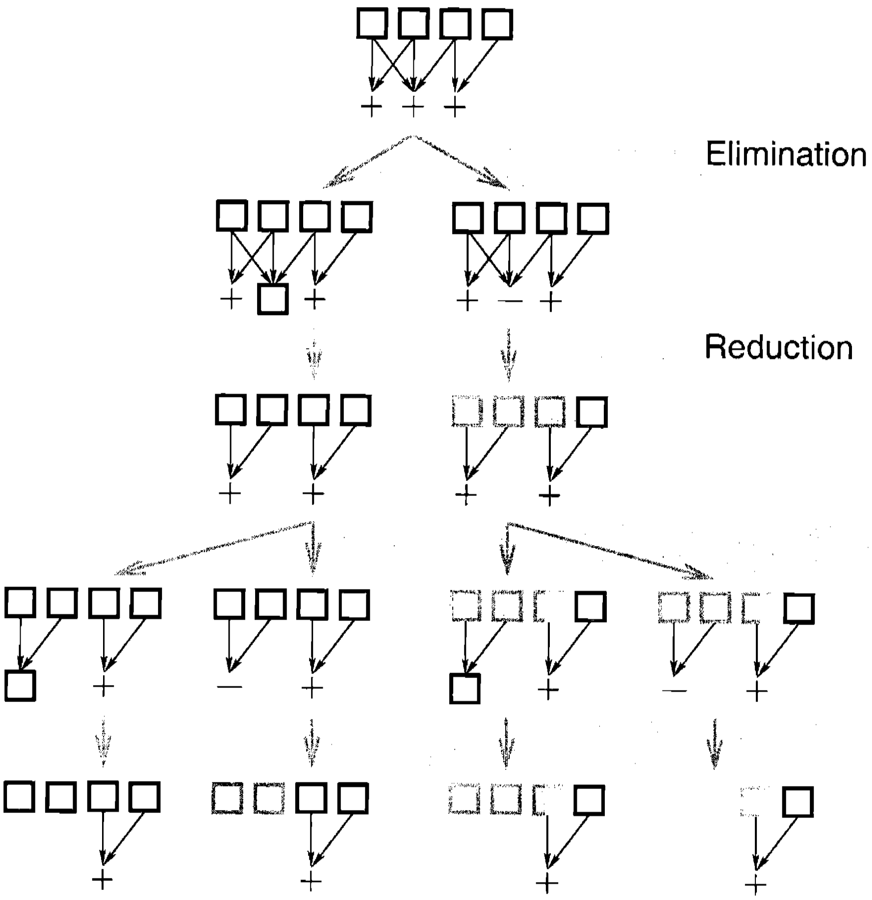


Figure 3-6: Recursion tree of a call to Quickscore. Only the elimination and reduction steps are shown. The shaded disease node borders have the same interpretation as in figure 3-3. Note that the positive manifestations are eliminated in the same order along each branch of the tree and that the graph becomes disconnected after the first elimination. Section 3.6 describes an algorithm that exploits disconnectedness to speed-up Quickscore.

of the form of equation 3.17[5], Quickscore may used to calculate posterior marginals for all of disease nodes. The efficiency of Quickscore comes from transforming the combinatorial sum over disease vectors in equation (3.17) into a combinatorial sum over the positive manifestations. In the QMR-DT, due to the sparsity of positive manifestations, the latter sum is tractable for some inference problems whereas the former sum, over 2^{570} disease configurations, is never tractable. Though Quickscore was introduced as a serial algorithm [15], here I present a recursive formulation that will facilitate later discussion.

3.5.2 Recursive Quickscore

Introduction

The recursive Quickscore algorithm consists of a single recursive function that returns the marginal probability of the evidence. Each call to the function may be divided into three stages: *elimination*, *reduction*, and *recursion*. In the elimination stage, a single positive manifestation is eliminated. This elimination replaces the original marginal probability with a difference of marginals between one where the eliminated positive manifestation is unobserved and the other where the manifestation is negative. In the reduction step, the two new marginals are reduced to weighted marginal probabilities calculated on BN2O networks with the eliminated node removed. In the recursion stage, Quickscore is called recursively on each of the two new BN2O networks. The recursion bottoms out when only a single positive manifestation remains and its marginal probability is easily computed. Since each recursive step eliminates one positive manifestation and doubles the number of marginal probabilities to be calculated, the time complexity of Quickscore is exponential in the number of positive manifestations. Figure 3-6 shows a graphical depiction of the Quickscore algorithm.

In the following, I describe each of the three steps in further detail. Note that I am assuming that the reduction described in section 3.3 has already been performed and, therefore, the inference problem consists only of multiparent positive manifestations.

[5] Note the numerator in this ratio has the prior probability, $P(D_k = 1)$, set to one.

I represent these manifestations with the set $\mathcal{I}^+$.

Elimination The elimination step in Quickscore uses the properties of BN2O detailed in section 3.3 to rewrite the marginal probability as a weighted difference of probabilities. Specifically, since the manifestation nodes are binary, for $i \in \mathcal{I}^+$, we can write

$$P(F_{\mathcal{I}^+\setminus\{i\}} = +) = P(F_i = +, F_{\mathcal{I}^+\setminus\{i\}} = +) + P(F_i = -, F_{\mathcal{I}^+\setminus\{i\}} = +) \quad (3.18)$$

which can be rearranged to give

$$P(F^+) = P(F_{\mathcal{I}^+\setminus\{i\}} = +) - P(F_{\mathcal{I}^+\setminus\{i\}} = +, F_i = -). \quad (3.19)$$

The RHS of equation (3.19) is a difference of two marginal probabilities each calculated on the same BN2O network but with different sets of evidence. The first set has the originally positive F_i unobserved, and the second set has F_i negative. In figure 3-6, these two new sets of evidence are represented as two similar graphical models that differ only in their node assignments.

Reduction Using the reductions described in section 3.3 both of the RHS terms in equation (3.19) can be reduced into marginal probabilities calculated in new graphs that don't contain the F_i node. In the first graph, F_i is simply dropped because it is unobserved. In the second graph, the negative evidence, $F_i = -$, is incorporated into the disease priors producing a new distribution $P^\dagger$. These transformations allow equation (3.19) to be written:

$$P(F^+) = P(F_{\mathcal{I}^+\setminus\{i\}} = +) - CP^\dagger(F_{\mathcal{I}^+\setminus\{i\}} = +), \quad (3.20)$$

where the weight $C = P(F_i = -)$ may be efficiently computed.

Recursion After the reduction, Quickscore is called recursively to compute each of the two new marginal probabilities. The recursion bottoms out on marginal probabilities of a single positive manifestation. The calculation of these probabilities is trivial since $P(F_i = +) = 1 - P(F_i = -)$.

Summary

In this section, I have presented a recursive formulation of the Quickscore algorithm. This formulation, like the serial one, has exponential time complexity in the number of positive manifestations. This time complexity limits the practical uses of the algorithm. For example, Jaakkola and Jordan report an average running time of 26.9 seconds [22] when using Quickscore on the 4 (of 48) CPC[6] cases with at most 20 positive manifestations. However, the median number of positive manifestations in the CPC cases is 36, so most of the CPC problems remain intractable. However, by exploiting the sparse connectivity of the QMR-DT, exact inference can be made more efficient. In the following section, I describe an extension to the Quickscore algorithm that may reduce its time complexity for sparsely connected BN2O networks.

3.6 Structural Quickscore

3.6.1 Introduction

This section investigates an extension of the Quickscore algorithm that exploits sparse connectivity in BN2O networks to simplify some of the computations of the marginal probabilities. The extension is based upon the SPI algorithm [6] as applied to the QMR-DT. This algorithm contains a step similar to the positive manifestation elimination. In reference to this step, D'Ambrosio [6] observed that whenever eliminating a positive manifestation causes a BN2O network to become disconnected, inference computations can be done independently in each of the resulting connected components. By exploiting this independence, D'Ambrosio was able to demonstrate substantial speed-ups on some CPC cases. Quickscore can be extended to exploit this

[6] The CPC cases are sets of pedagogical diagnostic problems encoded for use with the QMR-DT. They are further described in section 5.2.3

independence and achieve similar speed-ups.

This section describes an algorithm called *structural Quickscore* based on Quickscore. I begin the description by showing, in section 3.6.2, that marginal probabilities of manifestations in BN2O networks can be factored into the product of the marginals in each of the connected components of the model. This factoring is exploited by structural Quickscore, which is described in section 3.6.3. Section 3.7 evaluates this algorithm.

3.6.2 Factoring in Disconnected BN2O networks

If a BN2O network is disconnected, then the marginal probability of a set of positive manifestations in the whole network is equal to the product of a set of marginals over each of the individual components. In particular, if $\mathcal{I}^+$ is the set of indices of the positive manifestations, and $\{\mathcal{I}_n^+\}_{n=1}^N$ is a partition of $\mathcal{I}^+$ into the sets of indices of the positive manifestations in each of the N connected components of the graph, then

$$P(F_{\mathcal{I}^+} = +) = \prod_n P(F_{\mathcal{I}_n^+} = +). \tag{3.21}$$

In the following, I show that equation (3.21) holds for two connected components, i.e. $N = 2$. This argument can be easily extended by induction to show that equation (3.21) holds for arbitrary N.

Let $\mathcal{K}_1$ and $\mathcal{K}_2$ be the sets of the indices of the disease nodes in the two connected components. Then, since manifestation nodes only have parents from within the same connected component,

$$\begin{aligned} P(F_{\mathcal{I}_1^+}|\boldsymbol{D}) &= P(F_{\mathcal{I}_1^+}|\boldsymbol{D}_{\mathcal{K}_1}), & (3.22)\\ P(F_{\mathcal{I}_2^+}|\boldsymbol{D}) &= P(F_{\mathcal{I}_2^+}|\boldsymbol{D}_{\mathcal{K}_2}). & (3.23)\end{aligned}$$

Equations (3.22) and (3.23) can be used, together with the marginal independence of diseases and the conditional independence of manifestations, to write

$$P(F_{\mathcal{I}^+} = +) = \sum_{\boldsymbol{d}_{\mathcal{K}_1}} \left[P(\boldsymbol{d}_{\mathcal{K}_1}) P(F_{\mathcal{I}_1^+} | \boldsymbol{d}_{\mathcal{K}_1}) \left(\sum_{\boldsymbol{d}_{\mathcal{K}_2}} P(\boldsymbol{d}_{\mathcal{K}_2}) P(F_{\mathcal{I}_2^+} | \boldsymbol{d}_{\mathcal{K}_2}) \right) \right] . \qquad (3.24)$$

Since no term in the inner summation in equation (3.24) depends upon $\boldsymbol{d}_{\mathcal{K}_1}$, equation (3.24) may be written

$$P(F_{\mathcal{I}^+} = +) = \prod_{n=1}^{2} \sum_{\boldsymbol{d}_{\mathcal{K}_n}} P(\boldsymbol{d}_{\mathcal{K}_n}) P(F_{\mathcal{I}_n^+} = + | \boldsymbol{d}_{\mathcal{K}_n}). \qquad (3.25)$$

Using the substitution $P(F_{\mathcal{I}_n^+}) = \sum_{\boldsymbol{d}_{\mathcal{K}_n}} P(\boldsymbol{d}_{\mathcal{K}_n}) P(F_{\mathcal{I}_n^+} | \boldsymbol{d}_{\mathcal{K}_n})$ in equation (3.25) gives equation (3.21) as required.

Note, in collorary, that if a connected component n doesn't have any manifestations associated with it, i.e. $\mathcal{I}_n^+ = \emptyset$, then the component consists of a single disease node and $P(F_{\mathcal{I}_n^+} = +) = 1$.

Equation (3.21) may be exploited to speed up Quickscore on disconnected BN2O networks. Since each of the connected components is itself a BN2O network, Quickscore may be distributed over the components. The marginal probability of all the positive manifestations is the product of the output of Quickscore on each of the individual components. This distributed version of Quickscore can have significant savings over vanilla Quickscore since it is exponential only in the number of positive manifestations of the largest partition of $\mathcal{I}^+$. The following section describes structural Quickscore, an algorithm that uses distributed Quickscore as a subroutine. Structural Quickscore can be used to speed up the computation of the marginal probability in a BN2O network that, at least initially, consists of a single connected component.

3.6.3 Description of the Algorithm

Introduction

If a BN2O network is connected but not all possible disease-manifestation links are present, then, through judicious choice of the elimination ordering, it may be possible to disconnect the network before all of the positive manifestations have been eliminated. Disconnecting the network, as section 3.6.2 showed, allows the Quickscore computation to be distributed over the connected components leading to possibly substantial savings in computation time.

This section describes a two-stage algorithm that exploits sparse connectivity in BN2O networks to speed up Quickscore. In the first stage, an elimination ordering of the positive manifestations is determined. The elimination ordering is selected using a heuristic algorithm that aims to minimise the number of computations in the second stage. In the second stage, Quickscore is recursively applied to the network and the computation of marginals is distributed across disconnected components whenever possible.

Stage 1: Choosing the elimination ordering

Selecting a good elimination ordering is crucial, since some orderings will disassemble the graph much more quickly than others. Here, I describe three different heuristics to find a good elimination ordering. The three heuristics are compared in section 3.7.

The first heuristic, *most parents*, eliminates the positive manifestations in descending order of number of parents. This ordering removes the largest number of links at each step, possibly leading to easier disconnection of the graph. This heuristic was first described in [6].

The second heuristic, *greedy A*, eliminates the manifestation that best decreases an estimate of the remaining amount of computation. This estimate is the time complexity of distributed Quickscore on the connected components remaining after the node is eliminated. Specifically, this estimate, for manifestation node i, is

$$\sum_n |\mathcal{K}_n| 2^{|\mathcal{I}_n^+|}$$

where $\{\mathcal{I}_n^+\}$ and $\{\mathcal{K}_n\}$ are, respectively, the partitions of the indices of the remaining manifestation nodes and disease nodes in the various connected components. Ties are resolved by choosing the manifestation node with the largest number of parents.

```
$SQuickscore(G, i, \boldsymbol{p})$
$m_1 \leftarrow 1$;
$m_2 \leftarrow 1$;
$\{\boldsymbol{p}^\dagger, C\} \leftarrow$ add-neg-evidence$(G, i, \boldsymbol{p})$;
if $|G.\boldsymbol{f}| > 1$
  $\{\mathbf{CC}, \mathbf{next}\} \leftarrow$ reference-elim-order$(i)$;
  for $j \leftarrow 1$ to $|\mathbf{next}|$
      $m_1 \leftarrow m_1 \times SQuickscore(\mathbf{CC}[j], \mathbf{next}[j], \boldsymbol{p})$;
      $m_2 \leftarrow m_2 \times SQuickscore(\mathbf{CC}[j], \mathbf{next}[j], \boldsymbol{p}^\dagger)$;
  end for
end if
return $m_1 - C \times m_2$;
```

Figure 3-7: Pseudocode for the second stage of structural Quickscore

The third heuristic, *greedy B*, combines aspects of both *most parents* and *greedy A*. Under this heuristic, each manifestation node i has score $\sum_n 2^{|\mathcal{I}_n^+|}$ where $\{\mathcal{I}_n^+\}$ has the same definition as above. Ties are resolved, as in *greedy A*, by choosing the node with the most parents. Because the score doesn't depend on the number of diseases in the connected components, *greedy B* acts like *most parents* when the graph can't be disconnected by eliminating a single positive manifestation. When the graph can be disconnected, *greedy B* will often act like *greedy A* because, due to the exponentiation, both scoring metrics tend to favour eliminations that disconnect the graph.

While the elimination ordering is being generated, a data structure, $\boldsymbol{A}$, is constructed that is used to guide the recursion in the second stage. This structure associates each manifestation node with two lists. The first is a list of the connected components remaining after the node is eliminated. The second is a list of the next manifestation node to eliminate in each of the connected components.

Stage 2: Computing the marginal probability

In the second stage, the marginal probability is calculated by a recursive algorithm that follows the elimination ordering selected in the first stage. In each function call, the algorithm eliminates one positive manifestation and distributes the computation of the two resulting marginals, m_1 and m_2, over the connected components if possible. Pseudocode for the recursive function used by structural Quickscore is shown in figure 3-7. Here I describe this pseudocode in further detail.

The function, *SQuickscore,* has three input arguments:

1. G – a graph structure that describes the current connected component,

2. i – the index of the manifestation node to eliminate from G, and

3. $\boldsymbol{p}$ – the vector of the Bernoulli parameters for the factorial prior.

The structure G contains the field $G.\boldsymbol{f}$, a vector of the indices of the manifestation nodes in G. Note that $\boldsymbol{p}$ contains an entry for each disease node, even if the disease node isn't in G.

There are two subroutines called by *SQuickscore.* The first subroutine, *add-neg-evidence*, returns two values: $\boldsymbol{p}^\dagger$ and C. The new prior parameter vector, $\boldsymbol{p}^\dagger$, incorporates the negative manifestation, F_i, and the weight, $C = P(F_i = -)$, is the marginal probability of the evidence $F_i = -$ under the BN2O network specified by G and $\boldsymbol{p}$. The second subroutine, *reference-elim-order*, references A and returns two values: **CC** and **next**. The value, **CC**, is a list of graph structures corresponding to the connected components remaining after F_i is eliminated. The second value, **next**, is a list of indices, one for each graph structure. Each element of the list, **next**$[j]$, is the index of the next manifestation node to eliminate from the graph structure, **CC**$[j]$. These return values are used in the recursive calls to *SQuickscore* distributed across the connected components. Note if only a single manifestation node remains, then *SQuickscore* recursion bottoms out and returns the marginal probability of the single positive manifestation.

Note that *SQuickscore* differs from recursive Quickscore only in the recursion step. In structural Quickscore, unlike recursive Quickscore, the recursive call is distributed over the connected components in each of the two networks.

Summary

Here, I have described structural Quickscore, an extension of the Quickscore algorithm which can exploit sparse connectivity in a BN2O network. This recursive algorithm is simple, easy to implement, and depending on the connectivity of the BN2O network, may give a significant speed-up over the vanilla version of Quickscore. The algorithm distributes the computation of the marginal probabilities across connected components whenever eliminating a positive manifestation disconnects the graphical model. This distribution leads to speed-ups with respect to vanilla Quickscore. The computation associated with determining the connected components is cached when the elimination order is determined. The recursive function, *SQuickscore*, can be used with any elimination ordering, here I have presented three heuristics to determine the order. In the next section, I present experiments done on the structural Quickscore algorithm to estimate its theoretical running time on samples from the QMR-DT.

3.7 Evaluating Structural Quickscore

3.7.1 Introduction

This section describes experiments done to test the ability of structural Quickscore to exploit the sparse structure of the QMR-DT to speed up the computation of the marginal probability of sets of positive manifestations.

Previously, D'Ambrosio [6] showed that the SPI algorithm was able to achieve significant savings in time complexity over Quickscore on some of the CPC cases. These results are promising because structural Quickscore exploits the sparse connectivity of the QMR-DT in the same way as SPI. D'Ambrosio showed that it was possible to reduce the effective number of positive manifestations in 9 CPC cases with between 23 and 29 positive manifestation to below the 20 positive manifestation threshold of tractability for Quickscore. Given these results, it is possible that structural Quickscore can be used to produce exact posterior marginals for more complex inference problems in the QMR-DT. This tractability would arise if it is possible, through eliminating a small number of positive manifestations, to divide the remaining positive manifestations into connected components whose size is small and doesn't depend upon the initial number of positive manifestations. One candidate for these connected components are the clumps of positive manifestations caused by the separate active diseases in the patient. If these clumps can be easily disconnected, then each connected component would only contain the number of positive manifestations typically caused by a single disease.

In this section, I evaluate whether any of the three heuristics introduced in section 3.6.3 is able to find elimination orderings that effectively separate the disease-specific clumps of positive manifestations.

3.7.2 Evaluation Techniques

Introduction

I evaluate the elimination orderings by comparing their theoretical performance to that of Quickscore. The time complexity of Quickscore is linear in the number of diseases in the BN2O network and exponential in the number of positive findings. On the other hand, the complexity of structural Quickscore depends upon when and how the network becomes disconnected. In the next section, I describe how I estimate the time complexity of structural Quickscore. I measure the complexity in *equivalent number of positive manifestations.* This metric is described below. Note that actually running structural Quickscore and Quickscore on each test case is infeasible. It is, however, possible to efficiently determine the time complexity of each of these algorithms for specific problems.

Structural Quickscore Running Time

Structural Quickscore's time complexity can be evaluated by tracking the recursion of *SQuickscore.* Each call to *SQuickscore* takes time proportional to the number of diseases in G before recursing, so the total time is the sum of the numbers of diseases at each node in the tree. Despite the exponential size of the recursion tree, this sum can be efficiently calculated because the same elimination ordering is used in each branch of the tree.

I do not include the time required to compute the elimination ordering in the estimate of the time complexity. The three heuristics compared here only take time that is polynomial in the number of positive manifestations to compute elimination ordering. Any exponential terms in the time complexity of structural Quickscore will usually outweigh any polynomial complexity.

Equivalent Number of Positive Manifestations

I use the *equivalent number of positive manifestations* to evaluate the algorithms and elimination orderings compared in this section. The equivalent number of positive manifestations, e, for a call to structural Quickscore that has estimated running time t is

$$e = \log_2 t - \log_2 |\mathcal{K}|,$$

where $|\mathcal{K}|$ is the number of disease nodes in the BN2O network. The estimated running time of structural Quickscore is $|\mathcal{K}|2^{|\mathcal{I}^+|}$. Note that the equivalent number of positive manifestations for Quickscore is $|\mathcal{I}^+|$. The number, e, is the number of positive manifestations that a BN2O network (with the same number of diseases) must have for Quickscore to have the same time complexity. Note that this metric measures only the time complexity, i.e. the running times of Quickscore and structural Quickscore up to a constant of proportionality.

3.7.3 Experiments

Samples

The experiments were run on four sets of samples from the QMR-DT. Each set contained 100 inference problems. The four sets were sampled from $P(\boldsymbol{f}, \boldsymbol{d}| \sum_k d_k = n)$ for $n = \{0, 1, 2, 5\}$. I estimated the running time of structural Quickscore on the BN2O network containing all of the multiparent positive manifestations and their disease parents.

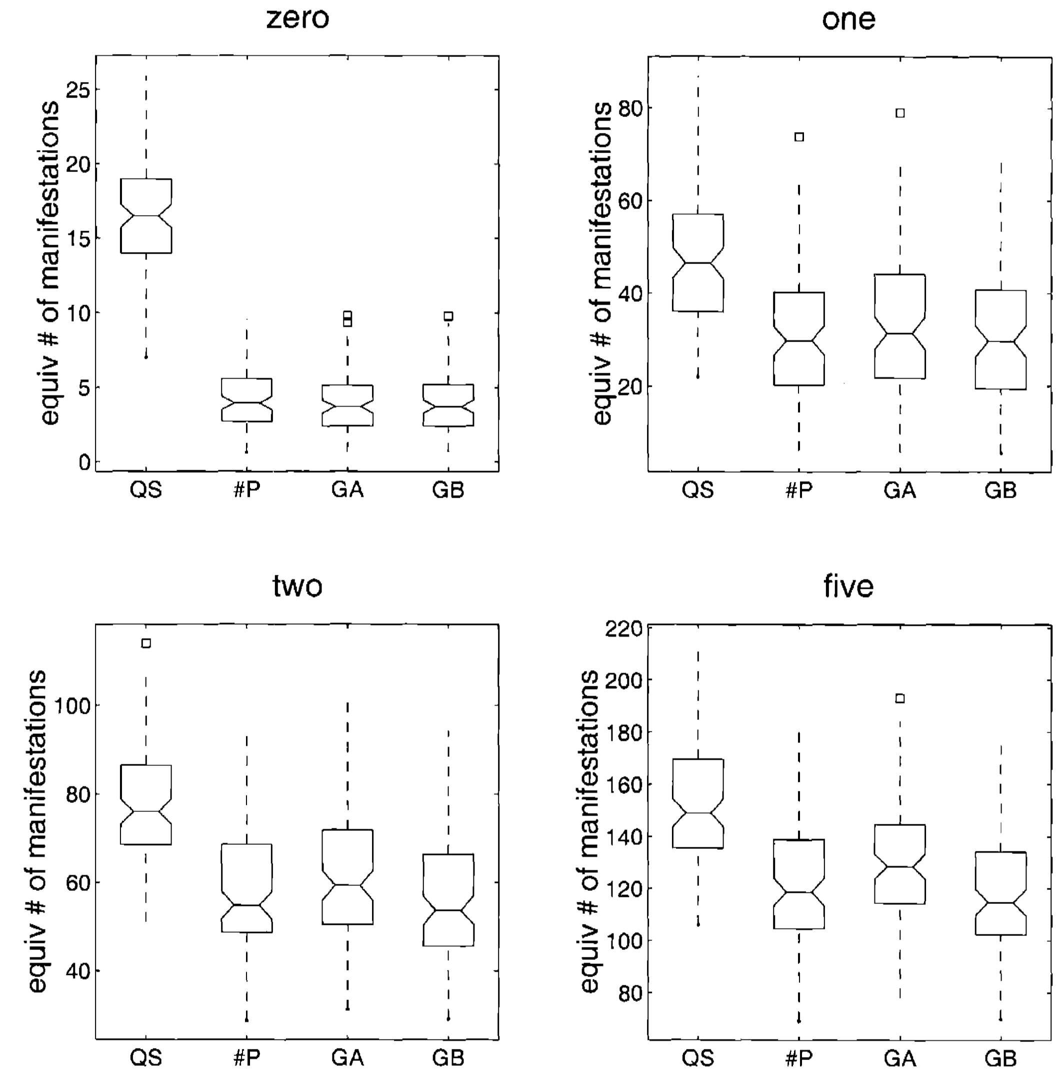


Figure 3-8: Comparison of elimination orderings. Figure shows results of elimination ordering comparisons on each of the four test sets. Results are shown with box plots. The center line in each box shows the median. The upper and lower lines are the upper and lower quartiles. The whiskers are the extent of the rest of the data up to a maximum distance away from the median. Points more than 1.5 times the interquartile distance away from the median are displayed with the square symbol. Each column represents a single inference method. QS, #P, GA, GB stand for Quickscore, *most parents*, *greedy A*, and *greedy B* respectively.

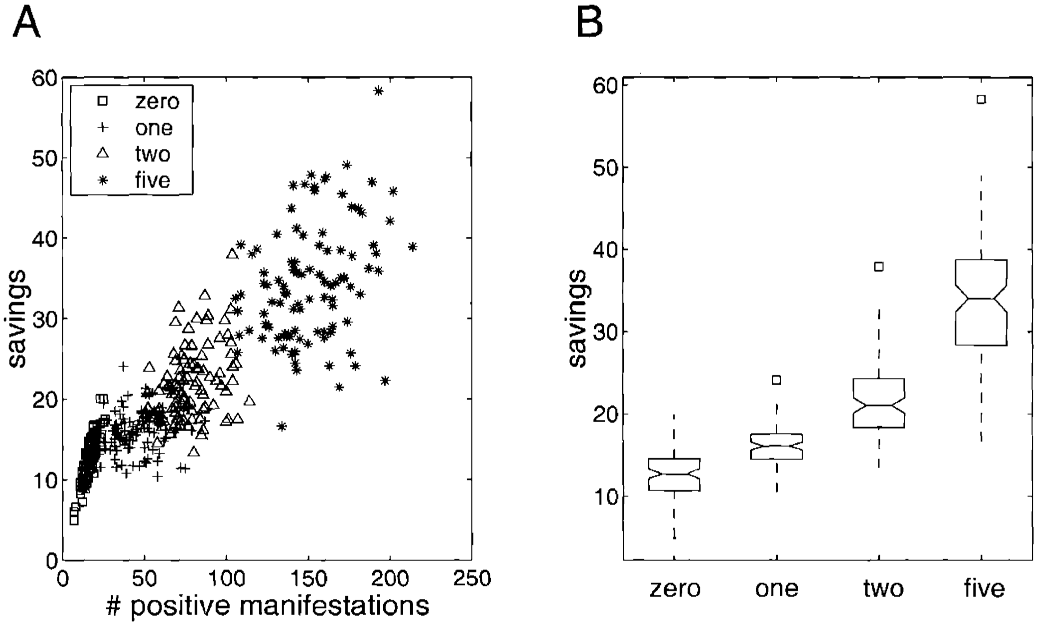


Figure 3-9: Savings using greedy B. A) The savings on all four datasets resulting from using greedy B. The savings is the difference between the number of positive manifestations and the equivalent number of positive manifestations under greedy B. B) Boxplot of the savings on each of the test sets. The interpretation of boxplots is described in figure 3-8.

Results

Figure 3-8 compares the equivalent number of positive manifestations of the three heuristics and Quickscore. The *greedy B* heuristic has the lowest median equivalent number for all four of the test sets.

Figure 3-9 investigates the savings in equivalent number of manifestations for the greedy B method further. Figure 3-9A shows the savings on all the samples and figure 3-9B shows the savings for each of the four datasets.

Notice that the mean savings for the zero and one datasets are very similar. This similarity occurs because in samples with one active disease, only the positive manifestations due to leak events can be disconnected from the main clump of positive manifestations caused by the active disease. The savings do, however, increase for the two and five disease cases, indicating that the algorithm can successfully disconnect the clumps of manifestations caused by the different diseases in the samples. This disconnection, however, comes at a great cost, for example, the equivalent number of positive manifestations for samples with five active diseases is more than one hundred.

Conclusions

In this section, I have evaluated structural Quickscore using samples from the QMR-DT. I have shown that the sparse connectivity of the QMR-DT does make structural Quickscore orders of magnitude faster than vanilla Quickscore. My experiments have shown the *greedy B* heuristic performs best of the three heuristics tested, though barely outperforming the *most parents* heuristic. I have further shown that structural Quickscore can indeed disconnect the clumps of positive manifestations associated with the each of the active diseases.

Unfortunately, however, the early promise of D'Ambrosio's results doesn't translate into tractability for problems with many more positive manifestations. Specifically, none of the heuristics manages to quickly disconnect the clumps of positive manifestations associated with the active diseases. There could be a number of reasons for this failure. One reason could be that a large proportion of the positive manifestations are children of more than one active disease. Because each of these manifestations would need to be eliminated before the clumps became separated, even the best possible elimination ordering would still be intractable. Another reason could be that there are positive manifestations from different clumps that are connected to the same disease nodes. In this case, all of these positive manifestations would need to be eliminated before the clumps could become disconnected. Eliminating all these manifestations could be computationally expensive. There are, however, ways to avoid this expensive procedure. One could instead marginalise over the disease nodes that bridge between the clumps. Though every disease node marginalised out would double the required number of computations, removing these bridging nodes may be less expensive than removing all of their positive children. Of course these bridging disease nodes would still need to be identified, which could in itself be a daunting task. However, failing the discovery of an algorithm for finding the bridging nodes, exact inference in the QMR-DT remains intractable.

3.8 Approximate Inference in the QMR-DT

3.8.1 Introduction

A number of different approximation strategies have been used on the QMR-DT. One strategy uses stochastic approximate inference. Shwe and Cooper [63] used self-importance sampling to estimate the disease posterior marginals. Using their method, they achieved reasonably accurate results on two CPC cases after substantial simulation time. They further showed that the accuracy of SIS was increased by using Markov blanket scoring to calculate the update to the disease posterior marginals implied by a single sample of the disease state. In Markov blanket scoring, the update is done using the distribution of the disease node conditioned on the sampled states of the other variables rather than by simply using the state of the node. With the addition of Markov blanket scoring, SIS is a feasible, though slow inference method. Later approximate methods, however, have achieved higher accuracy on some CPC cases with less computation time.

Another approach is to compute approximate posterior marginals by treating some of the positive manifestations exactly and the rest inexactly. The most extreme version of this approach is incremental SPI [6] in which the untreated manifestations are ignored. Jaakkola and Jordan [22] extended this approach by introducing a variational method (JJ99) to approximate the effect of the untreated manifestations. Their algorithm is described in depth in section 4.6.1. A significant advantage to their approach is that they give upper and lower bounds for the probability of the evidence $P(F^+_{\text{mp}})$. They use these bounds to evaluate the quality of other approximate methods and to choose which subset of the manifestations to treat exactly. JJ99 was shown to be faster and more accurate than both Gibbs sampling [21] and a self-importance sampling method [22] on inference problems for the QMR-DT.

Due to the high concentration of prior probability mass on sparse disease configurations, approximation methods based on enumerating disease configurations are feasible. Henrion [17] used properties of the noisy-OR to simplify a search for configurations with high posterior probability mass. Though his method worked well on inference problems with a small number of active diseases, he didn't apply his method to problems with many active diseases in the gold standard diagnosis.

More recent approaches include loopy belief propagation [40] and sequential inclusive trees [11]. The former method failed to converge on some of the tractable CPC cases and faired badly in comparison with the latter method. The sequential inclusive tree method is a type of adaptive density filtering [38]. The method fits a tree-structured distribution to the disease posteriors. The positive manifestations are absorbed iteratively into the distribution. Each time a new manifestation is absorbed, the distribution is updated. The new distribution minimises the KL-divergence to the old distribution and the disease posterior given the positive manifestation in question. This method depends on the order in which the positive manifestations are absorbed and has cubic time complexity in the number of manifestations.

In summary, approximation inference algorithms for the QMR-DT fall into one of two categories: ones that approximate the effect of a subset of the positive manifestations on the posterior and ones that enumerate (or sample) a subset of the disease configurations. I compare recognition networks with a method from the first category.

3.9 Summary

In this chapter I described the QMR-DT probability model and some common inference algorithms associated with it. The QMR-DT is a BN2O network that models the occurence and manifestation of disease. It has sparse connectivity and sparse disease priors which lead to a high concentration of probability mass on configurations with very few positive manifestations. These properties, along with structural properties of BN2O networks, have been exploited in a number of exact and approximate inference algorithms. I presented a recursive formulation of the Quickscore algorithm, the most prominent of the exact inference algorithms designed specifically for the QMR-DT. A small change in recursive Quickscore extended the algorithm to exploit the QMR-DT's sparse connectivity. This new algorithm, structural Quickscore, was evaluated on a set of samples from the QMR-DT to determine whether it could be used to establish a gold standard for evaluating approximate inference algorithms. Unfortunately, the exact inference for problems with large numbers of positive manifestations was still intractable. The chapter finished with a description of the major strategies for approximating inference in the QMR-DT.

Chapter 4

Recognition Models for Approximate Inference

4.1 Introduction

This chapter is devoted to detailing how to build recognition models for approximate inference. In this role it performs a number of functions. One function is to introduce a general framework for recognition models. Two new recognition models falling within this general class are described. One of these models, based on a multilayer perceptron, is further investigated here.

Another function of this chapter is to describe how to build recognition models for approximate inference in the QMR-DT. Inference in the QMR-DT using recognition models is especially difficult because of its large number of disease and manifestation nodes. Since a manifestation node can have one of three states: positive, negative, or unknown, a naive input encoding of the observation would require at least 4075×2 input units. A weight matrix fully connecting these inputs to an output layer that contained a unit for every disease node would contain more than four million weights! Much of this chapter is devoted to making the optimisation of recognition models for the QMR-DT more efficient. One method to speed up the optimisation of the recognition models is to reduce the number of parameters in the model. This speed-up results from using a concise encoding of the observation that has half as many inputs as the naive encoding but which nonetheless contains the same amount of information about the disease posterior.

The final function of this chapter is to demonstrate that recognition models are accurate inference methods for the QMR-DT. Toward this goal, I compare different recognition models to a benchmark approximate inference method.

This chapter is divided into seven sections. In section 4.2, I describe the general framework of recognition models and in section 4.3 I describe two examples of models that fit into the framework: multilayer perceptrons (MLP) and mixtures of experts (MoE). Both of these models use the same input encoding of the set of observed manifestations. This encoding summarises the observation into a concise input representation for the recognition model. This concise encoding is described in section 4.4. The recognition models evaluated on the QMR-DT are either LR network or MLP-based. Section 4.5 is devoted to describing the training of these models. In section 4.6, I present experimental comparisons between the algorithms and a benchmark approximate inference method. Section 4.7 contains a discussion of the results in the chapter.

4.2 General Framework for Recognition Models

A recognition model is a vector-valued function, parameterised by Ω, whose input is a configuration, $\boldsymbol{f}$, of the evidence variables and whose output is a vector of parameters, $\boldsymbol{z}$, that select a member, $Q(\boldsymbol{d}; \boldsymbol{z})$, of a pre-specified family of distributions over the unobservable variables. Typically the recognition model is composed of a fixed mapping $\boldsymbol{x}(\boldsymbol{f})$ followed by an Ω-dependent deformable mapping, $\boldsymbol{z}(\boldsymbol{x}; \Omega)$. Where its meaning is clear, I will simply use $\boldsymbol{z}$ to refer to the output of the recognition model when applied to the evidence, i.e. $\boldsymbol{z}(\boldsymbol{x}(\boldsymbol{f}); \Omega)$.

The fixed mapping $\boldsymbol{x}(\boldsymbol{f})$ encodes the evidence vector, $\boldsymbol{f}$, which may appear in a non-numeric form, into a suitable numeric form, $\boldsymbol{x}$. For example, the state of a non-binary categorical variable, e.g. $f_i \in \{+, -, ?\}$, may be encoded using a 1-of-N encoding scheme, e.g. $\boldsymbol{x}(f_i) \in \{(0,0,1), (0,1,0), (0,0,1)\}$.

Ideally, given a distribution $P(\boldsymbol{d}, \boldsymbol{f})$, the parameters, Ω, should be set so that for all evidence, $\boldsymbol{f}$,

$$\boldsymbol{z}(\boldsymbol{x}(\boldsymbol{f}); \Omega) = \underset{\boldsymbol{z}'}{\operatorname{argmin}} \operatorname{KL}\left[\, P(\cdot|\boldsymbol{f}) \,||\, Q(\cdot|\boldsymbol{z}') \,\right] \tag{4.1}$$

where $\operatorname{KL}\left[\, P(\cdot) \,||\, Q(\cdot) \,\right]$ is a measure of the divergence between distributions and has the form

$$\operatorname{KL}\left[\, P(\cdot) \,||\, Q(\cdot) \,\right] = \sum_{\boldsymbol{d}} P(\boldsymbol{d}) \log \frac{P(\boldsymbol{d})}{Q(\boldsymbol{d})}. \tag{4.2}$$

In practice, however, $\boldsymbol{z}$ may not be flexible enough to satisfy equation (4.1) for all evidence (indeed for any evidence). A more appropriate objective function for less flexible $\boldsymbol{z}$ is

$$E(\Omega) = \sum_{\boldsymbol{f}} P(\boldsymbol{f}) \operatorname{KL}\left[\, P(\cdot|\boldsymbol{f}) \,||\, Q(\cdot|\boldsymbol{z}(\boldsymbol{x}(\boldsymbol{f}); \Omega)) \,\right]. \tag{4.3}$$

This objective favours parameter settings that generate good approximate posterior for evidence vectors $\boldsymbol{f}$ with large marginal probability $P(\boldsymbol{f})$. However, since equation (4.3) contains an intractable sum, I approximate the objective function by sampling. Specifically, given a sample set $\{(\boldsymbol{d}^{(n)}, \boldsymbol{f}^{(n)})\}$,

$$\tilde{E}(\Omega) = \sum_{n} \frac{\log P(\boldsymbol{d}^{(n)}|\boldsymbol{f}^{(n)})}{\log Q\left(\boldsymbol{d}^{(n)}; \boldsymbol{z}\left\{\boldsymbol{x}(\boldsymbol{f}^{(n)}); \Omega\right\}\right)} \tag{4.4}$$

is a Monte Carlo approximation to equation (4.3). However, because the intractable $\log P(\boldsymbol{d}^{(n)}|\boldsymbol{f}^{(n)})$ is independent of Ω, minimising

$$\hat{E}(\Omega) = -\sum_{n} \log Q\left(\boldsymbol{d}^{(n)}; \boldsymbol{z}\left\{\boldsymbol{x}(\boldsymbol{f}^{(n)}); \Omega\right\}\right). \tag{4.5}$$

also minimises equation (4.4) with respect to Ω. Note that equation (4.5) is simply the negative log likelihood of the hidden configuration given the distribution selected by the recognition model. If $Q(\boldsymbol{d}; \boldsymbol{z})$ and $\boldsymbol{z}$ are partially differentiable with respect to both $\boldsymbol{z}$ and Ω, then the recognition parameters may be optimised using gradient-based, non-linear optimisation methods.

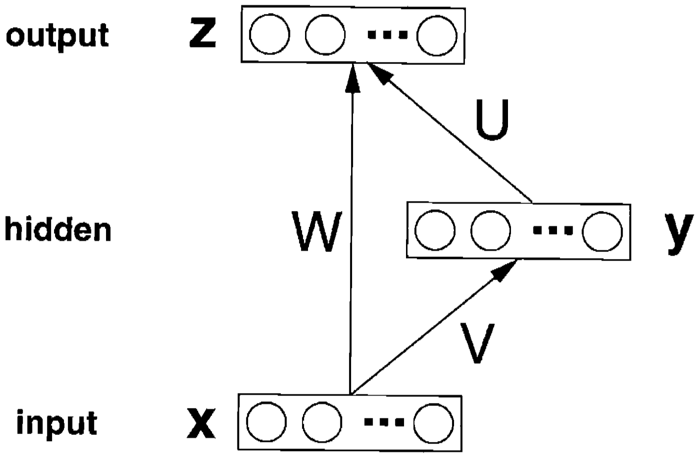


Figure 4-1: Architecture of the multilayer perceptron-based recognition model. The arrows between the layers indicate a weight matrix that fully connects the layers. The input layer contains an additional bias unit that is always on. The hidden layer non-linearity is the tanh function. The output layer non-linearity is the logistic function.

4.3 Examples of Recognition Models

This section describes two types of recognition models for binary hidden variables that will be used in this thesis. Both models are natural extensions of the logistic regression (LR) network-based recognition model described in section 2.7. Compared to the LR network, one extension, described in section 4.3.1, uses a more flexible mapping that is based on multilayer perceptrons. This extension, however, uses the same family of approximating distributions as the LR network. The other extension selects from a richer class of distributions and is described in section 4.3.2. This recognition model is based on a mixture of experts architecture.

4.3.1 Multilayer Perceptrons

A natural extension from LR networks are recognition models built using multilayer perceptrons (MLPs). The output activities z of the MLP have the same probabilistic interpretation as those of the LR network, however, the addition of a hidden layer enables a much more flexible mapping to the outputs. In fact, given enough hidden

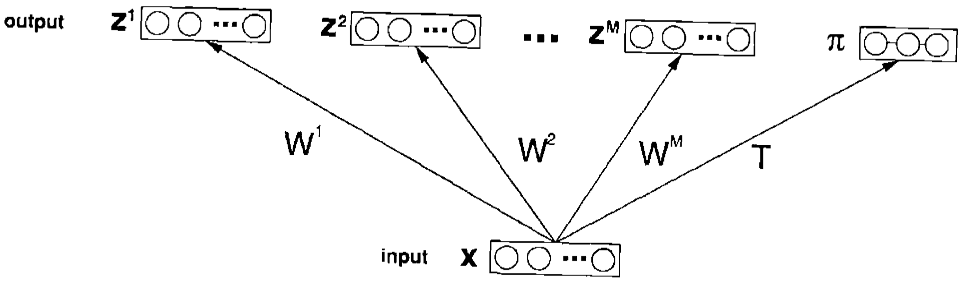


Figure 4-2: Architecture of the mixtures-of-expert-based recognition model. The arrows between the layers indicate a weight matrix that fully connects the layers. The input layer contains an additional bias unit that is always on. The π layer units are connected together, indicating that their values are tied via the multinomial logistic regression.

units, parameter settings exist so that MLPs can fit any continuous mapping [29], in principle, allowing equation (4.1) to be satisfied exactly.

I use an MLP with a single hidden layer, short-cut weights, and a bias unit. The general architecture of the MLP is shown in figure 4-1. I used the tanh function for the hidden unit non-linearity, as suggested by [46], to keep the outputs from the hidden units centered (i.e. zero meaned over the training set). Under this architecture, the activation of output unit z_k is given by

$$z_k(\boldsymbol{x}; \Omega_{MLP}) = \sigma(\boldsymbol{u}_k^\top \boldsymbol{y} + \boldsymbol{w}_k^\top \boldsymbol{x}),$$

where

$$y_j = \tanh(\boldsymbol{v}_j^\top \boldsymbol{x}).$$

4.3.2 Mixtures of Experts

Another way to extend the basic LR network recognition model is to use recognition models whose outputs parameterise a richer class of approximate posterior distributions. Here I use a mixture of factorial models as the class of distributions. Given a vector of mixing parameters $\boldsymbol{\pi}$ and a set of vectors of Bernoulli parameters $\{\boldsymbol{z}^m\}$, I construct a distribution, Q over the unobserved nodes $\boldsymbol{d}$ as follows:

$$Q(\boldsymbol{d}; \{\boldsymbol{z}^m\}, \boldsymbol{\pi}) = \sum_m \pi_m \prod_k (z_k^m)^{d_k}(1 - z_k^m)^{(1-d_k)}. \tag{4.6}$$

I use a mixture of experts architecture [24] to generate the evidence-dependent parameters $\boldsymbol{\pi}$ and $\{\boldsymbol{z}^m\}$. The architecture consists of two parts, a library of individual experts, $\{\boldsymbol{z}^m(\boldsymbol{x}; \Omega_{\mathrm{MoE}})\}$ and a gating network $\boldsymbol{\pi}(\boldsymbol{x}; \Omega_{\mathrm{MoE}})$ that outputs mixture weights for the experts. This architecture is shown in figure 4-2. I use LR networks for the individual experts, however, the MLPs can also be used.

The mixing proportions $\boldsymbol{\pi}$ are generated by a gating network. This network is either input-dependent or outputs fixed mixing proportions. The input-dependent network uses multinomial logistic regression, i.e.

$$\pi_m(\boldsymbol{f}; \Omega_{\mathrm{MoE}}) = \frac{\exp(\boldsymbol{t}_m^{\mathrm{T}}\boldsymbol{x}(\boldsymbol{f}))}{\sum_l \exp(\boldsymbol{t}_l^{\mathrm{T}}\boldsymbol{x}(\boldsymbol{f}))}, \tag{4.7}$$

to select the parameters. I use the vector $\boldsymbol{t}$ to parameterise the fixed mixing proportions, i.e.

$$\pi_m = \frac{\exp(t_m)}{\sum_l \exp(t_l)}. \tag{4.8}$$

The choice between input-dependent and fixed mixture proportions should depend upon the amount of training data available. Though input-dependent gating networks can be more accurate, they have more parameters and thus require more training data to ensure good generalisation performance. Mixture-of-expert based recognition models are used in section 5.6.

4.4 Concise Input Encoding

This section describes a concise encoding scheme of the observation vector to the input unit activities for recognition models for the QMR-DT. Under the concise scheme, the number of input units in the recognition model is less than the number of manifestation nodes. However, the concise encoding contains as much information about

Type of input	**Lossless**	**Concise**
bias	1	1
sparse	4075	2928
non-sparse	4075	570
Total	8151	3499

Table 4.1: Table comparing the lossless and concise encodings. The number of parameters in the LR networks is proportional to the total number of inputs. Inputs representing positive manifestations are guaranteed to be sparse. Inputs representing negative manifestation or log odds are not. Notice that the concise encoding has less than half as many parameters as the lossless encoding and about an eighth as many non-sparse inputs.

the disease posterior as the original observation.

The encoding uses two banks of input units, the *sparse* and the *non-sparse* banks. Each unit in the sparse bank is binary and corresponds to one of the 2928 multiparent manifestations. These units are only active when their corresponding manifestation is found to be positive. The units in the non-sparse bank are continuous-valued and collectively represent the effects of the negative multiparent manifestations and the single parent manifestations on the disease posterior. Specifically, the activity of a non-sparse input x_k is

$$x_k = \log \frac{P(D_k = 1|F^-, F^+_{\text{sp}})}{P(D_k = 0|F^+, F^+_{\text{sp}})} - \log \frac{p_k}{1-p_k}, \tag{4.9}$$

i.e. the log posterior odds that disease k is active given the negative and single parent positive manifestations minus the log prior odds that D_k is active. Section 3.3 shows how to calculate the log posterior odds.

This encoding is based on the BN2O network reduction described in section 3.3. It contains the same information as the reduced network, indeed the network can be constructed from the QMR-DT using the concise encoding. Because the disease posteriors in the reduced BN2O network are the same as those in the original network, no information about the disease posterior is lost by rerepresenting the observation using the concise encoding.

In addition to having a small number of input units, an advantage of the concise encoding is that the majority of the input units are only sparsely activated. Sparsity in the inputs speeds up both the prediction and the parameter fitting in the recognition model because the time complexity of many of the computations involved in both prediction and optimisation is proportional to the number of active input units.

Table 4.1 compares the number and type of inputs in the concise encoding to a different encoding of the observation. This encoding, the lossless encoding, represents the state of an evidence node using two units: a positive and a negative unit. The positive [negative] unit is only activated when the corresponding manifestation is positive [negative]. An unobserved manifestation is represented by a lack of activation in either of the two units. In this lossless encoding, the identity of every manifestation is explicitly represented, whereas under the concise encoding, the precise manifestation identity cannot always be uniquely determined. Because of this, the concise encoding could obscure information if the observation process is non-ignorable. This issue is discussed in depth in appendix B.

4.5 Training

4.5.1 Introduction

In this section, I describe the training of the recognition models. I optimise both LR networks and the MLP recognition models using a stochastic optimisation algorithm called stochastic meta-descent (SMD) [58]. This optimisation takes a long time to converge due to the scale of the problem, (e.g. the LR network has almost two million weights). This scale demands that the SMD algorithm be carefully tuned for convergence speed. However, even a carefully tuned SMD algorithm can take months to optimise an LR network. Another difficulty with training the models is the two different types of inputs. The optimisation of the weights from the sparse inputs needs to be carefully balanced with the optimisation of the weights from the non-sparse inputs. This balancing requires careful choice of the gain parameters of the SMD algorithm.

In section 4.5.2, I describe the basic SMD algorithm. Section 4.5.3 describes how the algorithm is used to fit recognition models based on LR networks. The training procedure for the MLPs is similar to the LR network procedure. The changes in the training protocol for the MLPs are described in section 4.5.4.

4.5.2 Stochastic Meta-Descent

Stochastic meta-descent (SMD) [58] is a local gain adaptation algorithm for non-linear minimisation using stochastic gradients. In SMD, the vector of parameters, Ω (hereafter called weights), are updated iteratively using gradient descent, i.e.

$$\Omega_{t+1} = \Omega_t - \boldsymbol{p}_t \odot \boldsymbol{g}_t \tag{4.10}$$

where $\odot$ denotes element-wise multiplication, $\boldsymbol{g}_t$ is the gradient of the objective function $E(\Omega)$ at Ω_t, and $\boldsymbol{p}_t$ is the vector of local gains (one for each weight). The gains are themselves updated multiplicatively via

$$\boldsymbol{p}_t = \boldsymbol{p}_{t-1} \odot \max(0.5, 1 + \boldsymbol{\mu} \odot \boldsymbol{v}_t \odot \boldsymbol{g}_t), \tag{4.11}$$

where the vector $\boldsymbol{\mu}$ is a set of fixed meta-gain step sizes.[1] The vector $\boldsymbol{v}_t$ is also iteratively updated, i.e.

$$\boldsymbol{v}_t = \lambda \boldsymbol{v}_{t-1} + \boldsymbol{p}_t \odot (\boldsymbol{g}_t - \lambda C_t \boldsymbol{v}_t) \tag{4.12}$$

where C_t is a curvature matrix (often the Hessian) defined at point Ω_t. The scalar λ, $0 < \lambda < 1$ implements a form of model trust region for the quadratic approximation implied by the curvature matrix. Setting $\lambda = 1$ and using the Hessian, H_t, at Ω_t for C_t, these update rules can be derived by doing dual gradient descent[2] of $E(\Omega)$ with respect

[1] In the SMD algorithm introduced by Schraudolph, μ is a scalar. However, I found it important to have weight-specific meta-gain step sizes.

[2] assuming that gradient descent is performed using equation (4.10)

to both Ω and $\log \boldsymbol{p}$ and using the approximation $e^u = \max(0.5, 1+u)$. Schraudolph shows that SMD converges more quickly than similar local gain adaptation methods on a benchmark problem using either $\lambda = 1$ and $C_t = H_t$ [58] or $\lambda < 1$ and $C_t = G_t$, the Gauss-Newton approximation to the Hessian [60].

Note that the product of curvature matrix and the vector can be calculated in time linear in the number of weights using R-propagation (see [49] for the Hessian and [60] for the Gauss-Newton).

The stability and convergence speed of the SMD algorithm depends critically on the selection of the meta-gain step sizes $\boldsymbol{\mu}$ and the initial values of the gains $\boldsymbol{p}^{\text{init}}$. Sections 4.5.3 and 4.5.4 describe how these values were selected for the LR networks and the MLP-based recognition models respectively. Note that due to the multiplicative gain updates, the SMD algorithm is still occasionally unstable even when the step sizes and initial gains are carefully selected.

In training the LR networks, I use $\lambda = 1$ and the Hessian for the curvature matrix. For increased stability, in the MLPs, I use $\lambda = 0.99$ and the Gauss-Newton approximation to the Hessian.

4.5.3 Logistic Regression Networks

Introduction

Here I describe the training of the LR networks. I divide the description of the training procedure into three parts: first I describe the initialisation of the weights, then I describe the changes to and parameter setting for the SMD algorithm, finally I describe how the mini-batch size was chosen.

The parameters of the LR network are:

- W^n, the weights connecting the non-sparse input units to the output units,
- W^s, the weights connecting the sparse inputs to the outputs, and
- $\boldsymbol{w}_0$, the weights from the bias unit to the outputs.

The architecture of the LR network is shown in figure 4-3.

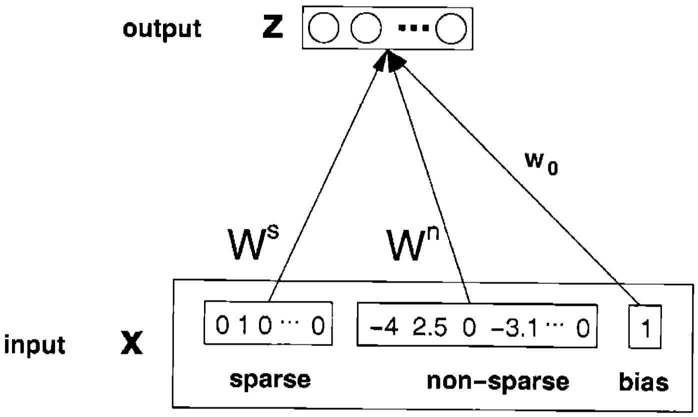


Figure 4-3: Structure of the LR network recognition models for the QMR-DT. Arrows connecting boxes indicate fully connectivity between the units in the boxes. The input vector $\boldsymbol{x}$ contains the *sparse*, *non-sparse* and *bias* input units. The bias weight vector $\boldsymbol{w}_0$ is optimised using the same parameter settings as the non-sparse weights.

Weight initialisation

The optimisation of the LR network parameters can be sped up considerably by choosing good initial weights. There is a natural choice for the initial value of some of the weights. Specifically, I use the identity matrix for the initial values of the weights leading from the non-sparse inputs, i.e. $W^n = I$, and I use the log prior odds as the initial values for the bias unit weights, i.e. $w_{0k} = \log p_k/(1-p_k)$. These are natural initial settings because if the observation doesn't include any multiparent positive manifestations, the LR network initialised in this way predicts the disease posterior marginals exactly. The prediction is exact because

$$P(D_k = 1|F^+_{\text{sp}}, F^-) = \sigma(x_k + \log \frac{p_k}{1-p_k}).$$

where x_k is determined by equation (4.9). All other weights, i.e. W^s, in the network are initialised to be zero.

Tailoring SMD

The behaviour of the SMD algorithm is governed by a number of parameter settings. Here I motivate my choices for the SMD parameters. I also describe an additional small change I made to the basic algorithm to further control its instability.

There are two sets of parameters that need be selected to implement the SMD algorithm: the value of the meta-gain step sizes $\boldsymbol{\mu}$ and the initial gains P^{init}. Each weight receives one of two step size and initial gain parameters, depending upon the input unit that it connects to. Specifically, the pair $(\mu_{ki}, P_{ki}^{\text{init}})$ associated with a particular weight W_{ki} takes the value

$$(\mu_i, \boldsymbol{P}_i^0) = \begin{cases} (\mu_s, P_s^{\text{init}}/N) & \text{if } x_i \text{ is a sparse input unit,} \\ (\mu_n, P_n^{\text{init}}/N) & \text{if } x_i \text{ is a non-sparse input unit,} \end{cases} \tag{4.13}$$

where N is the number of cases in the mini-batch. I found SMD to be most stable and converge most quickly when the parameters corresponding to the sparse input units were orders of magnitude larger than those for the non-sparse units.

Because of the multiplicative gain updates, SMD can be unstable, (see e.g. [60]). As an additional control to combat this instability, I sometimes bound the gains in the range $[10^{-10}, 10]$. This bounding increases stability but also slightly increases the processing time of a single mini-batch.

Choosing the mini-batch size

The stochastic gradient used by SMD is calculated on mini-batches of samples.[3] The size of the mini-batch plays a large role in the convergence speed of the optimisation. If the mini-batch size is small, weight updates happen frequently but the stochastic gradient may be quite noisy. If the noise level of the gradient is large, many updates may be required before the generalisation performance of the network improves. In this case, convergence to the minimum may be faster if the mini-batch size were

[3] Mini-batches are small sets of training examples

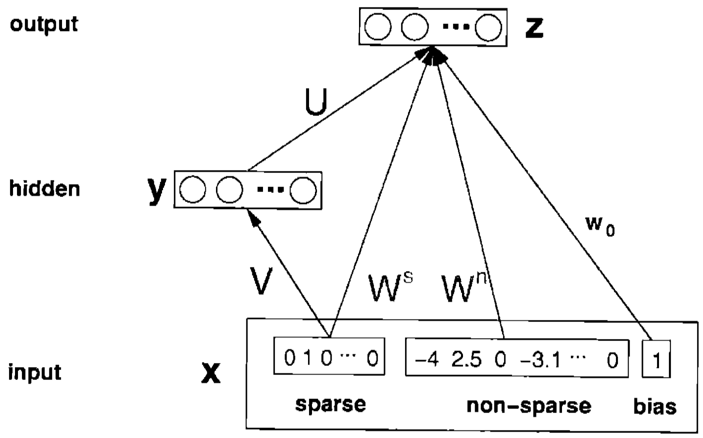


Figure 4-4: Architecture of the MLP trained on the QMR-DT. Arrows connecting boxes indicate fully connectivity between the units in the boxes. The input vector $\boldsymbol{x}$ contains the *sparse*, *non-sparse* and *bias* input units. The lightly shaded weight matrices W^s and W^n are fixed during training. Optimisation of the bias weight vector $\boldsymbol{w}_0$ copied from the LR network restarts using initial SMD parameter settings $(\mu_u, P_u^{\text{init}})$. Another bias unit (not shown) provides an additional input to the hidden layer. These bias weights use the same SMD parameter settings as the V weight matrix.

increased, thus reducing the noise in the gradient and eliminating the unnecessary computation involved in the unhelpful weight updates. On the other hand, if the noise is small then the gradient calculated on a subset of the examples in the mini-batch may be very close to that calculated using the whole batch. In this case, decreasing the size of the mini-batches would allow more constructive weight updates to be made for slightly more processing time.

Cache size is an additional consideration in choosing the mini-batch size. A mini-batch is processed much faster if all the data structures fit within the cache.

In the QMR-DT, the noise in the gradient is often quite high, as such, I typically use as large a mini-batch as the machine cache size will allow.

4.5.4 Multilayer Perceptrons

Introduction

The training protocol for the MLP-based recognition models is similar to that of the LR networks. Here, I discuss the differences in the protocols. There are three parts to this discussion: the setting of the short cut weights, the weight initialisation, and the tailoring of the SMD algorithm.

The architecture of the MLP is shown in figure 4-4.

Short cut weights

Schraudolph [59] showed that short-cut weights, if used properly, can speed the convergence of a MLP training algorithm and reduce the final error of the MLP. However, these weights only speed convergence when the slope is centered, i.e. the linear component of the error signal is removed before it is backpropagated to the hidden units (see [57] for details). Removing this component requires subtracting off the average value, across the training examples, of the error signal backpropagated to each hidden unit. However, if the slope is not properly centered, then the short cut weights slow down convergence by competing with the hidden units to model the linear part of the input to pre-sigmoid output mapping. Unfortunately, it isn't possible to center the slope exactly in a stochastic gradient descent algorithm, instead the average error signal needs to be estimated. Incorrect estimation adds additional noise to an already noisy gradient.

I avoid the need to center the slope by using a two-stage training protocol that ensures no competition occurs between the short-cut weights and the hidden units. In the first stage, only the short cut weights are optimised. In the second stage, the short cut weights are fixed and weights associated with the hidden units are optimised. Specifically, in the first stage, I train an LR network on the QMR-DT. and use its final weight matrix as the short-cut weights of the MLP. In the second stage, I update only the weights to and from the hidden units and the weights connected to the bias unit.[4]

[4] Note that the bias weights w_0 were originally trained with the short-cut weights but unlike the

Because the short cut weights are fixed and because they already model the linear part of the input to pre-output mapping, the error backpropagated to the hidden units needn't be centered.

Weight initialisation

The initial value of each weight u_{ji} and v_{jk} in the weight matrices U and V is sampled uniformly in the range $[-5.0 \times 10^{-4}, 5.0 \times 10^{-4}]$.

Tailoring the SMD algorithm

Like the LR networks, each weight in the MLP has its own meta-gain step size and initial gains. These parameter pairs can take on one of two values. The weights in V have pairs $(\mu_v, P_v^{\text{init}})$, the weights in U and the bias on the output have pairs $(\mu_u, P_u^{\text{init}})$.

4.6 Experiments

4.6.1 Benchmark Inference Algorithm

In this section, I use both a stochastic and a deterministic approximate inference algorithm to benchmark the performance of the recognition models. The stochastic algorithm is an importance sampling method called AIS-BN [5]. The deterministic algorithm is a variational method that I call V that was adapted from [21].

AIS-BN

The AIS-BN algorithm is a general two-stage adaptive importance sampling technique that may be using for any Bayesian network. I use 25000 samples in the initial phase to adapt the proposal distribution and 75000 in the subsequent stage to estimate the posterior marginals. I use the same training methods and parameter settings as ,

short-cut weights, I continue training the bias weights in the second stage.

except that I do not use the heuristic that sets the initial proposal distribution to be uniform in some cases.[5]

Though AIS-BN has never been directly evaluated on the QMR-DT, it was shown to be both faster and to generate more accurate posterior marginals than self-importance sampling [61] and likelihood weighting [61] on the CPCS network [50]. The CPCS network is a Bayesian network which like QMR-DT was derived from the QMR knowledge base and contains very similar parameters to the QMR-DT. Though likelihood weighting has previously been used on the QMR-DT [34], I chose AIS-BN as a benchmark algorithm because of the clear superiority of AIS-BN over likelihood weighting on the CPCS network.

V

The algorithm V generates posterior marginals by approximating the effect of each multiparent positive manifestation F_i with a variational parameter ξ_i. These parameters are set by minimising an upper bound, $P(\boldsymbol{f};\boldsymbol{\xi}^*)$, on the probability of evidence $P(\boldsymbol{f})$ that is implied by a upper bound, $P(\boldsymbol{f}|\boldsymbol{d};\boldsymbol{\xi}^*)$, on the likelihood function $P(\boldsymbol{f}|\boldsymbol{d})$. Here $\boldsymbol{\xi}^*$ is set using a standard non-linear optimisation routine. Given $\boldsymbol{\xi}$, the approximate posterior marginal q_k of disease k is

$$q_k = \sigma(\sum_i \xi_i \theta_{ki} + \log \frac{p_k}{1 - p_k}). \tag{4.14}$$

Note that V is a simplication of the algorithm described and tested on the QMR-DT in [21]. The version described here has previously been implemented in [11] and [39]. Results presented here should not be taken as indicative of the performance that the full version of the algorithm would have.

The full algorithm uses V together with a partial evaluation technique to generate posterior marginals. In this algorithm, a subset of the positive manifestations are treated exactly and V is used to approximate the effect of the remaining manifestations. This composite algorithm was shown to be faster and more accurate than both

[5] Initial tests showed using this heuristic leads to significantly worse performance.

Gibbs sampling [21] and self-importance sampling [22] on a set of medically realistic inference problems for the QMR-DT.

4.6.2 Evaluation Techniques

I compare algorithms using the average cross-entropy of their approximate posteriors on full and partially observed sets of samples from the QMR-DT. An algorithm's average cross entropy is equal to the average negative log likelihood of the sampled disease configuration under the algorithm's posterior $Q(\boldsymbol{d}; \boldsymbol{e}^{(n)})$ given the evidence vector $\boldsymbol{e}^{(n)}$, i.e.

$$\mathrm{CE} = -N^{-1} \sum_{n} \log Q(\boldsymbol{d}^{(n)}; \boldsymbol{e}^{(n)}) \tag{4.15}$$

where $\boldsymbol{e}^{(n)}$ is an evidence vector representing the observation, $e_i^{(n)} =?$ if the i-th manifestation is unobserved and is otherwise equal to the observed state of the i-th manifestation.

Notice the similarity between equation (4.15) and equation (4.5). The average cross-entropy is a Monte Carlo approximation to the expected KL divergence between the true and approximate posteriors plus an algorithm-independent constant. Algorithms with lower cross-entropy have lower expected KL divergence to the true posterior.

Note, however, that the average cross entropy metric is biased in favour of recognition models because it is exactly the objective function (calculated on the training set but not the test set) used to optimise the recognition model parameters.

The medical relevance of the average cross entropy is suspect. However in this chapter, I am only interested in how well recognition models approximate the posterior of the QMR-DT. A more medically relevant performance metric is described in section 5.4.

Network	μ_n	μ_s	P_n^{init}	P_s^{init}	# of mini-batches	total time (CPU days)
LR-10	0.01	0.5	10^{-4}	0.1	600800	15.0
LR-25	0.01	0.1	10^{-4}	0.1	395000	13.8
LR-50	0.01	0.1	10^{-4}	0.1	333000	12.9
LR-100	0.01	0.1	10^{-4}	0.1	655200	14.0

Table 4.2: Information on the training of the LR networks. This table gives the SMD parameter settings, the training times, and number of mini-batches used to train each of the LR networks. All network but LR-10 used mini-batches with 500 samples and had unbounded gains. The LR-10 network used smaller mini-batches (300 samples) and bounded gains. CPU times are measured on a 2.4 GHz Pentitum-4 processor with 512K cache.

Network	P_u^{init}	P_v^{init}	# of mini-batches	total time (CPU days)
MLP10	10	100	100000	3.5
MLP100	1	$1/\sqrt{10}$	150000	8.0
MLP1000	1	10	91000	13.5

Table 4.3: Information on the training of the MLPs. This table gives the SMD parameter settings, the training times, and the number of mini-batches used to train each of the multilayer perceptrons. All meta-gain step size parameters, μ, were set to 0.3. All MLPs used mini-batches with 500 samples and had bounded gains. CPU times are measured on a 1.4GHz Pentitum-III processor with 512K cache.

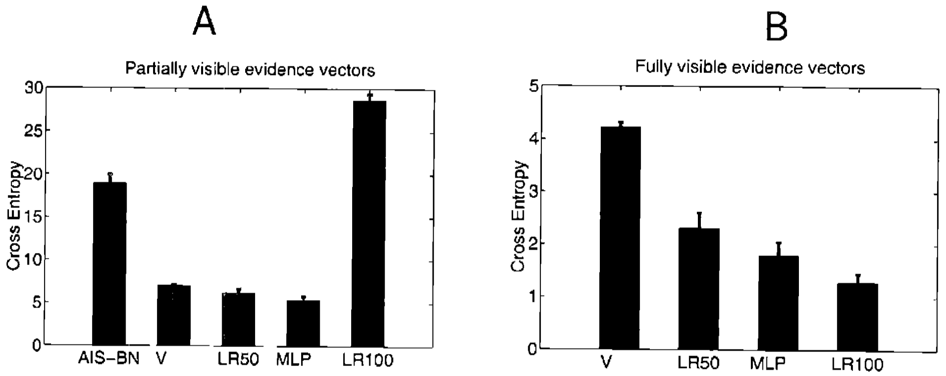


Figure 4-5: Performance comparison of LR and MLP-based recognition models against the variational method (V) adapted from [21] and an adaptive importance sampling method (AIS-BN) [5] on two test sets. The MLP and LR50 models were trained on partially visible samples, the LR100 model was trained on fully visible evidence vectors. All recognition models treat negative findings as being unobserved. In (A) the state of each manifestation in the test set was hidden with probability 0.5. In (B) posterior marginals for AIS-BN were unavailable. Bars in (A) and (B) show the mean bits of cross entropy (averaged over 1000 samples) between the reference diagnoses and the approximate posteriors generated by each method. The error bars show the upper bound of the symmetric 95% confidence interval of the mean. All differences in the graphs, except those between LR50 and MLP, are significant.

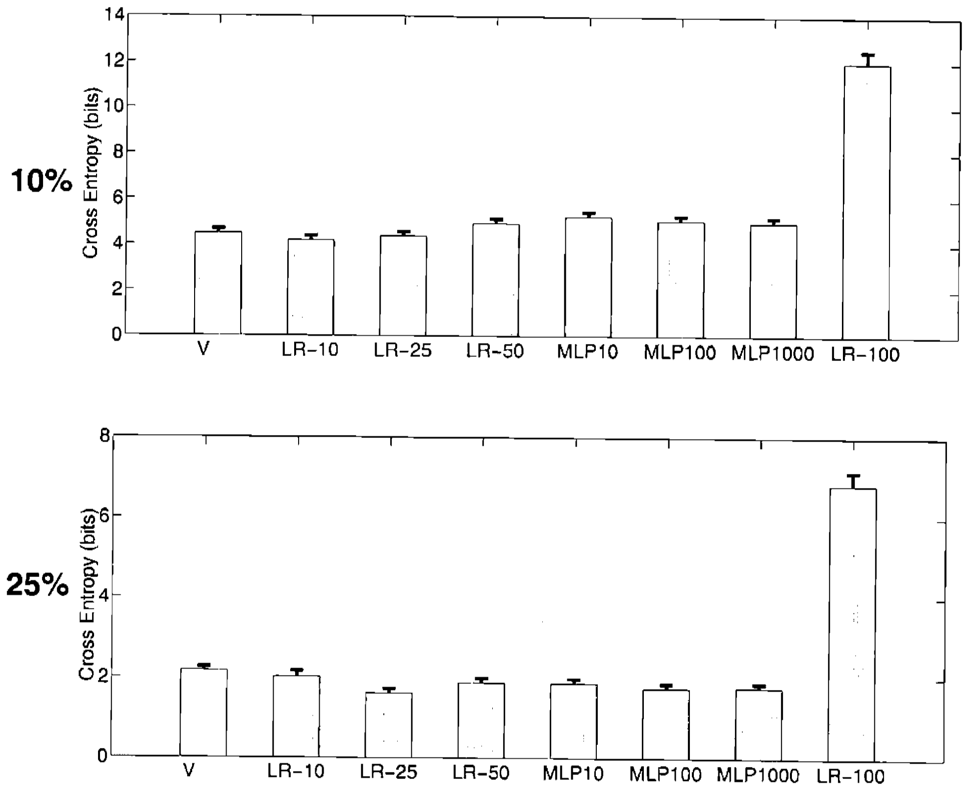


Figure 4-6: Performance comparison of LR network and MLP-based recognition models against V on two test sets. The LR-X models were trained on samples with $\phi = X/100$. The MLPX models were trained on samples with $\phi = 0.5$ and had X hidden units. Each subfigure represents a different test set of 3000 samples, the value of ϕ used to generate the test sets is given in the title bar. Bars in each subfigure show the mean bits of cross entropy averaged over the 3000 samples, between each sample's disease node configuration and the approximate posteriors generated given the sample's observed manifestations. The error bars show the upper bound of the symmetric 95% confidence interval of the mean.

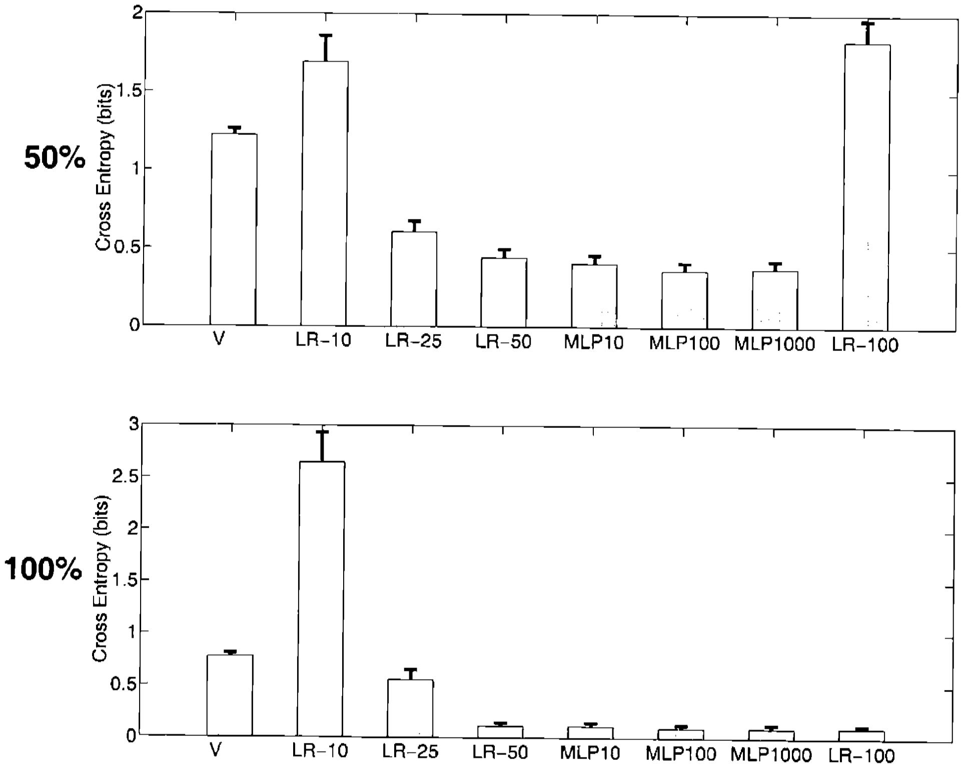


Figure 4-7: Performance comparison of LR network and MLP-based recognition models against V on two test sets. See caption of figure 4-6 for the interpretation of the figure.

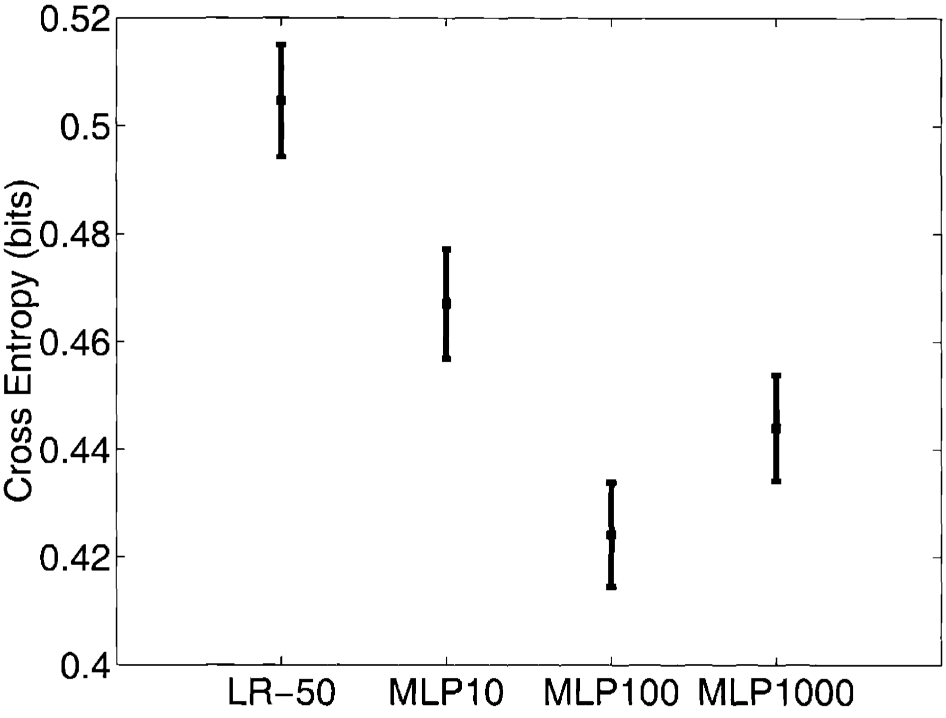


Figure 4-8: Performance comparison of LR network and MLP-based recognition models on $\phi = 0.5$ test set. Figure shows cross entropy of each method averaged across 10^5 samples from the QMR-DT with $\phi = 0.5$ The error bars show the 95% confidence interval of the mean. Note that the 1000 hidden unit network is still far from convergence. The cross-entropy of this network should decrease below that of the 100 unit network given more training time.

4.6.3 Results

I report results on two different versions of the QMR-DT. My preliminary results were generated using recognition models trained on the version of the QMR-DT used in [39]. These preliminary results show a clear superiority of V over AIS-BN. Because of this superiority, I compare recognition models to V and not to AIS-BN on the anonymised QMR-DT.

On both versions of the QMR-DT, I train recognition models on samples with both fully and partially observed manifestation vectors. Partially observed vectors are generated by randomly selecting a subset of the manifestations whose states are revealed. The state of each node is made available to the recognition model with probability ϕ.[6] Note when $\phi = 1$, the manifestation configuration is fully observed. The LR networks described in this section are labelled according to the value of ϕ that they were trained on. The notation LR-X indicates that the LR network was trained using the value $\phi = X/100$.

Preliminary Results

Here I report results generated using recognition models described in [39]. Note that the training protocol and input encoding of these networks is different than all other recognition models described in this thesis. I trained three recognition models: LR-50, MLP, and LR-100. The MLP had a hidden layer with 1000 units. All of these models ignored negative manifestations (i.e. a negative manifestation was interpreted as being an unobserved manifestation). Figure 4-5 shows results on fully and partial observed test sets. The disease vectors in the test set were sampled from $P(\boldsymbol{d}|\sum_k d_k = 5)$. Note that my implementation of the AIS-BN algorithm took two orders of magnitude more CPU time per sample than any of the other methods and was much less accurate. Based on these preliminary results, I chose to use only the variational algorithm, V, as the benchmark in the remainder of my experiments.

[6] This observation process is ignorable because the probability of revealing a manifestation doesn't depend on the state of the manifestation.

Anonymised QMR-DT Results

Here I train recognition models on samples generated using four different values of ϕ: $\phi \in \{0.1, 0.25, 0.5, 1\}$.[7]

I also train three MLPs on samples with $\phi = 0.5$. These networks have 10, 100, and 1000 hidden units respectively. The weights from the LR-50 network were used as the short-cut weights in each MLP.

I test the trained recognition models and the V algorithm on test sets sampled from the QMR-DT using $\phi \in \{0.1, 0.25, 0.5, 1\}$. The results of these tests are shown in figures 4-6 and 4-7. There are a number of points to draw from these figures. First note that the recognition models specialise to the value of ϕ that they were trained on. The LR network with the same value of ϕ as the test set never has significantly greater cross entropy than any other LR network. Also note that the recognition models generalise well to some untrained values of ϕ, but not all. However, V is more robust to changes in ϕ, every recognition model has significantly less average cross entropy than V for some ϕ values and significantly more average cross entropy for other values of ϕ. The V algorithm performs much better, compared to LR-100, on the $\phi = 1$ case in figure 4-7 than it does figure 4-5B. This improvement may come from the different test sets used in each figure. The disease vectors used in figure 4-5B contained exactly five active diseases whereas those in figure 4-7 are sampled from the QMR-DT prior.

Figure 4-8 studies the relationship between the LR networks and the MLP more carefully. This figures shows that adding additional hidden units significantly decreases the average cross entropy of the recognition model.

[7] To speed the training of the LR-100 network, I ignored the negative manifestation when calculating the non-sparse inputs. Because in the $\phi = 1$ condition, the manifestations were always fully observed, the effect of the ignored negative manifestations become incorporated into the bias weights

4.7 Discussion

In the experimental section, I have shown that recognition models are effective approximate inference algorithms when used to do inference on problems similar to those that they were trained on. However, recognition models do badly if the samples they are tested have many more or many fewer manifestations revealed than the aining set for the model. is effect occurs because of the objective function used to optimise the models, equation (4.4), preferentially rewards accuracy on inference problems that have high probability under the generative model used for training. Changing the observation process (i.e. the value of ϕ), changes the generative model, thus the observed set of manifestations is no longer a typical input for the recognition model and inferential accuracy degrades accordingly. I have also shown that the accuracy of a recognition model can be increased by adding a layer of hidden units. Increasing the number of hidden units further increases the accuracy of inference.

These findings suggest that recognition models with large numbers of hidden units can provide accurate inference when the observation process is well-characterised. Note, however, that training hidden units in addition to the short-cut weights is extremely time-consuming. In the remainder of the thesis, I use LR networks, which perform almost as well as the MLP recognition model but require less time and effort to optimise.

Chapter 5

Recognition Models for Medical Diagnosis

5.1 Introduction

This chapter describes recognition models that are designed to be used, together with the QMR-DT, as diagnostic support systems. Though QMR-DT was originally intended to be used for diagnostic support, it is, however, an incomplete model of the medical domain. Specifically, the QMR-DT doesn't contain an explicit model of the diagnostic procedure. The implicit model used by the QMR-DT assumes that the selection of the findings carries no information about the unobserved manifestation nodes, i.e. that the process that generated the observation was ignorable. This ignorability assumption is quite strong, yet has been made by all previous work on the QMR-DT. Here I show that this assumption is incorrect, that accurate medical diagnosis does indeed require a model of the medical diagnostic procedure. I do this in two ways: by reference to the diagnostic procedure and by examining a set of realistic medical problems encoded for the QMR-DT. Both of these inquiries show that the observation process is non-ignorable. To model some aspects of this non-ignorable observation process, I introduce the *diagnostic QMR-DT* (dQMR-DT), a model that contains both the QMR-DT and an observation process model. I use the dQMR-DT in the experiments described in the remainder of the chapter.

There are additional considerations when doing inference in the dQMR-DT rather than the QMR-DT. One important consideration is the evaluation of the inference algorithms for medical diagnosis. Evaluation of inference algorithm on the dQMR-DT is difficult because the addition of an observation process model makes inference much more difficult by destroying some of the structural properties of the QMR-DT that simplified inference. Another consideration is that inference algorithms for diagnosis should be evaluated on medically relevant endpoints. To address both these concerns, I introduce a metric that evaluates lists of diagnoses. These lists are natural endpoints of inference because, for example, they are used by physicians to represent uncertainty about the correct diagnosis. The diagnoses lists can be easily and efficiently generated from the posteriors typically produced by approximate inference algorithms. My metric, called the *posterior mass ratio*, compares the posterior mass covered by an algorithm's diagnoses list to that covered by a reference pool of diagnoses.

One of the changes in inference in the dQMR-DT versus the QMR-DT is that unobserved manifestations may no longer be removed from the graph without affecting the disease posterior. This change occurs because in the dQMR-DT, the fact that a manifestation is unobserved provides information about the unknown state of that manifestation. This change has bearing on the choice of input encoding for the recognition models. The concise encoding, described in section 4.4, no longer contains the same information as the observed configuration of manifestations. Despite this loss of information, there are still significant advantages to using the concise encoding, namely, the reduction in the number of parameters in the recognition model. Appendix B discusses this issue further and shows that the loss of information is not so severe as to overcome the advantages of the encoding.

I evaluate approximate inference algorithms for medical diagnosis under two different conditions. In one condition, I assume that the observation process is known and a probabilistic model of the process is available. In this condition, the procedure for training a recognition model for the dQMR-DT is similar to that for the QMR-DT. I describe and evaluate techniques for using the recognition model along with the dQMR-DT to generate good diagnoses lists. I also identify some of the types of errors that appear when the observation process is ignored. In the other condition, I assume that the observation process is only partially known. Specifically, I assume that the observation process is implicitly defined by a large number of presolved medical cases and some basic observation process models that are only accurate under limited conditions. To address this condition, I use a composite strategy: I train a library of individual recognition models on each of the provided observation processes and then combine their predictions using a gating network.

This chapter contains seven sections. Section 5.2 considers the observation process implied by the diagnostic procedure and observed in medical cases encoded for the QMR-DT. Section 5.3 introduces the diagnostic QMR-DT, a probabilistic model designed to represent the complete medical domain, that the recognition models are trained and tested on. Section 5.4 introduces the evaluation techniques used to evaluate approximate inference algorithms for medical diagnosis. Section 5.5 evaluates recognition models on a version of the dQMR-DT where the observation process is fully available. Section 5.6 evaluates recognition models on a dQMR-DT, when the observation process is only weakly-specified. Section 5.7 contains a summary and discussion of the chapter.

5.2 Observation Processes for Medical Diagnosis

5.2.1 Introduction

In this section, I describe the observation process implicit in the medical diagnostic procedure and show that the process is non-ignorable. Section 5.2.2 describes the medical diagnostic procedure and argues that elements of this procedure imply a non-ignorable observation process. Section 5.2.3 provides further evidence of a non-ignorable observation process by referring to properties of realistic medical cases encoded for use with the QMR-DT.

I assume that diagnostic problems encoded for inference are drawn from patient charts. The chart records all of the patient-specific information that may be relevant to the diagnosis. In this condition, the information available to the diagnostic support system is the same as that available to a physician who looked at the chart but has not had contact with the patient.

5.2.2 Diagnostic Procedure

Here, I give a brief description of some elements of the diagnostic procedure that are relevant to building an observation process model. The diagnostic procedure is a goal-oriented process that is contributed to by both patient and physician.[1] This procedure is multistage and often the selection of further investigations depends upon previous findings. Some of the manifestations of disease are only observable by the physician.

An important aspect of the diagnostic procedure is the different contributions of the patient and the physician. Here I clarify what those contributions are. Specifically, I divide the findings (i.e. observed manifestations) into two groups depending on who made the finding. I say that a finding was made by a patient if the patient spontaneously offers the information without being prompted for it by the physician. Otherwise, the physician is said to have made the finding. It is important to make this distinction because the patient and physician both use different decision processes to make their respective findings and also contribute different types of findings. Patients primarily contribute findings by reporting their symptoms, i.e. subjective manifestations of disease. Patients may also spontaneously offer elements of their medical history that they believe to be relevant. Physicians contribute to the procedure by making investigations. Investigations can be made by measuring signs (outwardly observable manifestations of disease), doing diagnostic tests, or by interviewing the patient and eliciting descriptions of symptoms.

A physician's main goal in choosing investigations is to determine how best to treat the patient. Choosing the best treatment often requires determining the patient's disease state with some certainty. If the disease state is uncertain, the chosen

[1] To simplify the presentation, I am assuming that only one physician is involved in the diagnostic procedure.

treatment may be harmful to the patient or may fail to address a potentially fatal condition. A physician should, therefore, consider investigations that best reduce uncertainty about the patient's disease state. Of course, some types of uncertainty (e.g. regarding disease states that should be treated differently, particularly those that are potentially fatal) are more important to reduce than others (e.g. regarding disease states with similar treatments).

Note, however, that because the state of the manifestation is not available until the investigation is actually performed, a physician must base their choice on the *expected* reduction in uncertainty that arises from making specific investigations. Calculating this expected reduction requires knowing the distribution over the state of manifestation being considered. This distribution depends upon the current uncertainty about the patient's disease state. Often the investigations that are most informative also have the property that their result is not easily predicted, i.e. the distribution over the manifestation has high entropy. Because manifestations are rarely positive, investigations predicted to be informative will usually lead to positive findings more often than randomly selected manifestations.

There are, of course, additional criteria for choosing investigations. For example, one should also consider the morbidity associated with making the investigation. Some investigations (e.g. a liver biopsy) are more harmful than others (e.g. measuring blood pressure).

A physician's behaviour can be modelled using Bayesian decision process with a cost function that weighs all of the criteria used to choose an investigation. Therefore, assuming that a physician is acting optimally, her actions at each stage can be completely predicted given the previously observed findings. However, because physicians may differ in how they value the different criteria and because physicians don't always act optimally, a probability distribution over the choice of investigation may be more appropriate. Nonetheless, one may assume that physicians only use the states of findings recorded on the patient's chart to select further investigations.

The patient also makes an important contribution to the diagnostic procedure. A major component of that contribution is their initial presentation of symptoms to the physician. This initial presentation provides the starting point for the physician's series of choices of investigations. The patient may continue to play a role throughout the procedure by spontaneously reporting additional findings to the physician.

However, the patient's contribution to the diagnostic procedure is very different than that of the physician. Specifically, patients can use internal information that isn't available to the physician when deciding which symptoms to report. Patients know, for example, what hurts and what doesn't hurt. Because a patient will generally report abnormalities (e.g. positive manifestations) more often than normalities, the patient's selection of findings will depend upon the states of manifestations not recorded in the patient's chart. Note also that there is a lot of variation between patients in how they select which symptoms to report. A stoic patient, for example, may not report a pain that a more sensitive patient would. Patients, however, are restricted in the findings that they can make because they don't have access to the same facilities as the physician (e.g. an MRI scanner).

In this section, I have described a number of important features of the medical diagnostic procedure. One feature is the temporal structure of the problem, findings are made sequentially and often the choice of investigations depends upon previous findings. This is especially true of the findings made by the physicians. Another feature is the different information used by patient and physician in their decision processes. Usually all the patient-specific information that physicians use to choose investigations is recorded on the patient's chart. Patients, on the other hand, have internal access to the states of unrecorded manifestations. Physicians and patients differ in the types of findings that they contribute. Patients are restricted in the types of manifestations that they can report. Physicians, however, by questioning the patient, can make any finding that a patient can make. A last important feature is a bias in the diagnostic procedure towards revealing positive manifestations. Patients are more likely to report abnormalities and physician are more likely to select investigations that are informative and therefore positive.

All of these features point to a non-ignorable observation process with a bias towards revealing positive manifestations. This conjecture is supported in section

Type	% positive	# positive	# negative	Sources
SAM	**49 ± 6**	**20 ± 4**	**18 ± 2**	MSH91
CPC	**67 ± 4**	**38 ± 3**	**19 ± 3**	MPM82, SC91, J97
QMR-DT	1.2	**49**	**4026**	

Table 5.1: Table comparing realistic diagnostic problems to fully observed samples from the anonymised QMR-DT. The entries in the table are empirical means. The displayed margins of error are 95% confidence intervals of the mean. Note that I have analytically calculated the expected number of positive findings in a sample from the QMR-DT. SAM and CPC diagnoses contained exactly one disease and between one and six diseases respectively. The expected number of diseases under the QMR-DT prior is 1.04. MSH91, MPM82, SC91, and J97 stand for [34, 37, 63, 21] respectively.

5.2.3 with evidence from the QMR-DT literature.

5.2.3 Evidence for Non-ignorability

Here I present evidence showing that the observation process which produces realistic medical cases for the QMR-DT is non-ignorable. This evidence comes from sets of realistic medical problems encoded for the QMR-DT.

There are two sets of realistic diagnostic problems associated with the QMR KB: the CPC and SAM cases. The CPC cases represent difficult diagnostic problems, with between one and six diseases in the reference diagnosis and a large number of positive findings [22]. The SAM cases are pedagogical diagnostic problems designed by experts which usually contain a single disease in the reference diagnosis [34]. These problems have been used to evaluate inference algorithms on the QMR-DT. Unfortunately, at present, no such diagnostic problems are available for the anonymised QMR knowledge base. However, I can use properties of these cases described in the literature to argue for non-ignorability.

The presence of a non-ignorable observation process in the diagnostic procedure is shown by two phenomena. First, there is an overabundance of positive findings in realistic medical cases, as shown in table 5.1, suggesting that the diagnostic procedure preferentially reveals positive findings. Supporting this interpretation is the second phenomenon: a systematic bias in true QMR-DT posteriors toward disease configurations that contain many false positives. For two CPC cases with gold standard diagnoses containing three and five diseases respectively, Shwe and Cooper [1991] report the true QMR-DT posterior gave highest mass to configurations containing 10 diseases. In the four CPC cases used in [22] where exact posterior marginals can be computed, the expected number of diseases under the QMR-DT posterior are 6.9, 4.0, 4.5, 4.6, whereas the gold standard diagnoses for these cases had 1, 1, 2, 2 diseases respectively. Since diseases are the primary cause of positive findings, overrepresentation of positive findings in the observed sample, would explain the overprediction bias in the QMR-DT posterior.

There are two properties of the diagnostic procedure, described in section 5.2.2 which could explain this bias. One possible cause is a patient preference for reporting abnormalities (i.e. positive manifestations) rather than normalities (i.e. negative manifestations). Another explanation could be a physician's bias to choosing informative investigations. Investigations thought by the physician to be informative will be more likely on average to lead to positive findings.

Whatever the source of the bias, its presence indicates that using information about the medical diagnostic procedure together with the QMR-DT could significantly improve the quality of the diagnostic inference; in particular, by removing systematic overprediction in the QMR-DT posterior. The difficulty lies in accurately modelling this observation process. In the following section, I propose a modelling framework for this process.

5.3 The Diagnostic QMR-DT

5.3.1 Introduction

This section introduces the diagnostic QMR-DT (dQMR-DT), a graphical model that combines the QMR-DT with an observation process model. This model will be used in the experiments described in sections 5.5 and 5.6. In the following, I first describe the general model framework and then a particular instantiation of the model.

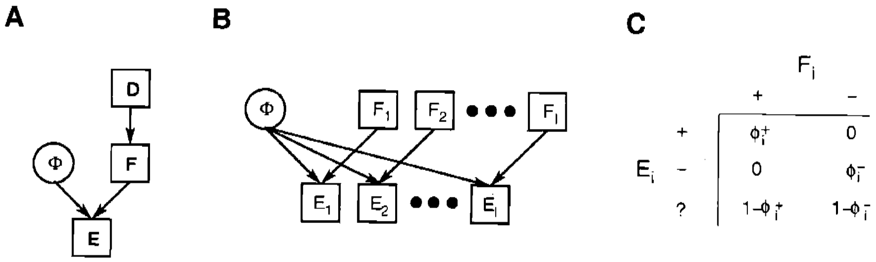


Figure 5-1: Diagnostic QMR-DT specification. A) Graphical models of the diagnostic QMR-DT (dQMR-DT). Squares indicate discrete-valued variables, circles are continuous valued variables. All variables shown in (A) are multivariate. The distributions $P(\boldsymbol{D})$ and $P(\boldsymbol{F}|\boldsymbol{D})$ are pre-specified by the QMR-DT. The observation process is conditioned on $\boldsymbol{\Phi}$ which has density $P(\boldsymbol{\Phi})$. B) A close-up of the observation process. The variables $\{E_1, E_2, \ldots, E_I\}$ and $\{F_1, F_2, \ldots, F_I\}$ are the elements of the random vectors $\boldsymbol{E}$ and $\boldsymbol{F}$ respectively. The dependence of $\{F_1, F_2, \ldots, F_I\}$ on $\boldsymbol{D}$ is not shown. C) The conditional probability table (CPT) of each finding. The dependence of $P(F_i|E_i, \Phi)$ on Φ is through the parameters (ϕ_i^+, ϕ_i^-) of the CPT.

5.3.2 General Framework

The diagnostic QMR-DT consists of two parts: the QMR-DT and the observation process (OP) model. The QMR-DT is connected to the OP model only through the manifestation nodes. The OP model contains a set of ternary evidence nodes, $\boldsymbol{E}$ ($E_i \in \{+, -, ?\}$), and a continuous-valued random vector $\boldsymbol{\Phi}$. The structure of the model is shown in figures 5-1A and 5-1B. The configuration of the evidence nodes represents the information available to the diagnosing physician (i.e. is on the patient's chart). Specifically, the state of an evidence node E_i is either the state of the corresponding manifestation node F_i or is '?' indicating that the state of the corresponding manifestation doesn't appear on the patient's chart. The vector $\boldsymbol{\phi}$ parameterises the conditional probability tables of the evidence nodes. Specifically, the vector

$$\boldsymbol{\phi} = [\phi_1^+, \phi_2^+, \ldots, \phi_I^+, \phi_1^-, \phi_2^-, \ldots, \phi_I^-]$$

contains two entries (ϕ_i^+, ϕ_i^-) corresponding to each evidence node E_i. These two values specify the conditional probability table $P(E_i|F_i, \boldsymbol{\phi})$ as shown in figure 5-

1C. The values ϕ_i^+ and ϕ_i^- are the probability that a manifestation variable will be observed given that it is positive or negative respectively.

There are a number of important aspects of this model. One feature is that with an appropriate selection of $\boldsymbol{\phi}$, samples taken from the dQMR-DT will contain the positive finding bias described in section 5.2.3. For example, a realistic setting of the ϕ_i^- values would have $\sum_i \phi_i^-/4075 \approx 0.01$ since only a very small proportion of the negative findings are ever observed (see table 5.1). Another feature of this model is that $P(\boldsymbol{d}, \boldsymbol{e}|\boldsymbol{\phi})$ can be computed efficiently since

$$P(\boldsymbol{d}, \boldsymbol{e}|\boldsymbol{\phi}) = P(\boldsymbol{d}) \prod_i \sum_{f_i} P(e_i|f_i, \boldsymbol{\phi}) P(f_i|\boldsymbol{d}). \tag{5.1}$$

Section 5.4 discusses the advantages of being able to tractably compute $P(\boldsymbol{d}, \boldsymbol{e}|\boldsymbol{\phi})$.

However, this tractability comes at the expense of approximation. This observation process model ignores the temporal component of the diagnostic procedure. Specifically, it does not allow a dependence of the choice of further investigations upon the results of previous investigations and unreported findings. However, $P(\boldsymbol{d}, \boldsymbol{e}|\boldsymbol{\phi})$ would be unlikely to be tractably computable in a more complicated observation process model. Furthermore, building a full model of the diagnostic procedure would require labels for the manifestation nodes and the disease nodes and exhaustive knowledge of diagnostic procedure, neither of which are currently available. Building this model would require a lot of resources to model all of the possible variation due to different physicians, patients, and hospitals. In the spirit of the QMR-DT, this observation process model is computationally attractive but oversimplified.

5.3.3 Parameterisation of $P(\boldsymbol{\Phi})$

In this subsection, I describe the specific form of $P(\boldsymbol{\Phi})$ that I use throughout the thesis. In particular, each value $\boldsymbol{\phi}$ with non-zero density under $P(\boldsymbol{\phi})$ is a deterministic function $\boldsymbol{\phi}(q, r)$ of the states of two univariate random variables, Q and R that are distributed in the range $(0.0, 0.5)$. Note that the form of the distribution $P(Q, R)$ dictates the distribution $P(\boldsymbol{\Phi})$. For convenience, I refer to these variables respectively as the *physician's* and the *patient's* contributions to the observation process. I also assign each of the manifestations into one of two categories: *presentable* and *testable*. A random half of the manifestations is assigned to each category. This assignment of manifestation to a category is fixed, i.e. it doesn't vary from sample to sample. The pairs $(\phi_i^+(q,r), \phi_i^-(q,r))$ that make up $\boldsymbol{\phi}(q,r)$ take on one of two values, depending on the category of the manifestation F_i:

$$(\phi_i^+(q,r), \phi_i^-(q,r)) = \begin{cases} (q+r, K^{\frac{r+3q}{2}}) & \text{if manifestation } i \text{ is presentable,} \\ (q, K^{\frac{3q}{2}}) & \text{otherwise.} \end{cases} \tag{5.2}$$

Here K is a constant used to ensure approximately equal numbers of positive and negative findings,

$$K = \frac{\langle \sum_i f_i \rangle_{P(\boldsymbol{f})}}{4075 - \langle \sum_i f_i \rangle_{P(\boldsymbol{f})}},$$

where $f_i = 1$ if $F_i = +$ and $f_i = 0$ otherwise.

The parameterisation of $P(\boldsymbol{\Phi})$ described here is simple but has similar aspects to the diagnostic procedure. One aspect of this model is the two different categories of manifestations. In the diagnostic procedure there are at least two different categories of findings: those that only a physician can make, i.e. diagnostic tests, and those that either a patient or a physician could make, i.e. symptoms or aspects of the patient's medical history. In $P(\boldsymbol{\Phi})$ this difference appears in the testable versus presentable categories of manifestations. However, since labels are unavailable, manifestation nodes are assigned randomly to each category.[2] Note, however, in the model, the choice of assignment of manifestation to the testable versus presentable categories bears no relationship to a medically relevant assignment to these categories. Another aspect captured by this model is the positive finding bias described in section 5.2.3. Also, depending on $P(Q,R)$, the observation process in the dQMR-DT could model different types of patients and stages in the diagnostic procedure. This (Q,R) manifold contains points, with low q and r values, where very few of the positive

[2] I include a list of these category assignments with the distribution of the MATLAB code for generating the QMR-DT from the anonymised QMR KB.

manifestations are likely to be observed and points, with high q and r values, where almost all of the positive manifestations are visible.

The observation processes described here are not accurate models of the diagnostic procedure but they provide an interesting model system. This system can be used both to evaluate the recognition model approach and to underline the importance of correctly modelling the observation process. This underlining is done, for example, by identifying systematic inaccuracies that result from mismodelling.

5.4 Evaluation Techniques

5.4.1 Introduction

This section describes the new evaluation metric called the *posterior mass ratio.* This ratio measures the quality of *diagnoses lists.* Diagnoses lists are an endpoint of inference that are both medically relevant and suggest a convenient way to incorporate additional information from the domain model. This additional information is added to a diagnoses list by a scoring process. Since the endpoint of most inference algorithms is a probability distribution over the unobservable variables, I also describe how to generate a diagnoses list from an approximate posterior.

5.4.2 Diagnoses Lists

I define a *diagnosis* to be a configuration, $\boldsymbol{d}$, of the disease variables $\boldsymbol{D}$. A diagnosis can be represented by the set of all the diseases k for which $d_k = 1$. Examples of diagnoses would be that a patient has no diseases, or that a patient has both the flu and a heart murmur.

An inference algorithm's *diagnoses list* is an ordered list of diagnoses that the algorithm predicts will have high probability under the posterior.[3] These lists are

[3] Note that these diagnoses lists are different from the *differential diagnosis list* often generated by physicians as part of the diagnostic procedure. A differential diagnosis list is a list of single diseases, each of which explain *some* of the findings. My diagnoses list is a list of *diagnoses* and each diagnoses is a, possibly empty, *set of diseases* and is intended to explain *all* of the findings.

both a convenient representation for incorporating additional information from the observation process and are a natural endpoint of medical diagnostic inference.

5.4.3 Scoring Diagnoses Lists

One advantage of using diagnoses lists is that in some cases additional information can be added to the list by scoring diagnoses. Where $\boldsymbol{e}^*$ is the evidence vector, I use $P(\boldsymbol{d}, \boldsymbol{e}^*)$ as the score of a diagnosis $\boldsymbol{d}$. Calculating this score requires an observation process model, $P(\boldsymbol{e}|\boldsymbol{f}, \boldsymbol{d})$. Given this model,

$$P(\boldsymbol{d}, \boldsymbol{e}^*) = \sum_{\boldsymbol{f}} P(\boldsymbol{e}^*|\boldsymbol{f}, \boldsymbol{d}) P(\boldsymbol{f}, \boldsymbol{d}). \tag{5.3}$$

However, this value, $P(\boldsymbol{d}, \boldsymbol{e}^*)$, is only useful if $P(\boldsymbol{e}|\boldsymbol{f}, \boldsymbol{d})$ has a form that makes computing equation (5.3) tractable. For example, under the observation process model described in section 5.3

$$P(\boldsymbol{e}|\boldsymbol{f}, \boldsymbol{d}) = \int_{\boldsymbol{\Phi}} dP(\boldsymbol{\Phi}) \prod_i \sum_{f_i} P(e_i|f_i, \boldsymbol{\Phi}). \tag{5.4}$$

If $P(\boldsymbol{\Phi})$ is appropriately chosen, in equation (5.3), e.g. $P(\boldsymbol{\Phi})$ is delta function centered at some value $\boldsymbol{\phi}^*$, then equation (5.3) can be efficiently calculated. If the true observation process model is not of the appropriate form, then that model may be replaced in equation (5.3) with an approximation, $\tilde{P}(\boldsymbol{e}|\boldsymbol{f}, \boldsymbol{d})$, that supports efficient calculation of an approximate score,

$$\tilde{P}(\boldsymbol{d}, \boldsymbol{e}^*) = \sum_{\boldsymbol{f}} \tilde{P}(\boldsymbol{e}^*|\boldsymbol{f}, \boldsymbol{d}) P(\boldsymbol{f}, \boldsymbol{d}). \tag{5.5}$$

Since $P(\boldsymbol{d}, \boldsymbol{e}^*)$ is proportional to $P(\boldsymbol{d}|\boldsymbol{e}^*)$, the ratio of the [approximate] scores of two diagnoses is equal to the ratio of their [approximate] posterior masses.[4] This score may, for instance, be used to rerank the diagnoses list in order of decreasing order

[4] Note that this condition is true of any function $S(\boldsymbol{d}, \boldsymbol{e}^*)$ such that $S(\boldsymbol{d}, \boldsymbol{e}^*) = Z(\boldsymbol{e}^*)P(\boldsymbol{d}|\boldsymbol{e}^*)$. $P(\boldsymbol{d}, \boldsymbol{e}^*)$ is a special case of these generalised scoring functions where $Z(\boldsymbol{e}^*) = P(\boldsymbol{e}^*)$.

of true posterior probability. Posthoc reranking can correct incorrect assumptions about the observation process made by the inference algorithms that generated the diagnoses list. A scored diagnoses list also implies a distribution, $\hat{P}$, over diagnoses,

$$\hat{P}(\boldsymbol{d}) = \frac{\sum_n \delta(\boldsymbol{d}, \boldsymbol{d}^{(n)}) P(\boldsymbol{d}, \boldsymbol{e}^*)}{\sum_n P(\boldsymbol{d}^{(n)}, \boldsymbol{e}^*)}, \qquad (5.6)$$

where $\delta(\boldsymbol{d}, \boldsymbol{d}^{(n)}) = 1$ if all elements of the two vectors are equal, otherwise $\delta(\boldsymbol{d}, \boldsymbol{d}^{(n)}) = 0$.

Note that $P(\boldsymbol{d}, \boldsymbol{e}^*)$ is only one possibility for the scoring function though it does have some convenient properties described in section 5.4.5. Other possible scoring functions include those that make use of a gold standard diagnosis $\boldsymbol{d}^*$, if one is available, and measure the cost of misdiagnosing the patient. One may also consider using the QMR-DT to score diagnoses, i.e. using $P(\boldsymbol{d}, F^+, F^-)$ where $\mathcal{I}^+$ and $\mathcal{I}^-$ are the indices of the positive and negative manifestations in $\boldsymbol{e}^*$.

5.4.4 Generating Diagnoses Lists

Many approximate inference algorithms output probability distributions over the unobserved variables. A diagnosis list may be generated from a probability distribution either by sampling or by enumerating the N most likely disease variable configurations. Nilsson [1998] gives an efficient algorithm to do the latter. Here, I always use the enumeration technique to generate the diagnoses list though I don't use Nilsson's algorithm. Instead I use an algorithm designed specifically for factorial distributions.

I generate two types of lists in my experiments. The first is the *basic* N diagnoses list. This list contains the N diagnoses with the highest posterior probability under the approximate posterior generated by an inference algorithm. When diagnoses may be scored, I also generate a *best* N *of* M diagnoses list. This list contains the N diagnoses with the highest score among the basic M diagnoses list for the approximate inference algorithm.

5.4.5 Posterior Mass Ratio

The *posterior mass ratio* measures the quality of a diagnoses list. This value is the ratio of the posterior mass of a diagnoses list to the posterior mass of a *reference list* of disease configurations. This reference list is used as the gold standard for the particular diagnostic problem. The posterior mass ratio R for an observation $\boldsymbol{e}^*$ is the ratio of the total scores of a diagnoses list, $\{\boldsymbol{d}^{(n)}\}$ to that of the reference list, L, i.e.

$$R = \frac{\sum_n P(\boldsymbol{d}^{(n)}, \boldsymbol{e}^*)}{\hat{P}(\boldsymbol{e}^*)}, \tag{5.7}$$

where

$$\hat{P}(\boldsymbol{e}^*) = \sum_{\boldsymbol{d} \in L} P(\boldsymbol{d}, \boldsymbol{e}^*). \tag{5.8}$$

Note that since $\hat{P}(\boldsymbol{e}^*) \leq P(\boldsymbol{e}^*) = \sum_{\boldsymbol{d}} P(\boldsymbol{d}, \boldsymbol{e}^*)$,

$$R \geq \sum_n P(\boldsymbol{d}^{(n)} | \boldsymbol{e}^*).$$

The tightness of this upper bound depends on how much of the posterior mass is covered by the disease configurations in L.

Note that computing the posterior mass ratio requires a probabilistic model of the observation process and furthermore that the scoring function $P(\boldsymbol{d}, \boldsymbol{e})$ be computable. If the observation process model does not support tractable scoring, then the posterior mass ratio may be calculated using an approximate score (see equation (5.5)).

5.4.6 Test Sets

I will be evaluating algorithms using diagnostic problem sampled from the diagnostic QMR-DT. A diagnostic problem consists of an observation $\boldsymbol{e}^*$ and its reference diagnosis $\boldsymbol{d}^*$. The observation $\boldsymbol{e}^*$ is sampled from $P(\boldsymbol{e}|\boldsymbol{d}^*)$ where $P(\boldsymbol{e}|\boldsymbol{d}^*)$ is implied by the QMR-DT and the true observation process model. Note that in sampled diagnostic problems, unlike real diagnostic problems, the reference diagnosis is not the gold standard, since $\boldsymbol{d}^*$ may not be the disease configuration that maximises $P(\boldsymbol{d}|\boldsymbol{e}^*)$.

On these sampled diagnostic problems, a reference list is constructed automatically using the *diagnoses pool* for the problem. This reference list is comprised of the N' highest scoring diagnoses among the union of the diagnoses pool and the reference diagnosis. The diagnoses pool for $\boldsymbol{e}^*$ contains the union of all diagnoses lists generated for $\boldsymbol{e}^*$ by the various algorithms being compared. Additional disease configurations may be added to the pool, if necessary, however, this is not done in the experiments reported here. Because the quality of the pool depends on the collective quality of the inference algorithms, the pool may only cover a small amount of the posterior mass if all the inference algorithms do badly.

5.4.7 Discussion

Previous work on the QMR-DT has used the set of disease posterior marginals as the output from the inference algorithm and has evaluated this set against gold standard disease posterior marginals either calculated exactly by Quickscore (e.g. [40]) or estimated by long sampling runs [22]. However, there are some advantages to using a reference list of diagnoses rather than posterior marginals for evaluation. One advantage is tractability of evaluation. There is currently no algorithm for efficiently computing exact posterior marginals for the dQMR-DT because, unlike unobserved manifestations in the QMR-DT, unobserved evidence nodes cannot be pruned from the dQMR-DT. Generating a reference list generally takes less time than a long sampling run. Another advantage to using reference lists is that the posterior mass ratio, unlike posterior marginal-based metrics, rewards inference algorithms that correctly capture posterior dependencies between diseases.

5.5 Known observation processes

5.5.1 Introduction

This section describes and evaluates techniques when a fully-specified probabilistic model of the observation process is available.

Networks	μ_n	μ_s	p_n^{init}	p_s^{init}
B-E, H-J	0.01	0.3	10^{-4}	0.1
K	0.01	0.1	10^{-4}	0.1

Table 5.2: Parameter settings for LR network SMD training algorithm. All networks were trained on 1.1×10^8 training examples using mini-batches containing 500 examples. These parameters were chosen to be similar to those used to train the LR networks in section 4.5.3.

The first technique uses a two-pass process. In a first pass, an inference method generates a basic M diagnoses list. The inference method used in the first pass may make incorrect assumptions about the observation process, for example assuming that the process is ignorable. In the second pass, the basic M list is scored using the complete domain model (i.e. one incorporating the QMR-DT with the correct observation process model) and a best N of M diagnoses list is constructed. This second pass may repair the damage done by incorrect assumptions made in the first pass. This two-pass approach is appropriate when inference assuming an ignorable (or inaccurate) observation process is much easier than inference in the complete model; which is particularly true in the dQMR-DT. Note that this two-pass technique is only usable when the observation process model supports efficient scoring of diagnoses. In this thesis, I do not investigate the feasibility of using approximately scored diagnoses in the second pass.

However, many observation process model that do not support efficient scoring, do support efficient sampling. My second technique uses samples from the complete domain model to train a recognition model. The two techniques may be combined by using a recognition model trained on the correct observation process to build the diagnoses list used in the first pass of the two-pass technique.

5.5.2 Preliminaries

In this section, I use the dQMR-DT with a fixed, known value $\boldsymbol{\Phi} = \boldsymbol{\phi}$, to represent the complete domain model. The inference algorithms have access to both the dQMR-DT and the given values of $\boldsymbol{\phi}$. Because $\boldsymbol{\phi}$ is fixed, equation (5.4) can be tractably

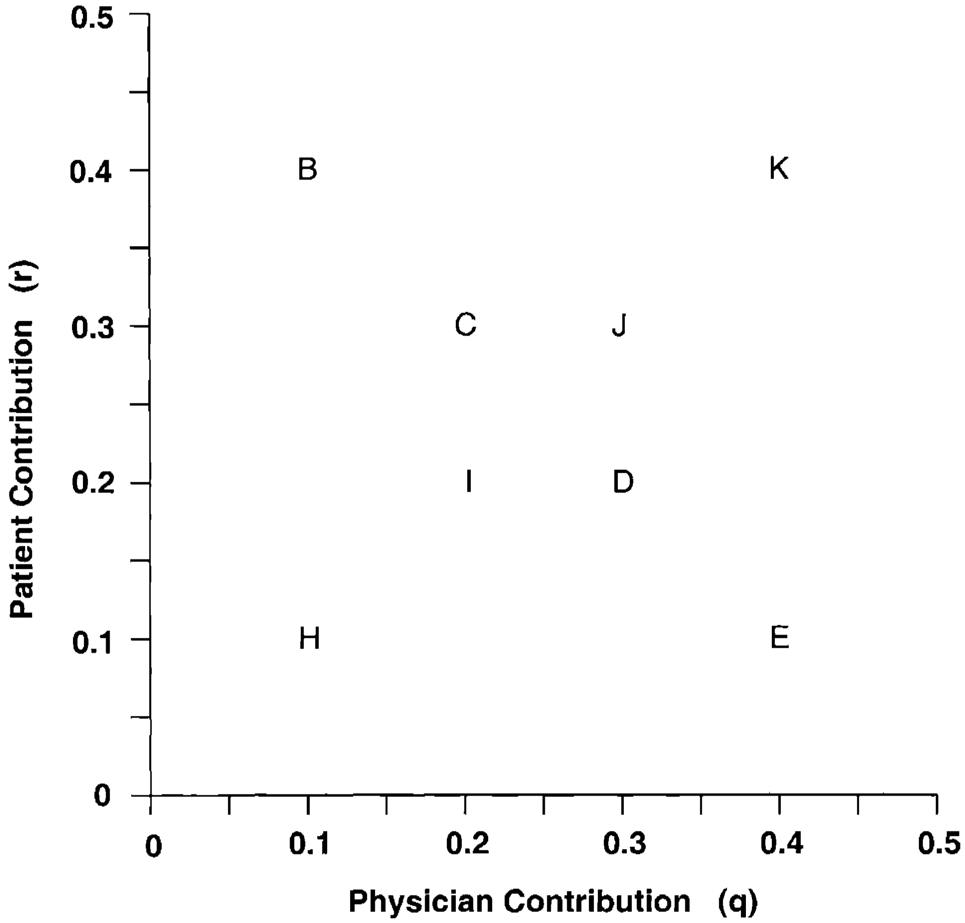


Figure 5-2: Locations of eight simple observation processes. Each point in the (q, r)-space maps into a value of $\phi(q, r)$. The letters (B-E, H-K) label the eight observation processes used to train the LR networks and evaluate the inference algorithms.

computed and thus diagnoses can be efficiently scored. This efficient scoring enables the use of the posterior mass ratio (PMR) to compare algorithms. However, though the observation process contained in the dQMR-DT does support efficient scoring, not all observation processes (OP) do. Here, I use the dQMR-DT observation process to represent both types of OP models.

All results reported in this section were generated using test sets containing 100 diagnostic problems. Each diagnostic problem contained a reference diagnosis, $\boldsymbol{d}^*$, sampled from $P(\boldsymbol{d} \mid \sum_k d_k = 5)$, i.e. the QMR-DT disease prior conditioned on there being exactly five diseases in the reference diagnosis. The corresponding evidence vector $\boldsymbol{e}^*$ was sampled from $P(\boldsymbol{e}|\boldsymbol{d}^*, \boldsymbol{\phi})$, the conditional distribution implied by the dQMR-DT for the provided value of $\boldsymbol{\phi}$.

Figure 5-2 describes the eight different observation processes {B-E, H-K} used in the experiments. I trained an LR network on each of the eight processes, {LR-[B-E], LR-[H-K]}. Table 5.2 shows the SMD gain parameters used in the networks. I used similar training sets[5] and the same SMD parameters to train all of the LR networks except those trained for observation processes I and K. The SMD minimisation diverged for both of these networks. Changing the training set was sufficient to achieve stable convergence for observation process I. However, the SMD minimisation for the LR network for K was only stabilised when the meta-gain parameter μ_s was lowered. This lowering reduced the large variation in the gain parameters that was the cause of the instability for network K. This large variation in the gain parameters could be due to the fact that K reveals a much larger proportion of the manifestations than any of the other processes. The total training time for the networks ranged from a minimum of 17.6 CPU days[6] for LR-B to a maximum of 20.0 CPU days for LR-K.

To evaluate inference algorithms, I use the posterior mass ratio of both the basic 20 diagnoses list, and the best 20 of 1000 list. The basic 20 list is the best an algorithm can do if diagnoses cannot be efficiently scored. The size 20 was selected for the list

[5] All disease and manifestation vectors were the same, but the evidence vectors were different because of the differences in the observation processes.

[6] On a 1 GHz Pentitum-III processor with a 256K cache

to limit the list size to manageable length for a physician. The upper limit of M was selected because of timing considerations. Generating and scoring a 1000 diagnoses list took between 4 and 6 seconds[7] for each set of posterior marginals.

5.5.3 Results

I compare the inference algorithms under two different conditions. In the first condition, I assume that diagnoses cannot be tractably scored under the provided observation process model. In this condition, I compare two approaches: using a basic M list and scoring diagnoses assuming an ignorable observation process, i.e. using the QMR-DT. In the second condition, I assume that diagnoses can be scored and that it is possible to build a best M of N list.

Figures 5-3 and 5-4 show a comparison of three different ways of selecting 20 diagnoses out of a basic 1000 diagnoses list. *Basic 20* selects the 20 most probable diagnoses under the approximate posterior (i.e. the basic 20 diagnoses list), *QMR 20* selects the 20 most probable under the QMR-DT, and *Best 20* selects the 20 most probable under the dQMR-DT with the correct observation process (i.e. the best 20 of 1000 diagnoses list). These methods were compared using the PMR calculated on 100 samples from $P(\boldsymbol{d}, \boldsymbol{e} | \sum_k d_k = 5, \boldsymbol{\phi}(q, r))$ where the pair (q, r) defines the observation process. The diagnoses pool for each sample consisted of the reference diagnosis and the basic 1000 diagnoses lists for the LR networks for all the observation processes $\{$LR-[B-E], LR-[H-K]$\}$ plus the LR networks $\{$LR-10, LR-25, LR-50, LR-100$\}$ and the V algorithm from chapter 4. Results are displayed using box plots, the interpretation of box plots is described in figure 3-8. On each observation process, I show a box plot for the LR network trained for that process along with the box plot of the approximate inference method from chapter 4 with the highest median *Best 20* PMR.

There are a couple of points to draw from figures 5-3 and 5-4. The first point is that being able to score diagnoses under the correct observation process model is a huge advantage. The 25th percentile PMR of *Best 20* is never lower than 0.8 even

[7] On a 2.4 GHz Pentitum-4 processor with 512K cache

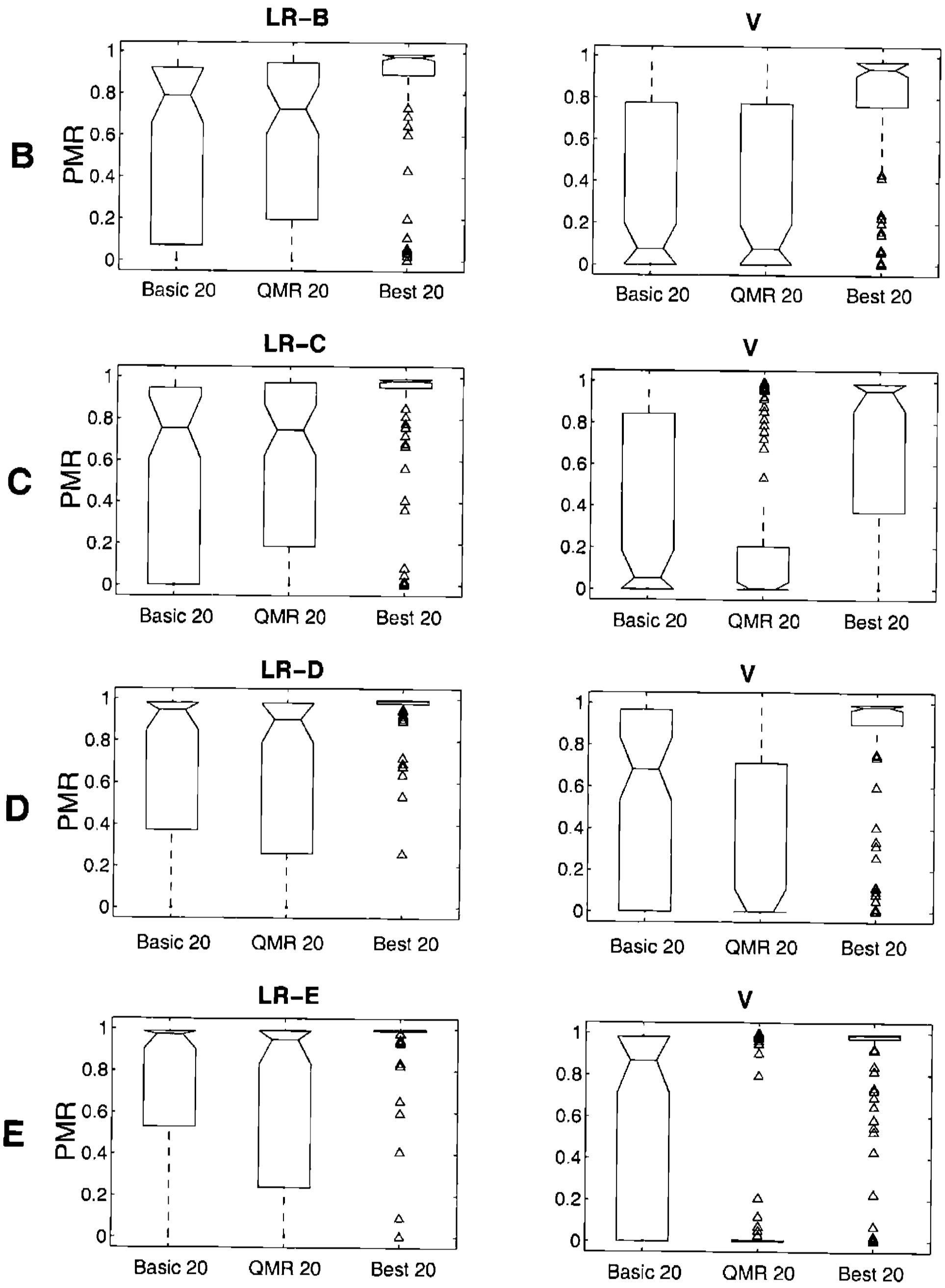


Figure 5-3: Performance of various inference methods on observation processes B,C,D,E. For each approximate inference method and each observation process, I compare three different ways of selecting 20 diagnoses out of the algorithm's basic 1000 diagnoses list. Further description is provided in the text. The title of each of the eight subfigures is the approximate inference algorithm used to generate the diagnoses lists analysed in the figure. Each row of subfigure matrix is labelled with the observation process analysed in the row.

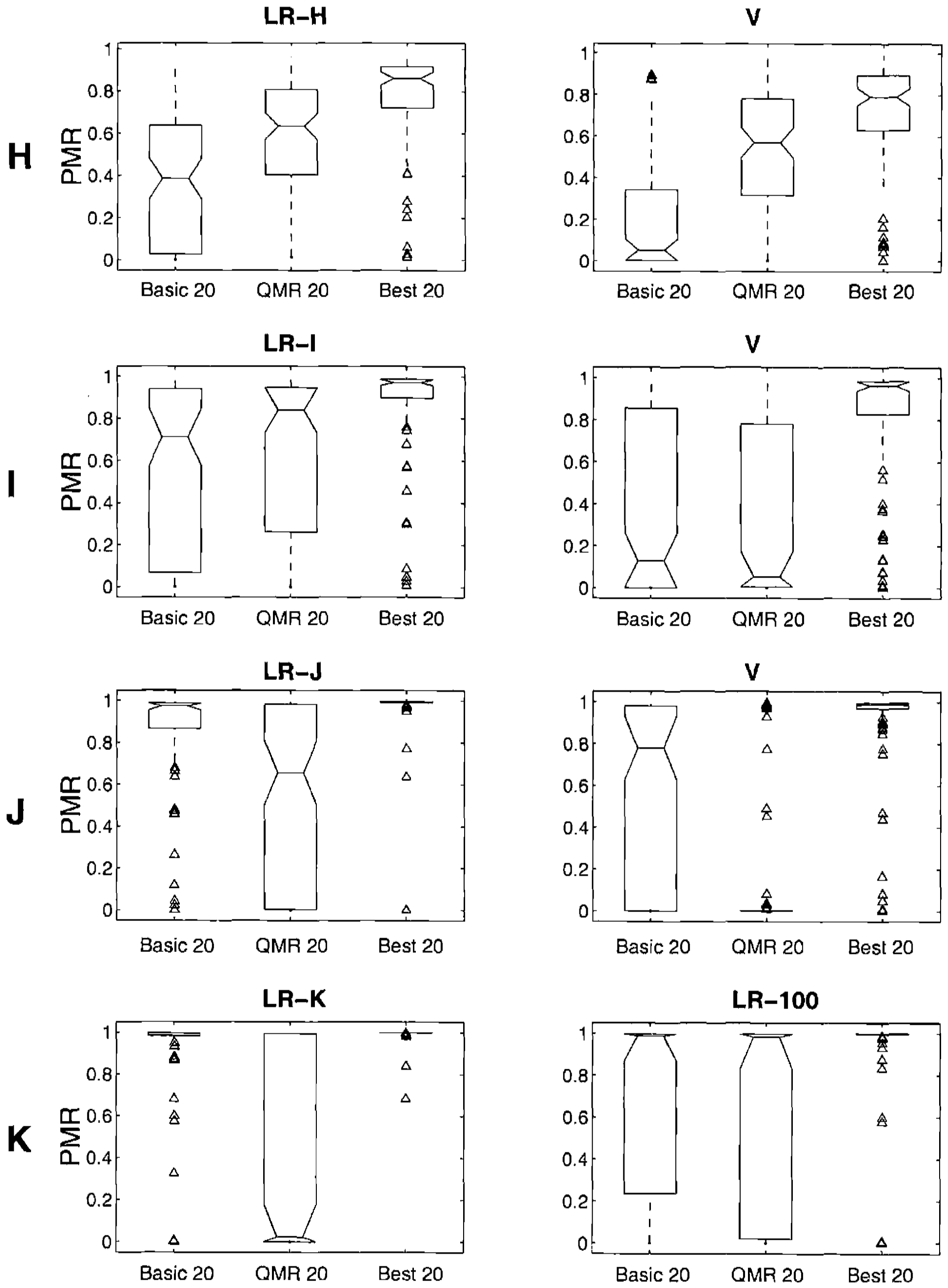


Figure 5-4: Performance of the LR networks and V on observation processes H,I,J,K. See figure 5-3 and the text for a description of this figure.

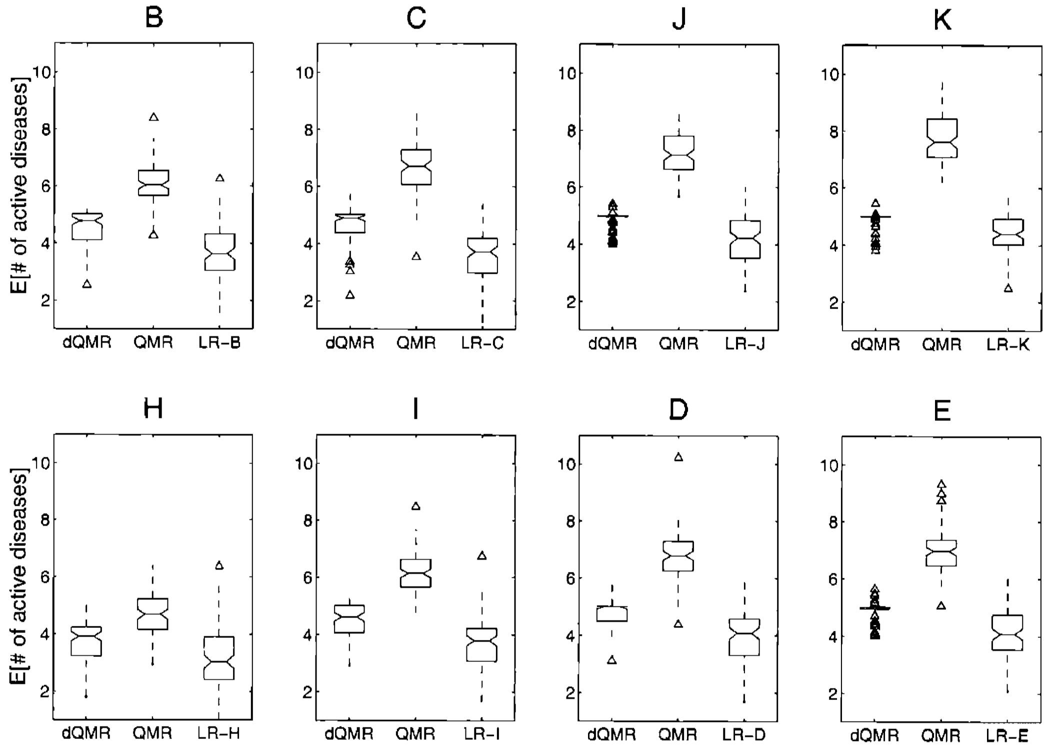


Figure 5-5: Comparison of the expected number of diseases under the dQMR-DT, QMR-DT, and LR network posteriors. The title of each subfigure identifies the observation process analysed. Each subfigure shows the expected number of active diseases under the three approximate posteriors over the 100 sample test sets described in section 5.5.2 for the given observation process. Note that the average probability that a positive manifestation will be observed increases from the lower left corner of the figure (H) to the upper right hand corner (K). For each sample, the dQMR and the QMR posteriors were derived from the scored diagnoses pool (where diagnoses were scored using the dQMR-DT and the QMR-DT respectively) using equation (5.6). Note that because of the sparse disease prior, observation processes which produce fewer findings (e.g. H) have an expected number of diseases that is less than five, despite the five active diseases in the reference diagnoses.

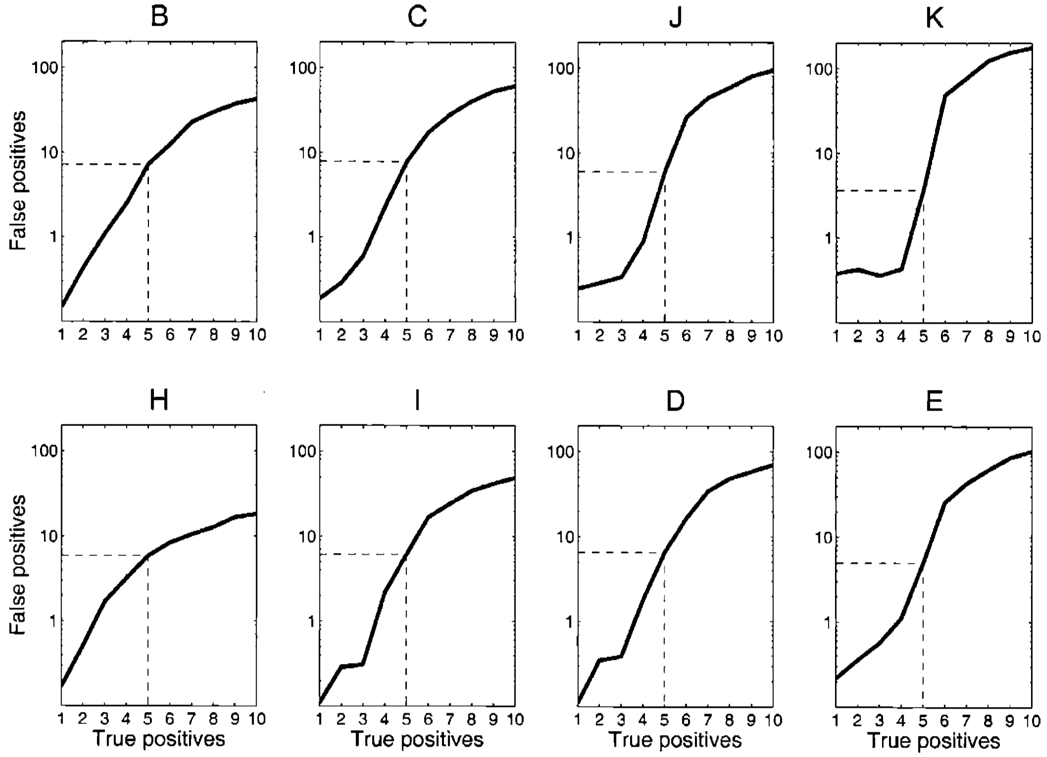


Figure 5-6: Comparison of the ranking of disease posterior marginals under the dQMR-DT and QMR-DT. The title of each subfigure identifies observation process analysed. Each subfigures compares the average rank among the QMR-DT posterior marginals of the 10 diseases with the highest posterior marginals under the dQMR-DT on the 100 sample test sets described in section 5.5.2. Specifically, the false positive value corresponding to true positive value N is $N' - N$ where N' is the average size of the list of the highest ranking diseases under the QMR-DT that contains all of the N highest ranking diseases under the dQMR-DT. The QMR-DT and dQMR-DT disease posterior marginals were calculated using the posteriors implied by the appropriate scored diagnoses pools. These posterior marginals are approximations to the true posterior marginals.

on observation process H where the median of *Basic 20* is less than 0.5. The second point is that the QMR-DT must be used carefully with non-ignorable observation processes. For some processes, e.g. H and I, the QMR 20 is clearly better than the Basic 20 and for some processes, e.g. J and K, the QMR 20 is much worse. The change in efficacy of the QMR 20 is most striking in figure 5-4, where the proportion of visible positive manifestations increases significantly between observation processes H and K and the QMR 20 goes from being much better than the Basic 20 to being much worse. There is a much smaller effect in figure 5-3 where the proportion of visible positive manifestations stays approximately constant. These results suggest a dependence of the performance of QMR 20 on the average proportion of positive manifestations that are visible.

This dependence can be explained by examining the differences between the score, $P(\boldsymbol{d}, \boldsymbol{e}^*|\boldsymbol{\phi})$, assigned to a diagnosis by the dQMR-DT and the score, $P(\boldsymbol{d}, F^+, F^-)$[8], assigned by the QMR-DT. Note that because the score of the dQMR-DT can be written

$$P(\boldsymbol{d}, \boldsymbol{e}^*|\boldsymbol{\phi}) = P(\boldsymbol{d}) \prod_i P(e_i^*|\boldsymbol{d}, \boldsymbol{\phi}),$$

the only diagnosis-dependent difference between the QMR-DT and the dQMR-DT scores is the conditional probability assigned to the event that a manifestation is unobserved, i.e. $E_i =?$, given diagnosis $\boldsymbol{d}$. Under the QMR-DT, the probability of this event is independent of the diagnosis being scored. However, under the dQMR-DT, the conditional probability of this event does depend on the diagnosis. Specifically, the dQMR-DT assigns this event conditional probability

$$P(E_i =?|\boldsymbol{d}, \boldsymbol{\phi}) = (1 - \phi_i^+)P(F_i = +|\boldsymbol{d}) + (1 - \phi_i^-)P(F_i = -|\boldsymbol{d}). \qquad (5.9)$$

which can be rewritten as

$$P(E_i =?|\boldsymbol{d}, \boldsymbol{\phi}) = (1 - \phi_i^-) \left\{ \left(\frac{1 - \phi_i^+}{1 - \phi_i^-} \right) P(F_i = +|\boldsymbol{d}) + P(F_i = -|\boldsymbol{d}) \right\}$$

[8] where $\mathcal{I}^+$ $[\mathcal{I}^-]$ contains the indices of the positive [negative] manifestations in $\boldsymbol{e}^*$

When $(1-\phi_i^+)/(1-\phi_i^-) = 1$ then equation (5.9) is independent of the diagnosis, as it is in QMR-DT score. However, under realistic values of $\boldsymbol{\phi}$, $1-\phi_i^-$ is almost always very close to one. The accuracy of the QMR-DT score therefore degrades as the average probability that a positive manifestations will be observed increases, i.e. $1-\phi_i^+$ decreases from one to zero.

Figures 5-5 and 5-6 investigate the effects that this degradation has upon the QMR-DT's scoring of the diagnoses pool. One effect is that the QMR-DT's inaccurate scoring leads to an increase the predicted number of active diseases. As shown in figure 5-5, as the average probability that positive manifestations will be observed increases, the expected number of diseases under the posterior implied by the QMR-DT's scoring increases rapidly. Compare this increase with the slower increase of the expected number under the dQMR-DT's posterior. This overprediction of the number of active diseases is also observed in when the QMR-DT is used for diagnosis in realistic cases (see section 5.2.3 for details). The inaccurate scoring also leads to misranking of the disease marginal probabilities. Figure 5-6 shows this effect. Note that for observation processes where fewer positive manifestations are revealed, the misranking isn't as severe, e.g. for observation process H, finding the 10 top ranked diseases under the dQMR-DT posterior marginals requires examining, on average, the 30 top ranked diseases under the QMR-DT. However, for observation processes where most of the manifestations are revealed, e.g. K, finding the top 10 diseases requires examining a list of at least the 100 top ranked diseases under the QMR-DT.

5.5.4 Discussion

This section investigated two ways to use an observation process model to improve domain-specific probabilistic inference. One use of the observation process model is to score the disease configurations on an algorithm's basic diagnoses list. This technique significantly improves the posterior mass ratios of all inference algorithms. Scoring may even, in some cases, make up for incorrect assumptions about the observation process, as is seen with inference algorithms designed for ignorable observation processes. Another use of the model is to generate samples that can be used to train a recognition model on the complete domain model (i.e. the QMR-DT augmented with the known observation process). This recognition model not only does well when diagnoses may be scored but also puts diagnoses with high posterior probability near the top of its basic diagnoses list, making it appropriate to use these recognition models when scoring is intractable.

This section has also investigated the effect of ignoring a known observation process model. This effect was evaluated under eight different observation process models, each with a similiar bias towards revealing positive manifestations as that observed in the real diagnostic procedure. Under each of the eight models, ignoring the OP model led to inaccuracies in the disease posterior. The severity of the inaccuracy depended upon the probability of observing a given positive manifestation under the model. When the probability was low, i.e. few of the possible positive findings were made, scoring an LR network's diagnoses list using the QMR-DT improved the quality of diagnostic inference. Specifically, the 20 diagnoses with the highest score under the QMR-DT had had higher posterior mass than the 20 diagnoses assigned the highest probability under the recognition model. However, as the probability of observing a positive manifestation increased, the quality of the diagnosis list scored using the QMR-DT decreased.

The major effect of ignoring the observation processes was an overprediction of the number of active diseases present in the patient. This overprediction became more pronounced as a higher proportion of the possible positive findings were made. There was also a reordering of the disease posterior marginals under the posterior implied by the diagnosis list scored using QMR-DT versus that of the list scored using the QMR-DT augmented with the correct observation process model.

In summary, recognition models can be used for effective approximate inference when the observation process is known. If the observation process also supports tractable scoring, the quality of inference using either a recognition model, or an inference method designed for use with the QMR-DT, is improved. Ignoring a known observation process affects the quality of the approximate inference for some, but not all, observation processes.

5.6 Weakly-specified observation processes

5.6.1 Introduction

The problem addressed in this section is motivated by a practical problem that arises in medical diagnostic inference. In section 5.5, I assumed that a model of the OP acting in the diagnostic procedure was available. However, it may not be feasible to build an observation process model that accurately captures variations in the diagnostic procedure due to differences in patients, doctors, and hospitals.[9] Even if such a model were available, it would likely not support tractable scoring of diagnoses. In this section, I present and evaluate strategies for doing medical diagnostic inference using a weakly-specified observation process. In particular, I assume that a small number of different observation process models are available, each accurate under a limited set of conditions, e.g. for a particular patient population at a given hospital. I also assume that a large number of real diagnostic problems, and corresponding gold standard diagnoses, are available both from the medical literature and patient records. These presolved problems may be used to glean further information about the diagnostic procedure. I will assume, however, that the number of presolved problems available is nowhere near enough to train *de novo* a recognition network of the style of section 5.5.

One strategy for doing inference using these two sources of observation process information is to use the given information to fit a probabilistic model of the observation process. Algorithms for fitting the parameters of a probabilistic model to data are widely studied (see e.g. [26]). However, even if a model of the observation process were learned, one would still need to do inference in the likely intractable QMR-DT augmented with the estimated observation process model.

My approach to this problem is avoid building an observation process model at all and to optimise an inference method directly. Specifically, I train a recognition model

[9] Variations in observation processes should be contrasted with the lack of variation in the QMR-DT. I am assuming that medical knowledge stays relatively fixed and that only the observation process changes

on the presolved inference problems to give a composite prediction that combines the predictions of a fixed library of basic inference algorithms. This recognition model is called a *gating network*. Each of the basic algorithms is itself a recognition model optimised to do inference for one of the prespecified observation processes.

Section 5.6.2 defines the problem addressed in this section in greater detail. Section 5.6.3 describes how the parameters of the gating network were optimised. Section 5.6.4 motivates the gating network approach with some intuition about forms of $P(\boldsymbol{\Phi})$ amenable to approximation and section 5.6.5 describes the experiments done to test the approach. Section 5.6.6 summarises and discusses the results of the section.

5.6.2 Problem Definition

Here, I provide further details of the problem solved by the gating network approach. Specifically, I precisely describe the observation process information available to the inference algorithms.

I use the dQMR-DT framework to represent the unknown observation process. I assume that $P(\boldsymbol{\Phi})$ is unknown but that all of the other distributions in the dQMR-DT are provided. The only information that the inference algorithm has about $P(\boldsymbol{\Phi})$ is a a set of assignments of Φ, $\{\phi^m\}$, such that $P(\phi^m) > 0$, and a large set of samples, $\{(\boldsymbol{d}^{(n)}, \boldsymbol{e}^{(n)}\}$, from

$$P(\boldsymbol{d}, \boldsymbol{e}) = \int_{\boldsymbol{\phi}} dP(\boldsymbol{\phi}) \sum_{\boldsymbol{f}} P(\boldsymbol{d}) P(\boldsymbol{f}|\boldsymbol{d}) P(\boldsymbol{e}|\boldsymbol{f}, \boldsymbol{\phi}).$$

Exact inference of the disease posteriors given the evidence vector would require integrating over Φ, i.e.

$$P(\boldsymbol{d}|\boldsymbol{e}) = \int_{\boldsymbol{\phi}} dP(\boldsymbol{\phi}|\boldsymbol{e}) P(\boldsymbol{d}|\boldsymbol{e}, \boldsymbol{\phi}),$$

so an inference algorithm need extract some information about $P(\boldsymbol{\Phi})$ from the given dataset. Learning $P(\boldsymbol{\Phi})$ could be quite difficult due to the high dimensionality of $\boldsymbol{\Phi}$. Section 5.6.4 gives some intuition as to when and how the gating network approach may be a successful strategy.

Remember that the gating network has no direct access to the distribution $P(\boldsymbol{\Phi})$ so inferring $\boldsymbol{\Phi}$ is impossible without fitting a statistical model to $P(\boldsymbol{\Phi})$.

5.6.3 Training

I use a gating network to combine the predictions of the fixed set of basic inference algorithms. Members of this library of algorithms will be called experts. I use LR networks as the individual experts, however the methods described herein will work for any deterministic inference algorithm that is insensitive to small changes in the assumed value of $\boldsymbol{\phi}$.

I use a mixture-of-experts architecture to combine each expert's predictions. Recognition models based on this architecture are described in section 4.3.2. For convenience, I reproduce some of that presentation here. The distributions parameterised by the mixtures-of-experts recognition model are mixtures of factorial distributions:

$$Q(\boldsymbol{d}; \{\boldsymbol{z}^m\}, \boldsymbol{\pi}) = \sum_m \pi_m \prod_k (z_k^m)^{d_k}(1 - z_k^m)^{(1-d_k)}. \tag{5.10}$$

where the parameters $\{\boldsymbol{z}^m\}$ and $\boldsymbol{\pi}$ are the outputs of a set of LR networks and a gating network respectively.

Though the standard mixture-of-experts learning algorithm optimises the individual experts as well as the gating network, because of the difficulty of training the individual LR networks and the relative paucity of presolved problems, the LR network parameters are fixed at their pre-trained values. I use both the input-dependent mixture-of-experts model and a set of fixed mixing weights.

I set the data-determined parameters of the gating network by maximising the likelihood function,

$$E^g(\Sigma) = \sum_n \log Q(\boldsymbol{d}^{(n)}; \{\boldsymbol{z}^m(\boldsymbol{e}^{(n)})\}, \boldsymbol{\pi}(\boldsymbol{e}^{(n)}))$$

on a training set of presolved inference problems, where $\boldsymbol{d}^{(n)}$ is the solution given for evidence vector $\boldsymbol{e}^{(n)}$. Here $\boldsymbol{e}^{(n)}$ is the patient profile and $\boldsymbol{d}^{(n)}$ is the corresponding gold standard diagnosis. For input-dependent gating networks, $\Sigma = \{\boldsymbol{t}_m\}$, for fixed networks, $\Sigma = \boldsymbol{t}$. Note that for the fixed networks, the parameters $\boldsymbol{t}$, optimised during training, specify fixed mixing weights for the experts. Note that for both types of networks, the likelihood function is non-linear and unimodal.

I use the Polak-Ribiere version of conjugate gradient descent (see [51] or [3] for details) to do both maximisations. The line minimisations were done using iterated quadratic and cubic polynomial interpolation.[10]

Since the dataset of presolved problems is limited, one need be careful to avoid overfitting the parameters of the gating network. I avoid overfitting by using a validation set held out from the training data to choose among parameter settings of differing complexity. One may also consider using Bayesian approaches to avoid overfitting (see e.g. [66]).

5.6.4 Intuition

The gating network approach is motivated by the assumption that despite $\boldsymbol{\Phi}$'s high dimensionality, most of the density of $P(\Phi)$ lies in a small region of space close to at least one of the provided values of $\boldsymbol{\phi}^m$. For example, most of the density of $P(\Phi)$ could lie near a low-dimensional manifold or in a small number of tightly clustered clouds. If this assumption holds, then gated predictions from a small, well-chosen, set of experts may be a feasible solution to the problem. For example, a small library of LR networks could be used to span a low-dimensional manifold in Φ-space. Assuming that the the LR networks are fairly insensitive to small changes in the assumed value of $\boldsymbol{\phi}$, each network could well approximate $P(\boldsymbol{d}|\boldsymbol{e}, \boldsymbol{\phi})$ for values of $\boldsymbol{\phi}$ in the region around the assumed value. The gating network, optimised on the presolved inference problems, could then perform inference by weighting the experts in proportion to the posterior probability of that $\boldsymbol{e}$ was generated by a value $\boldsymbol{\phi}$ in the region in which the

[10] Note this optimisation algorithm was implemented by Carl Rasmussen and the MATLAB code is available from his website: `http://www.gatsby.ucl.ac.uk/~edward/code/`

expert specialises. Specifically, if we write

$$P(\boldsymbol{d}|\boldsymbol{e}) = \int_{\boldsymbol{\phi}} dP(\boldsymbol{\phi}|\boldsymbol{e})P(\boldsymbol{d}|\boldsymbol{e},\boldsymbol{\phi}), \tag{5.11}$$

we can compare directly with equation (5.10), the approximate posterior generated by the gating network and the library of LR networks. Comparing equation (5.11) and equation (5.10), we see if the assumption about $P(\boldsymbol{\Phi})$ is correct, then $Q(\boldsymbol{d}|\{\boldsymbol{z}^m(\boldsymbol{e})\}, \boldsymbol{\phi}(\boldsymbol{e}))$ would be a good approximation to $P(\boldsymbol{d}|\boldsymbol{e})$ if the gating network outputs, $\pi^m(\boldsymbol{e})$ well approximated the posterior $P(\boldsymbol{\Phi} = \boldsymbol{\phi}^m|\boldsymbol{e})$.

5.6.5 Experiments

As the unknown observation process model, I use the model contained within the dQMR-DT. To define the distribution $P(\boldsymbol{\Phi})$, I use a uniform distribution $P(Q = q, R = r) =$ const over the ranges of Q and R. This distribution induces a uniform distribution on a linear manifold in $\boldsymbol{\phi}$-space.

To test the gating network approach, I use a small number of values of Φ on the linear manifold described in section 5.3.3 as well as a sizable number of samples from $P(\boldsymbol{d}, \boldsymbol{e})$. The set of limited observation processes provided to the inference algorithms are specified by a set of values, $\{\phi^m\}$ from the manifold in $\boldsymbol{\Phi}$-space. These values correspond to the (q, r) values shown in figure 5-8. Each value $\boldsymbol{\phi}^m$, together with the dQMR-DT, implies an observation process $P(\boldsymbol{e}|\boldsymbol{f}, \boldsymbol{\phi}^m)$. The presolved diagnostic problems provided to the inference algorithms are sampled from the dQMR-DT with the density of $P(\boldsymbol{\Phi})$ implied by the uniform density $P(Q, R)$

I use four values $\{\boldsymbol{\phi}^m\}$ and 2×10^5 samples $\{(\boldsymbol{d}^n, \boldsymbol{e}^n)\}$. One LR network is trained to approximate $P(\boldsymbol{d}|\boldsymbol{e}, \boldsymbol{\phi}^m)$ for each of the four provided values. The provided values are those implied by the (q, r) points {C, E, H, J}, described in figure 5-2. I use the LR networks {LR-C, LR-E, LR-H, LR-J}, described in section 5.5.2, that were trained on these processes. The predictions of the four networks are combined together using the gating network. I also used four additional LR networks, {LR-B, LR-D, LR-I, LR-K}, to act as benchmarks. For each value of $\boldsymbol{e}^n$, the diagnostic pool contained the

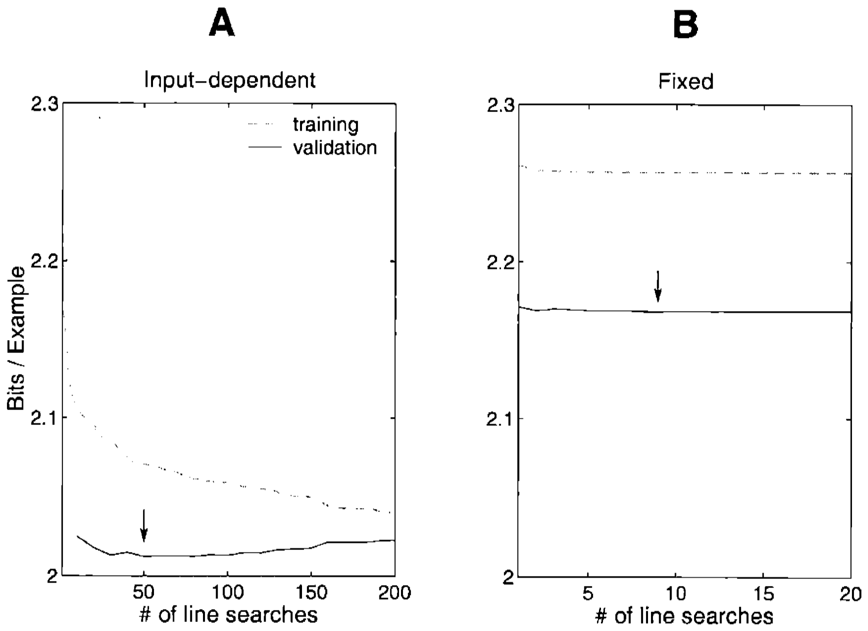


Figure 5-7: Learning curves for gating networks. Each plot shows the curves for the training and validation sets. Since the two sets are disjoint, the average error, measured in cross-entropy bits per example, is different in the two sets. A) The learning curves for the input-dependent gating network. B) The learning curves for the fixed mixture. In both (A) and (B), an arrow marks the final network chosen.

100 diagnoses lists from each of the eight LR networks and the reference diagnosis, $\boldsymbol{d}^n$.

I use the same training set containing 10^5 examples to set the parameters of both the fixed, learned gating network and the input-dependent network. A validation set containing 10^5 examples was used to control parameter complexity. I use an early stopping on the validation set to avoid overfitting the gating networks. I start all parameters at zero, which is equivalent to using equal mixing proportions. After each line search in the conjugate gradient optimisation, I compute the likelihood of the current parameters on a validation set. My final parameters are those that maximise the likelihood on this validation set. Figure 5-7 shows learning curves for the fixed and input-dependent networks.

To evaluate the networks, I used two test sets. The basic test set contains 81000

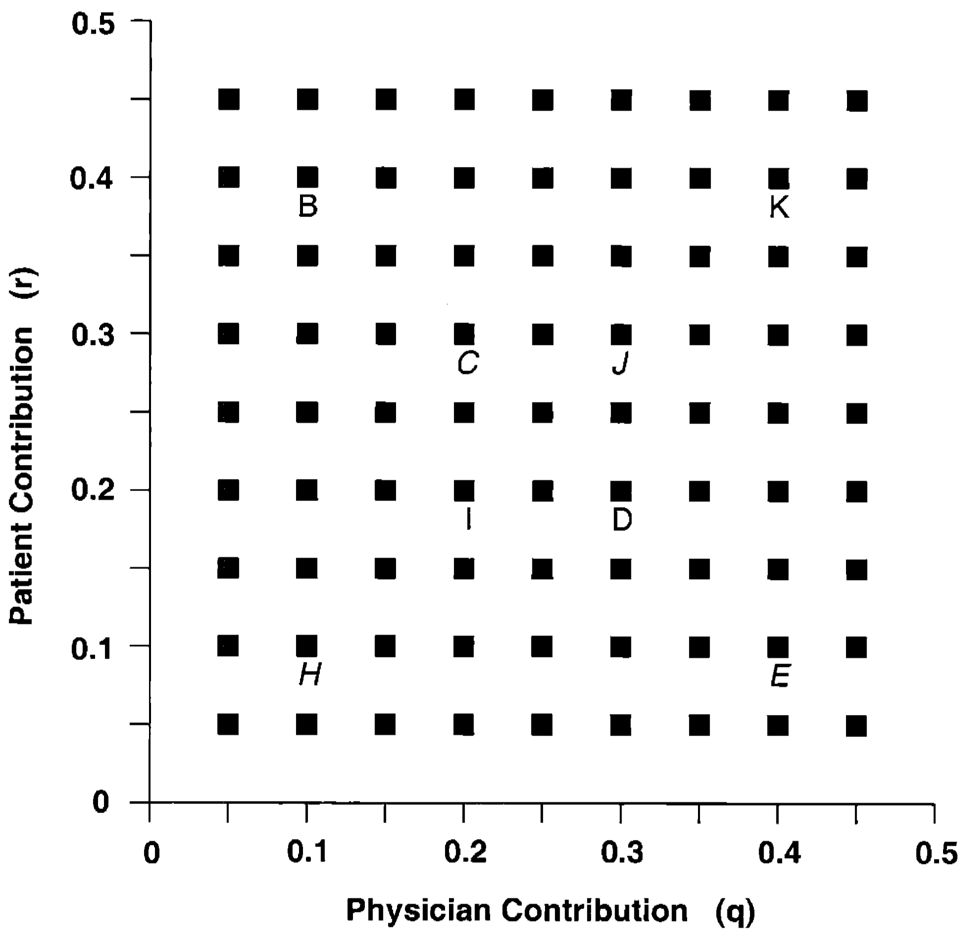


Figure 5-8: Grid points used in test set. Each point in the (q, r)-space maps into a set of dQMR-DT parameters $\phi(q, r)$. The basic test set contains 1,000 samples from $P(\boldsymbol{d}, \boldsymbol{e}|\phi(q, r))$ for each of the 81 grid points, 81,000 $(\boldsymbol{d}, \boldsymbol{e})$ pairs in all. The challenging test set contains 100 samples from $P(\boldsymbol{d}, \boldsymbol{e}|\phi(q, r), \sum_k d_k = 5)$ for each of the 81 grid points. The letters (B-E, H-K) label the locations of the parameter settings for each of the eight LR networks. The points {C, E, H, J} shown in italics correspond to the provided values of ϕ^m and the LR networks whose outputs are combined by the gating network.

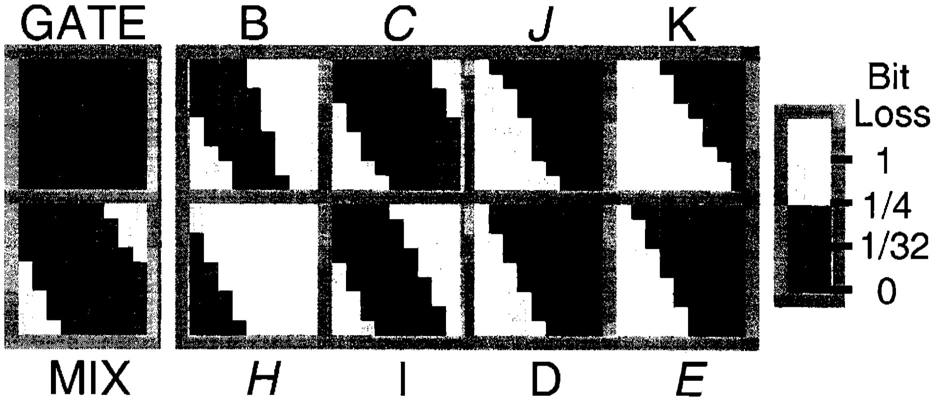


Figure 5-9: Bit loss images. The ten images in this figure show the performance of the gating network (GATE), the learned mixture (MIX) and the individual experts (B-E, H-K). Each image contains 81 pixels, each pixel corresponding to one of the 81 grid points in figure 5-8. The spatial arrangement of the pixels matches that of the grid points, i.e. the pixel in the upper left hand corner of each image shows the performance on samples from the observation process $P(e|\boldsymbol{f}, \phi(0.05, 0.45)$. Each pixel shows the bit loss of the expert (or gating network) on a test set drawn from the corresponding observation process. The labels of the four experts combined by the gating network, $\{C, E, H, J\}$, are shown in italics. A method's bit loss is the difference between its mean cross-entropy on the test set and the cross-entropy of the benchmark method for that test set. The benchmark method for a grid point is the expert with the lowest mean cross-entropy on that grid point's test set.

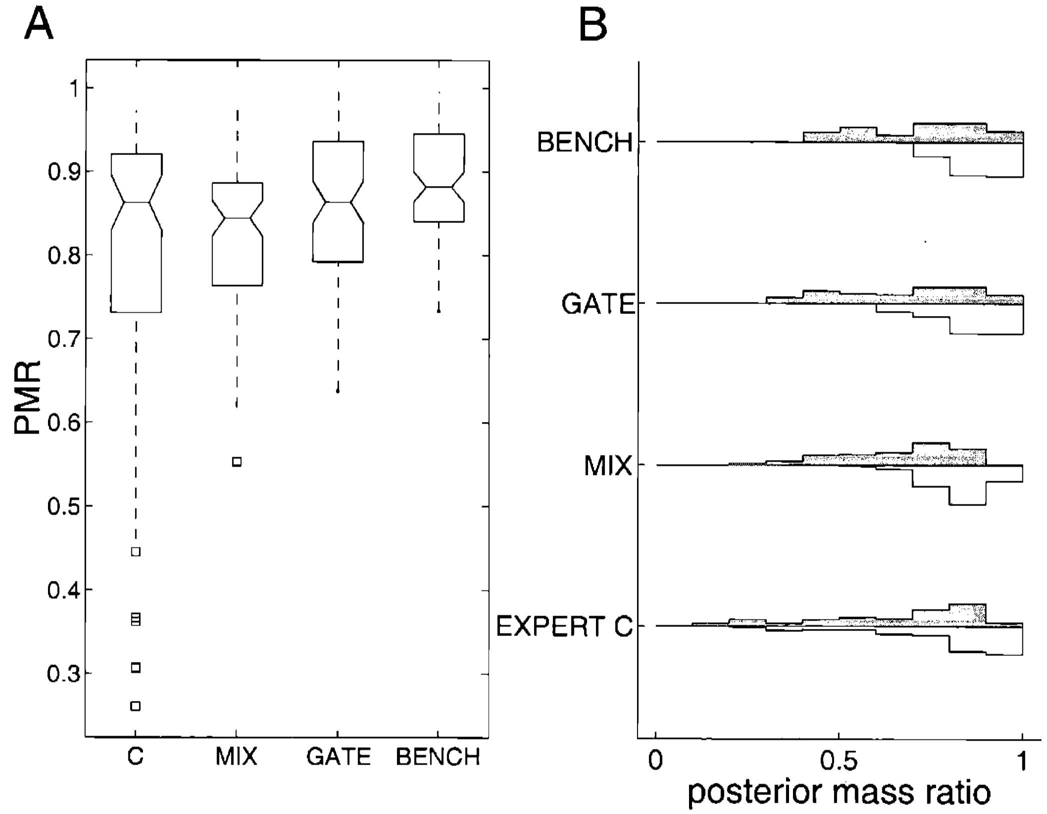


Figure 5-10: Distributions of posterior mass ratios. The plots compare the benchmark to the gating network, the fixed mixture and the individual expert, C, among the four with the best performance. Each plot shows the distribution of the mean posterior mass ratios across the 81 grid points. A) Box plot of the distribution of mean PMRs. The center line in each box is the median of the 81 means. The upper and lower lines show the upper and lower quartiles. The whiskers show the extent of the rest of the data up to a maximum distance away from the median. Points more than 1.5 times the interquartile distance away from the median are displayed with the square symbol. Note that though the median is high for expert C, it has a number of mean PMRs less than 0.5. B) Histogram of mean PMRs. Each row in this plot is composed of two histograms, one pointing upward and the other pointing downward. The upward pointing histogram shows the distribution of posterior mass ratios for the top 20 diagnoses in each list. The downward pointing histogram shows the distribution for the whole list of 100 diagnoses. Each histogram contains ten equally spaced bins.

samples of $(\boldsymbol{d}, \boldsymbol{e})$ comprised of 1000 samples at each of 81 uniformly spaced grid points in (q, r)-space. I also use a challenging test set containing more difficult inference problems. Figure 5-8 shows the grid used and describes the test sets in further detail.

Figure 5-9 shows that there is sufficient information available in the evidence vector to weigh the experts accurately. The bit loss of the gating network is always less than 1/4 of a bit, whereas the fixed mixture's cross-entropy increases at the lower-left (few observed manifestations) and upper-right (many observed manifestations) extremes of the manifold. This shows that the gating network is indeed able to identify samples arising from observation processes in specific regions and reallocate the mixing proportions appropriately. Note also that each of the individual experts have areas where they are making bad predictions, i.e. bit losses of more than one. Mixing the expert's predictions ensures robust diagnosis.

Figure 5-10 shows posterior mass ratio results on the challenging test set. While the aggregate performance of all four are similar in figure 5-10A, the individual expert has very low mean PMR for a number of the grid points, making it unsuitable to use in general. The performance of the gating network is similar to the benchmark, both in terms of the median and the spread of the PMRs. The similarity of the gating network and the benchmark can also be seen by comparing the histograms in figure 5-10B.

5.6.6 Discussion

This section has extended the results of section 5.5 to show how recognition networks could be used for medical diagnosis under less stringent assumptions about the availability of observation process information. Here, I merely assume the availability of a reasonably sized database of solved diagnostic problems and a few simple, limited observation process models. The solved diagnostic problems may be extracted from hospital records though the simple observation process models will have to be manufactured. I have shown how to use these data to fit a gating network which can extract observation process information from the manifestation vector and use that information to mix the predictions from a library of experts. Each expert in the library specialises in one of the prespecified limited observation processes. These experts can be optimised in parallel, reducing the amount of time it takes to optimise the whole system. The parameters of the gating network are optimised using the database of diagnostic problems. I have shown that this gating network works very well on both a simple and a more challenging test set.

The gating network approach has more general applicability than that presented in this chapter. As we saw in section 5.5, inference methods designed for ignorable observation processes perform well in some cases. Little is lost by including these methods among the individual experts. Furthermore, additional LR network-based experts can be trained to perform well for certain patient populations (i.e. those with higher susceptibility to certain diseases or those with compromised immune systems). Finally, the gating network approach could be a way to combine small amounts of real patient data (used to train the gating networks) with probabilistic patient models (used by inference algorithms to make predictions). In this way, by appropriate input-dependent mixing of the predictions of the inference algorithm, the patient data could be used to clean up inaccuracies resulting from inaccurate patient models.

The gating network approach is also very flexible to changes in the observation process. These changes could perhaps arise from changing diagnostic protocols. The changes can be readily accomodated by training a new gating network on a new set of presolved problems that reflect the new observation process. Training the new gating network is much less time consuming than retraining the library of LR networks.

In conclusion, recognition model-based gating networks are excellent methods of using small datasets of high quality data to overcome inaccuracies in recognition models trained on vast quantities of low quality data.

5.7 Discussion

This chapter has addressed the problem of using recognition models, along with the QMR-DT, to do provide diagnostic support for a physician. I have shown that this problem is much more difficult than it *prima facie* appears to be. The additional difficulty is due to the presence of a non-ignorable observation process that I have shown exists in the medical diagnostic procedure. I introduced the diagnostic QMR-DT (dQMR-DT) which contains a modelling framework for this observation process. Note that inference in the dQMR-DT is much more difficult than in the QMR-DT since unobserved manifestations can no longer be removed from the graph without affecting the posterior. One assumes that this is also true in the real observation process implied by the diagnostic procedure.

I have demonstrated that LR-based recognition models can be trained to do inference on the dQMR-DT containing a simple observation process model. In this simple model, the CPDs over the evidence nodes do not change from test case to test case. These simple recognition models can be combined into a composite recognition that is robust to changes in the evidence node CPDs. The combination mechanism is a gating network, itself a type of recognition model, trained on presolved diagnostic problems. I have argued that the limited observation process information (i.e. a collection of simple observation processes and a few presolved problems) used to build this composite may realistically be available whereas a full model of the medical observation process may not.

I have also used the dQMR-DT to underline the importance of doing inference with the correct observation process model. Assuming an ignorable observation process model, i.e. by scoring diagnoses using the QMR-DT, leads in some cases to overpredicting the number of active diseases in the patient and in all cases to a reordering of the diseases predicted to have highest posterior probability of being present in the patient.

I have shown a major advantage to having an observation process model that supports the tractable scoring of diagnoses. Namely, an inference algorithm's diagnoses list can be scored under the correct model. This procedure: generating a list using the recognition model posterior and then scoring the list using the dQMR-DT, is similar to a deterministic importance sampling procedure. Specifically, one can view the recognition model as outputing an input-dependent proposal distribution which is then used to generate disease configurations (in this case deterministically) which are scored using the correct probability model. This view suggests using a deterministic importance sampling algorithm to further improve the quality of the diagnoses lists. In this algorithm, new posterior marginals could be calculated using the scored diagnoses list and used to generate a new diagnoses list.

Chapter 6

Discussion

The unifying theme throughout this thesis has been the construction of a system that uses the QMR-DT to perform medical diagnostic inference. The construction and validation of such a diagnostic support system was done in two steps.

The first step, described in chapter 4, was to determine how to train recognition models to perform inference in the basic QMR-DT. As part of this step, in section 4.4, I developed a concise input encoding that reduces the number of parameters in the recognition model, making optimisation of these models much more feasible. In section 4.6.3 I used the basic QMR-DT to show that recognition models can indeed be trained to produce accurate posteriors for a large-scale inference problem. The accuracy of the posteriors was checked by comparing the average cross-entropy of recognition models to a benchmark approximate inference algorithm described in section 4.6.1. I also showed that the accuracy of the LR network-based recognition models was increased by adding a hidden layer. Adding more units to the hidden layer further increased the accuracy (see figure 4-8). An unfortunate property of the recognition models was the lack of generalisation to some untrained observation processes as shown in figures 4-6, 4-7, and 5-9. This deficiency arises out of the objective function, equation (4.3), used to set the model parameters. This function focuses the modelling power of the recognition model on typical training examples. Untrained observation processes produce atypical test cases, and the recognition model's accuracy suffers accordingly. However, section 5.6 describes how to build a composite recognition model that is robust to changes in the assumed observation process (see e.g. figure 5-9). This composite model uses a gating network to combine predictions from multiple recognition models trained on different observation processes.

The second step in the construction involved training recognition models for non-ignorable observation processes. Section 5.2 provided ample evidence that the observation process acting in the medical diagnostic procedure was both non-ignorable and complex. I introduced the diagnostic QMR-DT, in section 5.3, to provide a framework for modelling different types of observation processes. I trained and evaluated recognition models on both simple, fully-specified observation processes (in section 5.5) and complicated, weakly-specified observation processes (in section 5.6). The models were evaluated using diagnoses lists (described in section 5.4.2): sets of disease configurations generated using the posteriors output by the recognition models. These diagnoses lists are natural endpoints of medical inference, being similar to the differential diagnoses lists used by physicians. I showed in both sections 5.5 and 5.6 that the quality of diagnoses lists, measured in the posterior mass ratio (see section 5.4.5), is improved by scoring them (a technique described in section 5.4.3) using the dQMR-DT. Recognition models performed well under both these conditions both with scored diagnoses lists and without.

The recipe then for building a diagnostic support system using the QMR-DT would be to

1. provide the observation process information used in section 5.6, i.e. a small set of realistic but limited observation processes and a sizable set of presolved diagnostic inference problems,

2. train in parallel a set of LR network-based recognition models using the training protocol described in section 4.5.3, one on each of the provided observation processes,

3. train in parallel a set of MLP-based recognition models, one on each of the provided observation processes, using the LR network weights as short-cut weights

(as described in section 4.5.4) and using as many hidden units as your computing resources permit,

4. combine the predictions of the MLP models using a gating network trained, as described in section 5.6.3, on the presolved inference problems.

The output of this diagnostic support system should be a diagnoses list (described in section 5.4.2) generated from the mixture of factorial distributions.

Further contributions of this thesis include: describing (in section 3.6) and evaluating (in section 3.7) the structural Quickscore algorithm for faster exact inference in the QMR-DT than Quickscore, and characterising (in section 5.5) the danger of ignoring a non-ignorable observation process with similar properties as the real diagnostic procedure.

Note, however, that the contributions of this thesis, like the concise input encoding or the medically relevant observation processes, are not specific to the QMR-DT. There are other large-scale medical expert systems for diagnosis in internal medicine that either incorporate BN2O networks [27] or use a knowledge base which can be easily converted into a BN2O network [2]. The contributions of this thesis can easily be extended to these other networks. These other networks also likely suffer from the same problems with non-ignorable observation processes.

There are a number of directions that the work in this thesis can be extended. Here I describe extensions to both the work on recognition model-based inference methods and on observation processes for medical diagnoses.

In section 4.5.4, I argued that the short-cut weights of the MLP should be initialised using a previously optimised LR network. The natural extension of this approach would be to initialise some of the hidden unit weights in larger MLPs with weights from the units in smaller MLPs. For example, the MLP1000 could be constructed by adding 900 new hidden units to the MLP100. This extension suggests one promising direction of further work: methods for improving the recognition models once the diagnostic support system has been deployed. One may consider ways of adding parameters, and thus modelling power, to the recognition model besides increasing the number of hidden units. For example, one may consider building a composite recognition model, using the mixture-of-experts architecture, described in section 4.3.2. In this composite model, one of the experts would be the existing recognition model and the others could be completely new models. An important issue here is to ensure that the accuracy of the pre-existing recognition model is not compromised by adding additional parameters.

Another direction of further work is to use recognition models for inference in other domains. Besides expert systems, very large Bayesian networks have also been associated with error-correcting codes. Decoding of certain types of these codes requires probabilistic inference (e.g. [33]). The codes and the Bayesian networks used for decoding are fixed, making it feasible to train a large recognition model for inference.

One should also consider different types of recognition models. For example, radial basis function networks [4] could replace the MLPs as the deformable mapping. In the same way that the concise input encoding suggested natural initial values for the LR network weights, different input encodings may be more appropriate for different types of deformable mappings. One may also be able predict in advance the type of recognition model best suited to a particular Bayesian network. This prediction should be based on properties like the connectivity structure of the Bayesian network or the functional form of its conditional probability distributions.

One open question regarding observation processes for medical diagnosis is the effect of approximating the process with an overly simple model. A full model of the diagnostic procedure would require additional dependencies, not present in the dQMR-DT, between manifestations and evidence nodes and perhaps multiple layers of evidence nodes representing distinct stages in the diagnostic procedure. It is possible, however, that a simple model is a sufficiently accurate approximation to the true model. Indeed figure 5-9 shows that generalisation of the LR networks is almost complete along one direction of the observation process manifold. This issue should be investigated further.

This thesis has introduced a multi-part strategy for designing an inference method for medical diagnosis using the QMR-DT. The first part of the strategy is to build an probabilistic model that represents the observation process embodied by the diagnostic procedure. I have shown that this model is critical for ensuring accurate diagnosis because of the strong bias in the diagnostic procedure toward revealing positive manifestations of disease. Open questions remain as to how precise this model need be. The second part of the strategy is to train a recognition model on the QMR-DT augmented with the observation process model. I have shown that recognition models based on LR networks and MLPs produce accurate posteriors on diagnostic problems sampled from the dQMR-DT. Here open questions remain regarding the optimal design of the recognition model. In summary, the results presented in this thesis should be viewed as a significant step along a long, open road toward a reliable, accurate inference method for medical diagnosis.

Appendix A

Creating the aQMR-DT

A.1 Introduction

This appendix describes how the QMR-DT used in this thesis was derived from the the anonymised QMR knowledge base (aQMR KB) which we were provided with by the University of Pittsburgh through the efforts of Gregory F. Cooper, Randolph A. Miller, and Frances Connell. Shwe et al (1991) [64] give a recipe for building a QMR-DT from a similar knowledge base (the Internist-1), however, the aQMR KB doesn't contain the same ingredients as the Internist-1 knowledge base, necessitating a change in the preparation of the QMR-DT.

Section A.2 describes the contents of the aQMR KB. Section A.3 describes how the Θ parameters were set. This procedure is straightforward. However, because the aQMR KB contains neither disease labels nor the manifestation-specific information used by Shwe et al, the setting of the disease priors $\boldsymbol{p}$ and the leak terms, $\boldsymbol{\theta}_0$, is less straightforward. I describe how the disease priors were chosen in section A.4. In section A.5, I describe now the leak terms were set.

A.2 The Anonymised QMR Knowledge Base

The aQMR-DT KB contains 570 disease profiles. Each profile lists information for every manifestation with which that disease is associated. In total, there is infor-

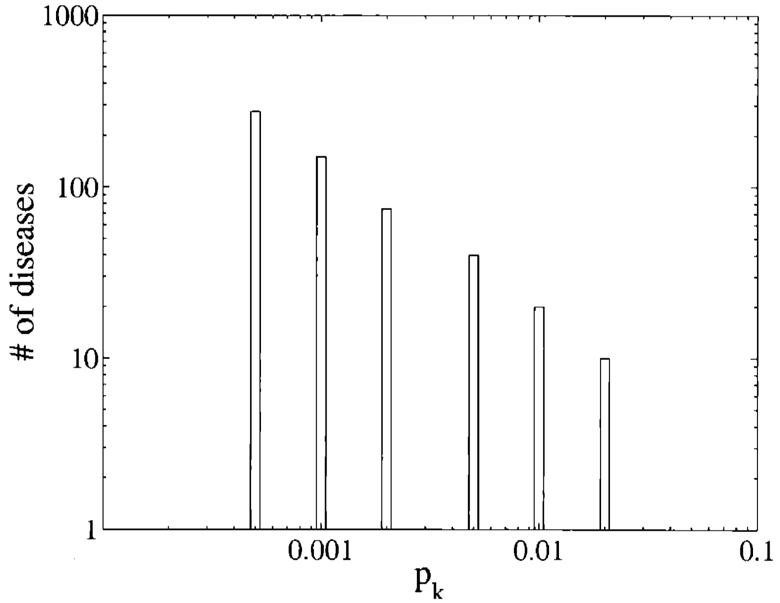


Figure A-1: The distribution of the values used to generate the random priors.

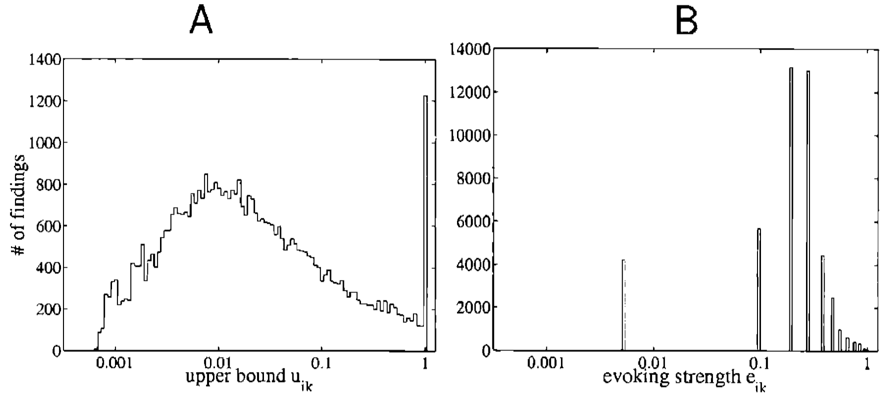


Figure A-2: Evoking strengths and upper bounds. A) The distribution of upper bounds, u_{ik} on estimates of the evoking strengths, $\hat{e}_{ik}$. Notice the spike at $u_{ik} = 1$, these are cases where F_i only has a single parent. B) The distribution of evoking strengths e_{ik} in the aQMR KB.

mation on 45,470 finding/disease pairs. Each listed pair is given a *frequency* value and an *evoking strength.* The frequency, f_{ik}, of manifestation F_i given disease D_k is an estimate of how often F_i is positive when the patient has disease D_i. The evoking strength, e_{ik}, of F_i for D_k is a measure of how strongly a physician should consider a diagnosis containing disease k given the finding that manifestation F_i is positive. Manifestation/disease pairings not listed in the KB are assumed to have zero frequency and zero evoking strength.

A.3 Calculating Θ

We follow [64] in assuming that the frequency value f_{ik} is equal to

$$q_{ik} = P(F_i = +|\text{only } D_k = 1).$$

where $\theta_{ik} = -\log(1 - q_{ik})$. This allows us to map the frequency values f_{ik} directly into the θ_{ik} parameters. I set $q_{ik} = f_{ik}$, i.e. $\theta_{ik} = -\log(1 - f_{ik})$. If no frequency value is given for a disease-manifestation pair, the corresponding θ value is set to zero, i.e. the pair isn't connected in the graph.

A.4 Generating p_k

In Shwe et al [64], the disease priors were derived from hospital discharge statistics and were in the range $[2.0 \times 10^{-5}, 0.02]$. Under priors used in subsequent versions of the QMR-DT, the expected number of active diseases was approximately one.

I generate random, though sensible, disease priors in the hope that realistic priors may eventually be distributed with the aQMR KB. Each prior p_k is selected from a pool of different prior values. The distribution of these values, shown in figure A-1, has a similar form as the distribution of priors used in previous versions of the QMR-DT. This distribution follows a Zipf law with exponentially more disease nodes with a low probability of activation than a high probability of activation. The values in the pool were randomly matched with disease nodes and the expected number of active diseases under this prior is approximately one.

I decided to use random priors rather than deriving them from the evoking strengths since the evoking strengths would imply unrealistically large priors.

A.5 Calculating $\boldsymbol{\theta}_0$

Calculating the leak terms is tricky. Unlike the KB used by [64], there is no manifestation-specific information that may be used to derive the leak terms directly. Instead, I need to derive these terms indirectly using the evoking strengths. Note in the following, I calculate θ_{i0} by first calculating

$$q_{i0} = P(F_i = +|\text{no active diseases})$$

then setting $\theta_{i0} = -\log(1 - q_{i0})$.

Based on suggestions by Dr. Miller, communicated through Greg Cooper, I interpret the evoking strength e_{ik} as an estimate of the posterior marginal probability, $P(d_k = 1 \mid F_i = +)$. Given the disease priors and the Θ parameters, the posterior marginal probability of any disease given $F_i = +$ is determined by the value of q_{i0}. Ideally one should set q_{i0} so that the posterior marginals under the QMR-DT equal the corresponding evoking strengths. However, this is difficult in practice because some of the evoking strengths cannot be achieved by any choice of the value of q_{i0}, as I will show in the following.

For each proposed value of q_{i0}, we can generate the implied posterior marginal, $\hat{e}_{ik}$,

$$\hat{e}_{ik} = P(D_k{=}1 \mid F_i{=}+) = P(D_k{=}1)\frac{P(F_i{=}+ \mid D_k{=}1)}{P(F_i{=}+)}, \tag{A.1}$$

where q_{i0} appears in the expressions for both $P(F_i = +)$ and $P(F_i = + \mid D_k = 1)$. However, since $q_{i0} \in [0, 1]$, $\hat{e}_{ik}$ values are bounded above and below by p_k and u_{ik} respectively, i.e.

$$p_k \leq \hat{e}_{ik} \leq u_{ik}. \tag{A.2}$$

The lower bound is due to the fact that in the noisy-OR function, the presence of a disease cannot decrease the probability that a finding will be positive, i.e.

$$P(F_i = + \mid D_k = 1)/P(F_i = +) \geq 1$$

with equality holding when $q_{i0} = 1$ since in this case, the finding i is always positive. The upper bound u_{ik} comes when $q_{i0} = 0$. This value u_{ik} has the interpretation as the posterior marginal probability that disease k is present given that the positive finding f_i wasn't caused by the leak term.

However, as figures A-2A and A-2B show, most of the upper bounds are quite small compared to the evoking strengths. In fact, in almost 90% of the cases $e_{ik} > u_{ik}$.[1] To ensure that this discrepancy wasn't the fault of our random priors, we computed bounds for four other sets of priors. Each of these sets had the same distribution of p_k values as our original priors. Among the four, there were two sets priors that were generated the same way and those in section A.4, a set of priors where disease nodes with more children had at least as large p_k values, and a set where disease nodes with larger average evoking strengths had at least as large p_k values. In each of the four cases, the proportion of evoking strengths above the upper bound was still almost 90%. The overly large evoking strengths could instead be due to the additive noise in the Anonymised QMR KB values or the fact that the evoking strengths are hard to estimate using the medical literature [13].

Whatever the source, additive noise in the smaller values of the evoking strength is more severe than that in the higher values. As such, we use all and only the achievable evoking strengths to estimate q_{i0}, i.e. we ignore any evoking strengths that lie outside of the bounds.

For each achievable evoking strength, e_{ik}, we can generate an estimate of q_{i0} by finding the value q_{i0} for which $e_{ik} = \hat{e}_{ik}$. If F_i has more than one parent, we may have multiple estimates of q_{i0} for the i-th manifestation node.

In the 25% of cases that there are multiple estimates, I set q_{i0} to be the geometric

[1] And 1% of the time $e_{ik} < p_k$

mean of those estimates. However, for 15% of manifestations there are no achievable evoking strengths, so the corresponding q_{i0} values cannot be assigned by this technique. Instead, in those cases, I randomly choose one of three values for each unassigned q_{i0}. These three values are

1. median of the smallest third of the q_{i0} values (i.e. the 17th percentile),

2. median of the q_{i0} values (i.e. the 50th percentile), and

3. median of the largest third of the q_{i0} values (i.e. the 83rd percentile).

The unassigned q_{i0}'s are distributed equally among the three possibilities. This procedure gives us q_{i0} values in the range $[2.0 \times 10^{-7}, 0.17]$ (compare this to the range $[5.8 \times 10^{-8}, 0.153]$ cited in [64]).

A.6 Conclusions

MATLAB code that transforms the aQMR KB into the QMR-DT described above is available through my website.

Appendix B

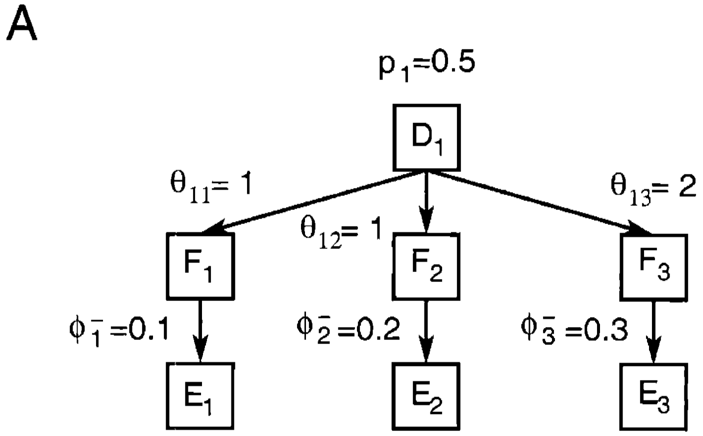
Information Loss in the Concise Encoding

B.1 Introduction

I use the concise input encoding described in section 4.4 for the recognition models in chapter 5. An unfortunate drawback to this encoding is that when the observation process is non-ignorable, this encoding may obscure some information about the evidence vector. In this appendix, I discuss the extent of this information loss. In section B.2, I describe some sets of evidence vectors which have the same concise encoding. Though these vectors are indistinguishable under the concise encoding, they could imply different posteriors. There are, however, additional ways that evidence could be obscured by the concise encoding. In section B.3, I quantify the information loss for a particular observation process by comparing recognition models trained on the concise encoding and a lossless encoding of the input.

B.2 Indistinguishable Evidence Vectors

One form of information loss under the concise encoding is that in some cases, different evidence vectors will have exactly the same encoding. Figure B-1 shows a simple example of two indistinguishable evidence vectors. An easy way to identify a subset of



B

	Encoding	
Finding Vector	**Non-sparse units**	**Sparse units**
? ? -	-2	0 0 0
? - ?	-1	0 0 0
- ? ?	-1	0 0 0

Figure B-1: Indistinguishable observations under the concise encoding. A) A simple BN2O network. B) Concise encodings of three different evidence vectors. See section 4.4 for a description of the two types of inputs. Note that the final two evidence vectors are indistinguishable under the concise encoding. However, since $\phi_1^- \neq \phi_2^-$, disease posteriors given each of the vectors are different.

the evidence vectors that will have the same concise encoding is to find manifestations with the same conditional probability distribution (CPD). Specifically, if two evidence nodes, E_i and E_j, are non-positive and their corresponding manifestation nodes F_i and F_j have the same CPD, then their values can be exchanged in an evidence vector without changing the concise encoding of that vector. Note also that if F_i and F_j are single parent manifestations then E_i and E_j can be exchanged no matter what their state is. Fortunately, there are few exchangeable nodes: only 182 single parent, and 17 two parent manifestations that have the same CPD as another manifestation. No manifestations with more than two parents have the same CPDs.

Manifestations that share the same CPD are problematic because these manifestations, e.g. F_i and F_j, won't necessarily have corresponding evidence nodes, E_i and E_j, with the same CPD. When the evidence node CPDs differ then there will be evidence vectors with the same concise encoding that nonetheless imply a different posterior distribution over the disease nodes.

B.3 Evaluating Information Loss

In this section, I compare a recognition model trained with the concise encoding of the evidence vector to one trained with the lossless encoding. The lossless encoding is described in section 4.4. I used an observation process that contains very few visible manifestations to test the information loss from using the concise encoding. Specifically, I used the observation process $P(\boldsymbol{e}|\boldsymbol{f}, \boldsymbol{\phi}(0.2, 0.2))$ defined by the diagnostic QMR-DT (see section 5.3). Under this observation process, there are very few visible manifestations compared, i.e. on average only 30% of the positive manifestations and 0.4% of the negative manifestations are visible.

I used the same training protocol, detailed in section 4.5.3, to train both the concise and lossless recognition models. Like the concise model, the lossless model has a bank of sparse and non-sparse input units. The sparse inputs for the lossless model consist of both the positive and negative units. The non-sparse input is the single bias unit. The meta gain parameters and initial gain parameters for both

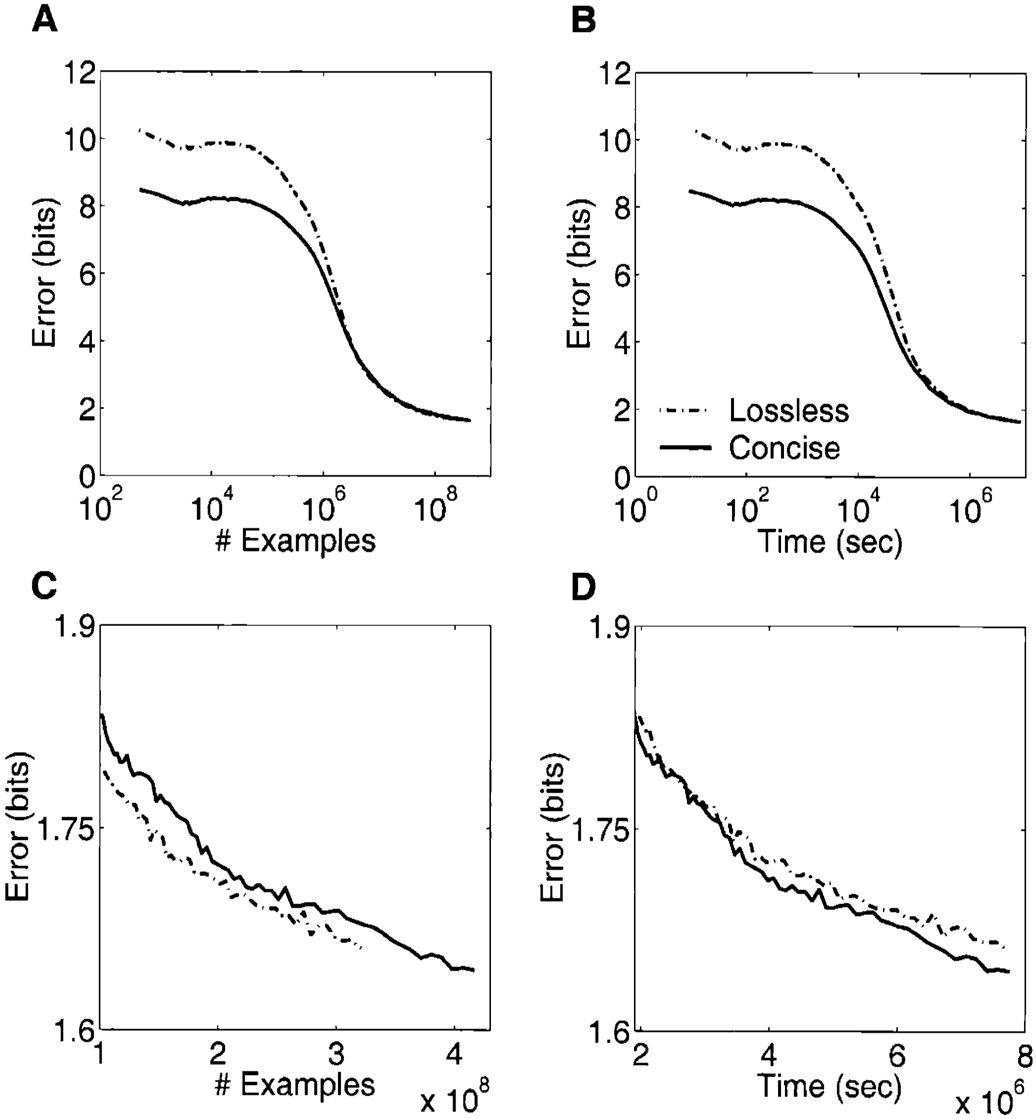


Figure B-2: Comparison of the lossless versus the concise input encoding for recognition models. Figures compare learning curves from the training of the lossless versus the concise recognition models on a dQMR-DT where $P(\Phi = \phi') = \delta(\phi' - \phi(0.2, 0.2))$. Curves are exponential traces of an estimate of the cross-entropy error for each model on untrained data. A) Cross-entropy versus number of training examples. B) Cross-entropy versus CPU time. C; D) Close-up of the tails of A and B respectively. The x-axis is scaled differently for A; B versus C; D.

models were $(\mu_n, \mu_s) = (0.01, 0.3)$ and $(P_n^{\text{init}}, P_s^{\text{init}}) = (10^{-4}, 0.1)$ respectively, and I used a mini-batch size of 500. I initialise all of the weights from the sparse inputs in the lossless model to 0. The weight from the bias unit to output unit z_k is initialised to the log prior odds of D_k, i.e. $\log\{p_k/(1-p_k)\}$ where $p_k = P(D_k = 1)$.

The training sets for the lossless and the concise model were similar. Every mini-batch used to train the lossless model was also used in training the concise model. However, the concise model was trained on a larger number of mini-batches. Specifically, the concise model was trained on 833800 mini-batches (for a total of 4.2×10^9 training examples) and the lossless model used 641000 mini-batches (3.2×10^9 training examples in all).

Despite the larger number of mini-batches for the concise model, both models were trained for approximately the same amount of CPU time. The CPU times per mini-batch for the lossless and concise models were 9.1 seconds and 7.2 seconds respectively[1]. The interpolated total training times for the lossless and concise LR networks compared are 67.7 CPU days and 69.7 CPU days respectively.

Figure B-2 compares the cross-entropy error of the two models. This figure shows that the concise encoding has an initial advantage because of the different initialisations. However toward the end of training, the lossless model has smaller error than the concise model for the same number of training examples (see figure B-2C). Nonetheless, because each mini-batch takes longer to process for the lossless model, the concise encoding has smaller error for the same amount of training time (see figure B-2D).

In summary, the lossless encoding does appear to take better advantage of each mini-batch in the training set, however, this advantage comes at the cost of increased amounts of CPU time. Note that because the models have yet to converge to their respective minimum errors, it is unclear if we should expect these minima to be much different.

[1] CPU times are reported on a XXX Pentitum with a 1 GB L2 cache **(murdock)**

B.4 Discussion

The concise encoding has a number of advantages over the lossless encoding without extensive loss of information. The concise encoding uses many fewer parameters than the lossless encoding and each mini-batch takes much less time to process. The mini-batch processing time of the concise encoding is much less sensitive to the observation process model. Specifically, if the proportion of visible negative manifestations increased from 0.4% to 50%, the mini-batch processing time of the concise encoding would hardly be affected whereas the processing time of the lossless encoding would increase significantly.

Bibliography

[1] *Proceedings of the Ninth International Conference on Artificial Neural Networks*, London, 1999. IEE.

[2] G Octo Barnett, James J Cimino, Jon A Hupp, and Edward P Hoffer. DXplain: An evolving diagnostic decision-support system. *Journal of the American Medical Association*, 258, July 1987.

[3] Christopher M Bishop. *Neural Networks for Pattern Recognition*. Clarendon Press, Oxford, England, 1995.

[4] D S Broomhead and D Lowe. Multivariate functional interpolation and adaptive networks. *Complex Systems*, 2:321–355, 1988.

[5] Jian Cheng and Marek J Druzdzel. AIS–BN: An adaptive importance sampling algorithm for evidential reasoning in large Bayesian networks. *Journal of Artificial Intelligence Research*, 13:155–188, 2000.

[6] Bruce D'Ambrosio. Symbolic probabilistic inference in large BN2O networks. In *Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence*, pages 128–135, 1994.

[7] Peter Dayan. Recurrent sampling models for the Helmholtz machine. *Neural Computation*, 11:653–677, 1999.

[8] Peter Dayan and Geoffrey E Hinton. Varieties of Helmholtz machines. *Neural Networks*, 9:1385–1403, 1996.

[9] Peter Dayan, Geoffrey E Hinton, Radford M Neal, and Richard S Zemel. The Helmholtz machine. *Neural Computation*, 7:889–904, 1995.

[10] Brendan J Frey, Geoffrey E Hinton, and Peter Dayan. Does the Wake–Sleep algorithm produce good density estimators? In Touretzky et al. [65], pages 661–667.

[11] Brendan J Frey, Relu Patrascu, Tommi S Jaakkola, and Jodi Moran. Sequentially fitting "inclusive" trees for inference in noisy–OR networks. In Leen et al. [31].

[12] R Fung and K C Chang. Weighting and integrating evidence for stochastic simulation in Bayesian networks. In Henrion et al. [16].

[13] N B Giuse, D A Giuse, R A Miller, R A Bankowitz, J E Janosky, F Davidoff, B E Hillner, G Hripcsak, M J Lincoln, B Middleton, and J G Peden Jr. Evaluating consensus among physicians in medical knowledge base construction. *Methods of Information in Medicine*, 32:137–145, 1993.

[14] D Heckerman and J Breese. A new look at causal independence. In *Proceedings of Tenth Conference on Uncertainty in Artificial Intelligence*, pages 122–127, 1995.

[15] David Heckerman. A tractable inference algorithm for diagnosing multiple diseases. In Max Henrion, editor, *Proceedings of the Fifth Workshop on Uncertainty in Artificial Intelligence*, pages 174–181, 1989.

[16] M Henrion, R Shachter, L N Kanal, and J Lemmer, editors. *Proceedings of the Fifth Conference on Uncertainty in Artificial Intelligence.* Elsevier Science, 1990.

[17] Max Henrion. Search–based methods to bound diagnostic probabilities in very large belief nets. In *Proceedings of the Sixth Conference on Uncertainty in Artificial Intelligence*, pages 498–508, 1991.

[18] Geoffery E Hinton, Peter Dayan, Brendan J Frey, and Radford M Neal. The wake–sleep algorithm for unsupervised neural networks. *Science*, 268:1158–1161, 1995.

[19] Geoffery E Hinton, Peter Dayan, Ava To, and Radford M Neal. The Helmholtz machine through time. In F Fogelman-Soulie and R Gallinari, editors, *Proceedings of the Fifth International Conference on Artificial Neural Networks*, pages 1158–1161, 1995.

[20] Geoffrey E Hinton and Zoubin Ghahramani. Generative models for discovering sparse distributed representations. *Philosophical Transactions of the Royal Society B*, 352:1177–1190, 1997.

[21] Tommi S Jaakkola. *Variational Methods for Inference and Estimation in Graphical Models.* PhD thesis, Massachusetts Institute of Technology, 1997.

[22] Tommi S Jaakkola and Michael I Jordan. Variational probabilistic inference and the QMR–DT network. *Journal of Artificial Intelligence Research*, 10:291–322, 1999.

[23] Tommi S Jaakkola, Lawrence K Saul, and Michael I Jordan. Fast learning by bounding likelihoods in sigmoid type belief networks. In Touretzky et al. [65].

[24] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton. Adaptive mixtures of local experts. *Neural Computation*, 3(1):79–87, 1991.

[25] Finn V Jensen. *An Introduction to Bayesian Networks.* Springer, New York, 1996.

[26] Michael I Jordan, editor. *Learning in Graphical Models.* MIT press, 1998.

[27] Bert Kappen. PROMEDAS: a probabilistic decision support system for medical diagnosis. Technical report, SNN Nijmegen, 2002.

[28] J H Kim and J Pearl. A computational model for causal and diagnostic reasoning in inference systems. In *Proceedings of the eighth international joint conference on artificial intelligence*, pages 190–193. American Assocation of Artificial Intelligence, 1983.

[29] A Lapedes and R Farber. How neural nets work. In D Z Anderson, editor, *Neural Information Processing Systems*, pages 442–456. American Institute of Physics, New York, 1988.

[30] S. Lauritzen and D. J. Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems (with discussion). *Journal of the Royal Statistical Society series B*, 50:157–224, 1988.

[31] T K Leen, T G Diettrich, and V Tresp, editors. *Advances in Neural Information Processing Systems*, volume 13. MIT Press, 2001.

[32] Roderick J A Little and Donald B Rubin. *Statistical Analysis with Missing Data.* John Wiley and Sons, 1987.

[33] D. J. C. MacKay. Good error correcting codes based on very sparse matrices. *IEEE Transactions on Information Theory*, 45(2):399–431, 1999.

[34] B Middleton, M A Shwe, David E Heckerman, Max Henrion, Eric J Horvitz, H P Lehmann, and G F Cooper. Probabilistic diagnosis using a reformulation of the INTERNIST–1/QMR knowledge base: II. Evaluation of diagnostic performance. *Methods of Information in Medicine*, 30:256–267, 1991.

[35] R A Miller. Medical diagnostic decision support systems – past, present, and future. *Journal of the American Medical Informatics Association*, 1:8–27, 1994.

[36] R A Miller, F E Masarie, and J D Myers. Quick medical reference for diagnostic assistance. *Medical Computing*, 3:34–48, 1986.

[37] R A Miller, H E Pople, and J D Myers. INTERNIST–1: an experimental computer–based diagnostic consultant for general internal medicine. *New England Journal of Medicine*, 307:468–476, 1982.

[38] Thomas P. Minka. *Expectation Propagation for approximate Bayesian inference.* PhD thesis, Massachusetts Institute of Technology, January 2001.

[39] Quaid Morris. Recognition networks for approximate inference in BN2O networks. In *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence*, 2001.

[40] Kevin P Murphy, Yair Weiss, and Michael I Jordan. Loopy belief propagation for approximate inference: An empirical study. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, 1999.

[41] Radford M. Neal. Connectionist learning of belief networks. *Artificial Intelligence*, 56(1):71–113, 1992.

[42] Radford M Neal and Peter Dayan. Factor analysis using delta-rule wake-sleep learning. *Neural Computation*, 9:1781–1803, 1997.

[43] Andrew Y Ng and Michael I Jordan. Approximate inference algorithms for two–layer Bayesian networks. In *Advances in Neural Information Processing Systems*, volume 12. MIT Press, 2000.

[44] D Nilsson. An efficient algorithm for finding the M most probable configurations in probabilistic expert systems. *Statistics and Computing*, 8(2):159–173, 1998.

[45] B A Olshausen and D J Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381:607–609, 1996.

[46] Genevieve B Orr and Klaus-Robert Müller, editors. *Neural Networks: Tricks of the Trade*, volume 1524 of *Lecture Notes in Computer Science*. Springer Verlag, Berlin, 1998.

[47] R C Parker and R A Miller. Using causal knowledge to create simulated patient cases: the CPCS project as an extension of INTERNIST–1. In *Proccedings of the Eleventh Annual Symposium on Computer Apllications in Medical Care*, pages 473–480. IEEE Comp Soc Press, 1987.

[48] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann, San Francisco, California, 1988.

[49] Barak A Pearlmutter. Fast exact multiplication by the hessian. *Neural Computation*, 6:147–160, 1994.

[50] Malcolm Pradhan, Gregory Provan, Blackford Middleton, and Max Henrion. Knowledge engineering for large belief networks. In *Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence*, pages 484–490, 1994.

[51] William H Press, Saul A Teukolsky, William T Vetterling, and Brian P Flannery. *Numerical Recipes in C.* Cambridge University Press, Cambridge, England, 1992.

[52] Irina Rish. *Efficient Reasoning in Graphical Models.* PhD thesis, University of California, Irvine, 1999.

[53] Donald B Rubin. Inference and missing data. *Biometrika*, 63(3):581–592, 1976.

[54] R Y Rubinstein. *Simulation and the Monte Carlo Method.* Wiley, New York, 1981.

[55] Lawrence K Saul, Tommi S Jaakkola, and Michael I Jordan. Mean field theory for sigmoid belief networks. *Journal of Artificial Intelligence Research*, 4:61–76, 1996.

[56] Lawrence K Saul and Michael I Jordan. Boltzmann chains and Hidden Markov Models. In G Tesauro, D Touretzky, and T Leen, editors, *Advances in Neural Information Processing Systems*, volume 7, pages 435–442. MIT Press, 1995.

[57] Nicol N Schraudolph. Centering neural network gradient factors. In Orr and Mūller [46].

[58] Nicol N Schraudolph. Local gain adaptation in stochastic gradient descent. In *Proceedings of the Ninth International Conference on Artificial Neural Networks* [1].

[59] Nicol N Schraudolph. Slope centering: Making shortcut weights effective. In *Proceedings of the Ninth International Conference on Artificial Neural Networks* [1].

[60] Nicol N Schraudolph. Fast curvature matrix-vector products for second-order gradient descent. *Neural Computation*, 14(7), 2002.

[61] R D Shachter and M Peot. Simulation approaches to general probabilistic inference on belief networks. In Henrion et al. [16].

[62] G Shafer and P Shenoy. Probability propagation. *Annals of Mathematics and Artificial Intelligence*, 2:327–352, 1990.

[63] M Shwe and G Cooper. An empirical analysis of likelihood–weighted simulation on a large, multiply connected medical belief net. *Computers and Biomedical Research*, 24:453–475, 1991.

[64] M A Shwe, B Middleton, David E Heckerman, Max Henrion, Eric J Horvitz, H P Lehmann, and G F Cooper. Probabilistic diagnosis using a reformulation of the INTERNIST–1/QMR knowledge base: I. The probabilistic model and inference algorithms. *Methods of Information in Medicine*, 30:241–255, 1991.

[65] David S Touretzky, Michael C Mozer, and M E Hasselmo, editors. *Advances in Neural Information Processing Systems*, volume 8, Cambridge MA, 1996. MIT Press.

[66] S R Waterhouse, D J C MacKay, and A J Robinson. Bayesian methods for mixtures of experts. In Touretzky et al. [65], pages 351–357.

[67] Yair Weiss. Correctness of local probability propagation in graphical models with loops. *Neural Computation*, 12:1–41, 2000.

[68] Johnathan S Yedidia, William T Freeman, and Yair Weiss. Generalized belief propagation. In Leen et al. [31].

[69] Alan Yuille. CCCP algorithms to minimize the Bethe and Kikuchi free energies: Convergent alternatives to belief propagation (unpublished manuscript), 2001.