Bachelor thesis
Computer Science

Radboud University

Towards Large Diagnostic Bayesian Network Models

Author:
Bas van Zadelhoff
4053125

First supervisor/assessor:
prof.dr. P. Lucas
peterl@cs.ru.nl

August 25, 2013

Abstract

The central question of this thesis is: "how can we make the development of large-scale, medical diagnostic Bayesian networks easier?" So far, small Bayesian networks have been used successfully for part of the task of medical diagnosis. Development and use of a Bayesian network becomes much more difficult when we wish to obtain a diagnostic system that covers the whole of medicine.

This paper offers possible solutions for making the development easier and the computational time needed to determine the correct diagnoses smaller. These solution will affect the accuracy of the diagnostic solutions obtained, and this will also be discussed.

Contents

Chapter 1

Introduction

Bayesian networks have so far been successfully used in the development of diagnostic systems, both medical and industrial, that diagnose a small number of defects and diseases. Examples are: MUNIN, Pathfinder, TREAT. These and related systems are able to diagnose up to a hundred different diseases. However, when we wish to develop a diagnostic system that covers the whole of medicine, several thousands of diseases must be covered. There are two reasons why this is hard:

- algorithms for probabilistic inference with Bayesian networks are NP hard in general, and thus, probabilistic inference with a large Bayesian network is likely intractable.
- the building of a Bayesian network that covers thousands of different diseases is a gigantic task. Only when it is possible to do this in a systematic fashion, using generic principles, this might be feasible.

Today there are already some companies that have developed ibased systems that use large-scale diagnostic Bayesian networks for analyzing the symptoms and diseases of patients. An example of these networks is .

When we are developing a network we must often think about for whom this network is intended. For example, will a given network be used to assist doctors or to inform patients? We should also consider how accurate a network is in drawing conclusions. Requirements with respect to accuracy have implications with respect to the graph structure, and thus also with respect to computational resources required to compute results. By analyzing the properties of a medical diagnostic Bayesian network, we wish to obtain a clearer picture of how more complicated diagnostic systems can be developed.

Research has already been conducted in this direction in the past. In particular, research on the QMR-DT network and the different algorithms that were used in this network can be taken as a starting point. I will use this research in my own research for the analysis of such large Bayesian

networks. Although QMR-DT is an example of a network that was designed from a previous existing system and thus may not give definite anwers on designing large complex diagnostic network models, some of the results that came out of this research are likely to be useful in this broader context.

This thesis identifies some of approaches that might be useful to the computational problems in current medical diagnostic Bayesian networks. Mainly the use of different network structure or the use of a combination of smaller network structures are explored.

This thesis is organized as follows. In Chapter 2, I describe some of the background that is needed to understand the remainder of my thesis. Chapter 3 summarizes related research. In Chapter 4, I discuss possible ways to deal with the problems in developing a Bayesian diagnostic system that covers the whole of medicine. Chapter 5 gives the final conclusions of the research.

Chapter 2

Preliminaries

2.1 Probabilistic graphical models

A *probabilistic graphical model* defines a joint, or multivariate, probability distribution in terms of a graph, directed, undirected or mixed, and a set of functions that together define the distribution. The idea is that the graph denotes the independent and dependent information that holds for the distribution. Probabilistic Graphical models are part of probability theory and they can be used as models in machine learning and statistics. Two types of graphical models are in particular commonly used, namely, Bayesian networks and Markov networks. Both types of probabilistic graphical model support factorization of a probability distribution according to the associated graph; however, they differ in the set of independences they can encode and the factorization of the distribution that they induce [1]. In Bayesian networks, the associated graph is directed, whereas in a Markov network it is undirected.

2.2 Bayesian network

A *Bayesian network* is a probabilistic graphical model that represents a set of random variables and their conditional dependencies via a directed acyclic graph. For example, a Bayesian network could represent the probabilistic relationships between diseases and symptoms. Given symptoms, the network can be used to compute the probabilities of the presence of various diseases [2]. We will briefly explain what a directed acyclic graph (DAG) is and how independences can be derived from graphical information by means of d-separation.

2.2.1 DAG

Directed acyclic graph or DAG, is a directed graph with no directed cycles. This graph is formed by a collection of vertices and directed edges. The vertices are connect by the edges to each other. The connections between the vertices is constructed in a way that there is no way to start at vertex X and end at vertex X by following the edges [3].

Definition 1. *A* directed graph *is defined as a pair $G = (V, E)$ where V is a finite set of variables, called* nodes *or* vertices*, and $E \subseteq V \times V$ is a set of ordered pairs of vertices (X_i, X_j), called* arcs*. For an arc $(X_i, X_j) \in E$, X_i is called the* parent *of X_j, and X_j is called the* child *of X_j.*

Now we can define what it means that a directed graph is acyclic.

Definition 2. *If $G = (V, E)$ is a directed graph, then a* path *is a sequence of nodes $X_1, X_2, \ldots, X_n$, such that $(X_i, X_{i+1}) \in E$ with $1 \leq i \leq n$. G is then an* acyclic *directed graph if it contains no paths $X_1, X_2, \ldots, X_n$, such that $X_1 = X_n$.*

2.2.2 Directed bipartite graph

A bipartite graph (or bigraph) is a graph whose vertices can be divided into two disjoint sets U and V such that every arc connects a vertex in U to one in V. That is, U and V are each independent sets. Equivalently, a bipartite graph is a graph that does not contain any odd-length cycles. Each arc has endpoints of differing colors, as is required in the graph coloring problem. We can see an example of a bipartite graph in figure 2.1.

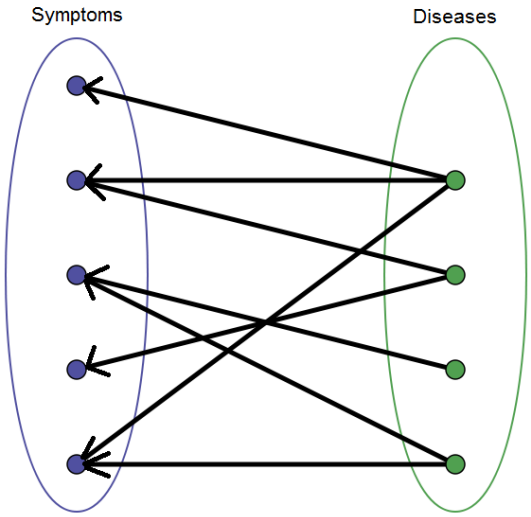

Figure 2.1: Example of a bipartite graph

One often writes $G = (U, V, E)$ to denote a bipartite graph whose partition has the parts U and V with E denoting the arcs of the graph. If a bipartite graph is not connected, it may have more than one bipartition; in this case, the (U, V, E) notation is helpful in specifying one particular bipartition that may be of importance in an application. If |U| = |V|, that is, if the two subsets have equal cardinality, then G is called a balanced bipartite graph. If vertices on the same side of the bipartition have the same degree, then G is called biregular [4].

Bipartite graphs may be characterized in several different ways [4] :

- A graph is bipartite if and only if it does not contain an odd cycle.
- A graph is bipartite if and only if it is 2-colorable.
- The spectrum of a graph is symmetric if and only if it's a bipartite graph

2.2.3 D-separation

In figure 2.2 there are three connections. The first is serial this is a connection with says X_1 causes X_2 and X_2 causes X_3. Knowledge about X_1 will cause a change in our beliefs in X_2 that will cause a change in our beliefs in X_3. Conversely, knowledge about X_3 will also in turn change our beliefs in X_1 and X_2. The serial connection in figure 2.2 says say that X_1 and X_3 are d-separated(directed separation) by X_2. This means if we have hard evidence about X_2 then knowledge about X_1 will not change our beliefs of X_3

because X_2 is already supplying support to X_3. Also knowledge about X_3 can't cause a change in our beliefs of X_1 because X_2 is already confirmed. The d-separated can also be found in diverging and converging.

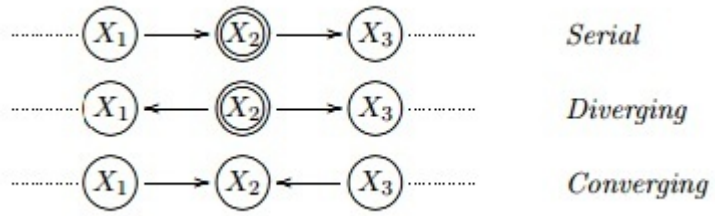

Figure 2.2: Possible connections

Formally we define d-separation as [5]:

Definition 3. *If $G = (V, E)$ is a directed acyclic graph(DAG) and $X, Y, Z \subseteq V$ are three disjoint sets. The set Y d-separates the sets X and Z. If for each V_i, Z_i and every connection $X_i, \ldots, V_i, V_j, V_k, \ldots Z_i$ in G the following conditions hold:*

1. *The connection V_i, V_j, V_k is serial or diverging and V_j is in Y*
2. *The connection V_i, V_j, V_k is converging and neither V_j nor its descendants are in Y*

We can assume that the edges represent causal relations as long as they hold up to the d-separation criterion.

Using the three definitions we can now define a Bayesian network as: [6][7]

Definition 4. *A Bayesian network B over a set of variables X is a Pair $B = (G, P)$ with:*

1. *$G = (V, E)$ is a DAG*
2. *$P = \{P(x_i \mid px_i) \mid X_i \in X\}$ is the set of local subjective probabilities.*

where $PX(i)$ is the set of parents of vertex i.

2.2.4 Chain rule

By using the chain rule and the previous definition of a Bayesian network we can derive the following theorem [8]: Formally, a Bayesian network is a pair $B = (G, P)$, Where $G = (V, E)$ is an acyclic directed graph and P the

associated joint probability distribution that is defined such that there is for each vertex in G a random variable in p, and:

$$P(X_v) = \prod_{v \varepsilon V} P(X_v \mid PX_i)$$

where $PX(i)$ is the set of parents of vertex i.

2.3 Marginal distribution

The marginal distribution of a subset of a collection of random variables is the probability distribution of the variables contained in the subset. It gives the probabilities of various values of the variables in the subset without reference to the values of the other variables. This contrasts with a conditional distribution, which gives the probabilities contingent upon the values of the other variables.

Given two random variables X and Y whose joint distribution is known, the marginal distribution of X is simply the probability distribution of X averaging over information about Y. It is the probability distribution of X when the value of Y is not known. This is typically calculated by summing or integrating the joint probability distribution over Y [9].

For discrete random variables, the marginal probability mass function [10] can be written as $P(X = x)$. This is

$$P(X = x) = \sum_y P(X = x, Y = y) = \sum_y P(X = x \mid Y = y)P(Y = y)$$

where $P(X = x, Y = y)$ is the joint distribution of X and Y, while $P(X = x \mid Y = y)$ is the conditional distribution of X given Y. In this case, the variable Y has been marginalized out.

2.4 Conditional probability distribution

In probability theory and statistics, given two jointly distributed random variables X and Y, the conditional probability distribution of Y given X is the probability distribution of Y when X is known to be a particular value. In some cases the conditional probabilities may be expressed as functions containing the unspecified value x of X as a parameter. The conditional distribution contrasts with the marginal distribution of a random variable, which is its distribution without reference to the value of the other variable [11].

Let P be a probability distribution. The conditional probability distribution of A given $B = b$, denoted by P(A | B = b), is defined by

$$P(A|B = b) = \frac{P(A, B = b)}{P(B = b)},$$

where $P(B = b) > 0$.

We can use the conditional probability distribution to deduce the bayes' rule[12].

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}$$

This rule can be used when we need to follow the arcs backwards in a Bayesian network.

2.5 Causal independence models

Let M be the set of all of the variables in mechanisms for causes C_1, ..., C_n and effect E. As in the case of causal independence models, the independence of the causal mechanisms is captured by

1. The conditional independence of the set of variables in each mechanism given the causes (i.e., for $i \neq j, I_i$ is independent of I_j given C_1, ..., C_n).

2. The independence between the set of all mechanism variables (M) and other variables in the DAG model given the causes C_i and effect E.

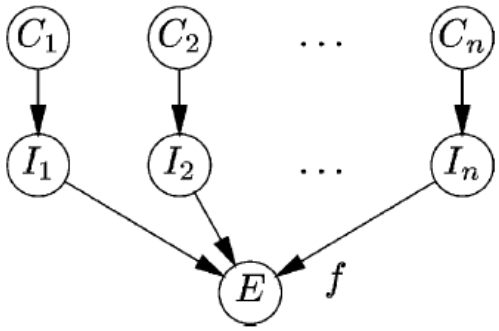

Figure 2.3: Causal independence model

Causal independence, also called noisy functional dependence, is a popular way to specify interactions among cause variables. The global structure of a causal independence model is shown in figure 2.3. It expresses the idea that causes C_1, ..., C_n influence a given common effect E through intermediate variables I_1, ..., I_n and a deterministic function f, called the interaction function. The impact of each cause C_k on the common effect E is independent of each other cause $C_j, j \neq k$. The function f represents in which way the intermediate effects I_K, and indirectly also the causes C_k, interact to yield the final effect E. Hence, the function f is defined in such a way that when a relationship, as modeled by the function f, between $I_k, k = 1, ..., n$,

and $E = T$ is satisfied, then it holds that $e = f(I_1, ..., I_n)$. It is assumed that $P(e \mid I_1, ..., I_n) = 1$ if $f(I_1, ..., I_n) = e$, and $P(e \mid I_1, ..., I_n) = 0$ if $f = (I_1, ..., I_n) = e$.

The conditional probability of the occurrence of the effect E given the causes C_1, ..., C_n, i.e. $P(e \mid C_1, ..., C_n)$, can be obtained from the conditional probabilities $P(I_k \mid C_k)$ as follows:

$$P(e \mid C_1, ..., C_n) = \sum_{f(I_1,...,I_n)=e} \prod_{k=1}^{n} P(I_k \mid C_k)$$

Well-known examples of causal independence models are the noisy-OR and noisy-AND models, where the function f represents a logical OR and a logical AND function [13].

Chapter 3

Related Work

To obtain an impression of the state-of-the-art of building large baysian networks, we consider some of the systems that where developed using in the past.

3.1 Structure of a medical Bayesian network

If we want to develop a complete network for the whole of medicine, we of course need knowledge about diseases and symptoms. The way that will get the best results is by making different nodes for every disease and symptom. Between these diseases and symptoms we have the mechanism that determines the direction and place of the arcs in this network. In medicine this mechanism is the pathophysiology. pathophysiology describes the abnormal or undesired condition, whereupon pathophysiology seeks to explain the physiological processes or mechanisms whereby such condition develops and progresses. Symptoms are not the only factor that can influence the chance of having a disease. For example age, smoking and drugs can change the probability of having a certain disease. These can group as environmental conditions. Figure 3.1 shows the best structure for a medical diagnostic Bayesian network.

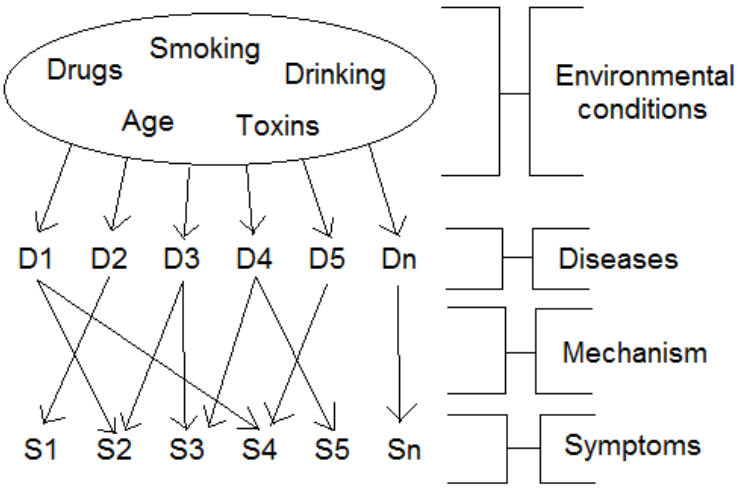

Figure 3.1: Ideal network structure

3.2 Pathfinder

3.2.1 Summary

The first system that will be considered is Pathfinder. Pathfinder is a system that was developed in 1992. The objective of this project was to develop an expert system that reasons efficiently and accurately about lymph-node diseases. Over 60 diseases can invade the lymph node (25 benign diseases, 9 Hodgkin's lymphomas, 18 non-Hodgkin's lymphomas, and 10 metastatic diseases). The computational architecture of the Pathfinder system is based on the method of sequential diagnosis [14]. After a user enters salient features, a list of plausible disease hypotheses, or a differential diagnosis is formulated based on these manifestations. Next, questions are selected that, if answered, can help to reduce the number of diseases under consideration. After the user answers these questions, a new set of hypotheses is formulated and the process is repeated until the user is satisfied that diagnosis is reached [15][16].

There were four versions of Pathfinder. The fourth version is of interest to us, because that is the only version that uses a Bayesian network. The fourth version used about 75,000 parameters, because of similarity networks allows it to be constructed with only 14,000 parameters. A similarity network is a network where there are nodes that can have multiple value that are mutually exclusive. It is also possible that certain value only uses a part of the network, which means you do not need all parameters to build this network. The Bayesian network model agreed with the predictions of

an expert pathologist in 50/53 cases. A later evaluation showed that the diagnostic accuracy of Pathfinder IV was at least as good as that of the expert used to design the system. When used with less expert pathologists, the system significantly improved the diagnostic accuracy of the physicians alone [1][17].

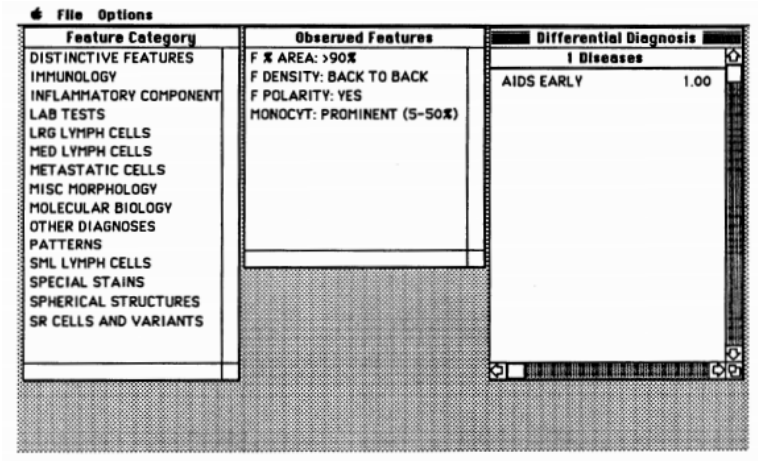

Figure 3.2: Pathfinder determines that only a single disease - AIDS EARLY(the early phase of AIDS) - is consistent with the four observations

3.2.2 Structure

In pathfinder we have some differences to the structure then described in section 3.1: Structure of a medical Bayesian network. The diseases are in one big node and there are no environmental conditions that have an effect on the diseases. The mechanism and symptoms are the same as in the best structure. Of course there are less diseases, symptoms and arcs, because this network does not cover the whole of medicine. In figure 3.3 we can see the structure of the pathfinder network.

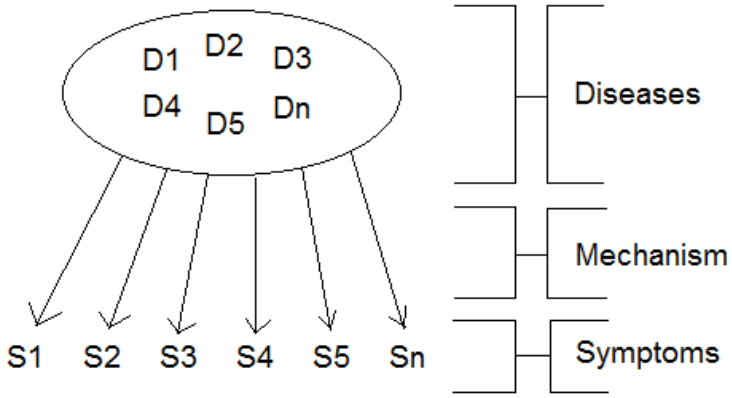

Figure 3.3: Network structure pathfinder

3.2.3 Conclusion

Technical

1. Pathfinder has single node with all the diseases in it. This will make it impossible to get more then one disease as a result.

Non-technical

1. One of the reasons that prevented the spread of Pathfinder as a medical diagnosis aid was the legal liability issues of misdiagnoses.
2. There was also an incompatibility with the physicians workflow.
3. Because it only focuses on one type of disease this will limit the results of the application.

3.3 Internist-1

3.3.1 CPCS-PM

Summary

The second system is the Computer-based Patient Case Simulation probabilistic model (CPCS-PM). This system has 422 nodes and 867 arcs. The CPCS-PM system is a knowledge base and simulation program designed to create patient scenarios in the medical sub-domain of hepatobiliary disease, and then evaluate medical students as they managed the simulated patient's problem. Unlike that of its predecessor lnternist-1, the CPCS-PM knowledge base models the pathophysiology of diseases-the intermediate states causally linked between diseases and manifestations [14]. In figure 3.4 is a

part of the CPCS-PM network.

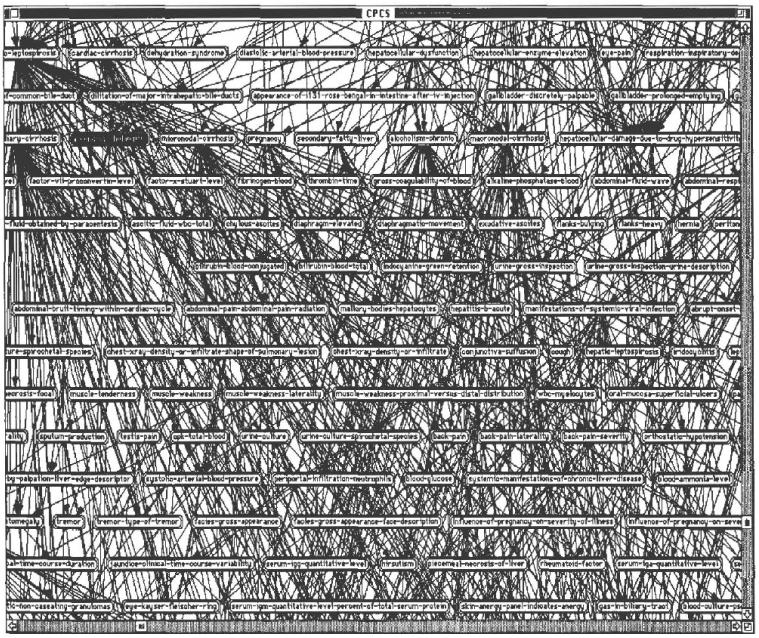

Figure 3.4: A small portion of the CPCS BN displayed in the Netview visualization program.

While the CPCS-PM knowledge base is derived from the Internist-1 knowledge base it has been significantly augmented by inclusion of the intermediate pathophysiological states (IPS) states, and multivalued representations of both diseases and manifestations of disease. The original Quick Medical Reference bayesian network (QMR-BN) transformation of the Internist-1 knowledge base used only binary valued disease and manifestation nodes. While conceptually simple, this approach does not adequately reflect the potential variation in presentation of disease manifestations, or the severity of diseases [14].

The CPCS BN is automatically generated, because CPCS BN was not intended for probabilistic interpretations. For this reason the system has to be manually checked using domain knowledge to edit the network. This means there have to be corrections in the sense of removing nodes, only use mutually exclusive values and making the values of nodes consistent.

Conclusion

Technical

1. The biggest problem of CPCS is that it isn't accurately generated and has to be updated by hand.

Non-technical

1. Because it only focuses on one type of disease this will limit the results of the application.
2. The cost of the application is high because there is always a need for a domain expert.
3. This system is only used to train students; this also says something about the reliability of this application.

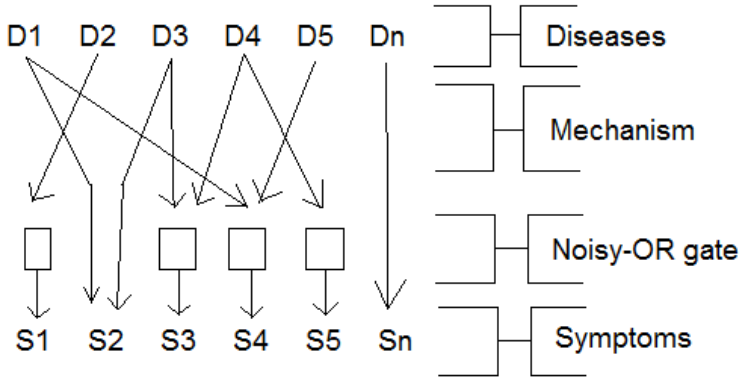

Figure 3.5: Network structure CPCS

3.3.2 QMR-DT

Summary

The third system is the Quick Medical Reference, Decision Theoretic (QMR-DT derived from Internist-1). This system has 534 diseases with 4040 nodes and 40740 arcs. QMR-DT is one of the three levels of QMR. The second is the possibility to show findings associated with diseases and the other way around and the third is the ability to show how particular groups of diseases and findings may co-occur. The QMR-DT model makes five assumptions to improve the representational and computational complexity of this model [18].

The first assumption is marginal independence of diseases; this means that the developers of this model choose not to include the connections between disease. This shows in the model by the lack of arcs between the diseases. This assumption is mostly correct but there will be exceptions, for example the probability of congestive heart failure increases in a patient that has aortic stenosis.

The second assumption is conditional independence of findings this means that there is also no connection between findings. This assumption can be used because we are talking about observed variables. If we observe that a patient has a fever and after that we observe that the patient also has back pain. The second observation has no influence on our first observation [19].

The third assumption is that diseases and findings are represented by binary variables. Simply it's saying a disease is present or it's not present. The consequences of this assumption is that some diseases with different levels of severity have also different outcomes or symptoms. This will result in different probabilities for the subsequent conditions or diseases.

The fourth assumption is causal independence: it concerns the mechanisms by which diseases cause a finding independent of one another and independent of any other events that may cause the finding to occur, such as the influence of other findings. This is done by using a noisy-OR (See section 2.5) gate in the model. The reason for making this assumption is that we reduce the number of conditional probabilities we have to use to calculate the probability of X.

The fifth and last assumption is findings as manifestations of disease. In the QMR-DT model all findings are represented as manifestations of disease. This eliminates the use of historical findings like a history of smoking [20].

Case	# of positive findings	# of negative findings
1	20	14
2	10	21
3	19	19
4	19	33

In panel (a) there were 8 positive findings treated exactly, and in (b) 12 positive findings were treated exactly. As expected, the bounds were tighter when more positive findings were treated exactly. The average running time across the four tractable CPC cases was 26.9 seconds for the exact method, 011 seconds for the variational method with 8 positive findings treated ex-

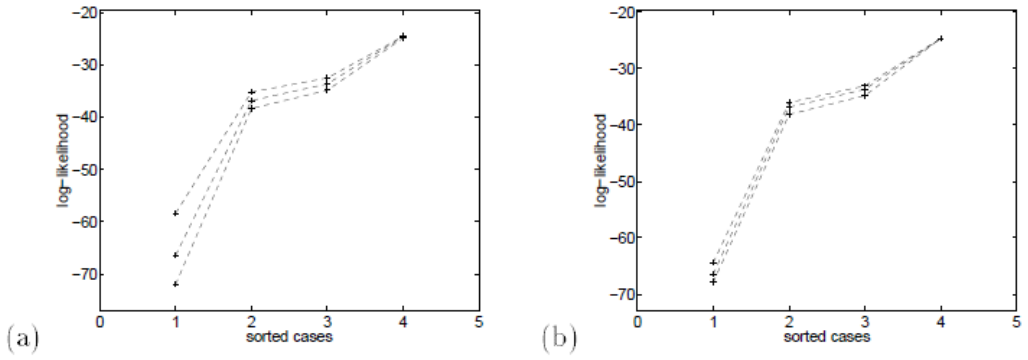

Figure 3.6: Exact values and variational upper and lower bounds on the log-likelihood $log(f^+ \mid \xi)$ for the four tractable CPC cases. In (a) 8 positive findings were treated exactly, and in (b) 12 positive findings were treated exactly.

actly, and 0.85 seconds for the variational method with 12 positive findings treated exactly. (These results were obtained on a 433 MHz DEC Alpha computer) [1] [19].

Conclusion

Technical

1. Because of the first assumption it is possible that a diagnose is less accurate than it can be.
2. The model is simpler because of the second assumption. This makes the system faster but this also makes it harder to find an error if it occurs.
3. As already said in the info on QMR-DT the third assumption can cause different probabilities for the subsequent conditions or diseases.
4. A consequence of the fourth assumption is that it's less accurate in cases where the diseases operate through a common pathway.
5. Historical findings can greatly change the chance of having or getting a certain disease.

Non-technical

[1] "treated exactly" simply means that the findings are not transformed variationally

1. As a medical diagnosis aid we have to deal with the legal liability issues of misdiagnoses.

2. Because focuses on more diseases then more other system, but it's not every diseases so there will be a limited to the results of the application.

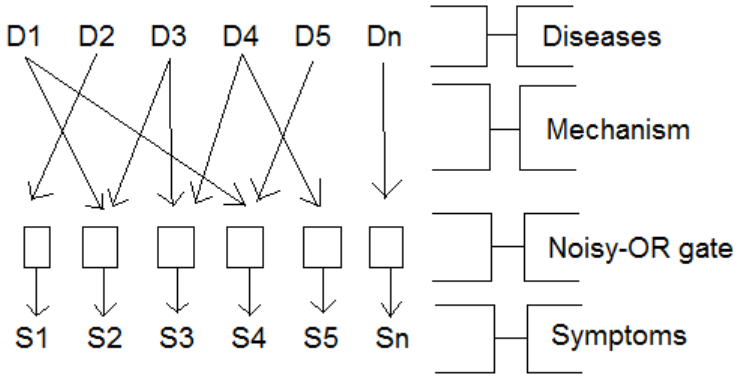

Figure 3.7: Network structure QMR

3.3.3 Structure

The CPCS structure is much like the structure of the medical baysian network in section 3.1. The diseases and symptoms are the same as in that structure, but the differences is that there is no environmental conditions in this network. Also there have been changes between the mechanism and diseases. A noisy-OR gate is sometimes used. In figure 3.5 we can see the structure of the CPCS network. The structure of QMR is almost the same, but with QMR there is always a noisy-OR gate while CPCS only uses it sometimes. In figure 3.7 we can see the structure of the QMR network.

3.3.4 Differences

The CPCS uses less diseases and symptoms then QMR. This will result in less time needed to calculate the diseases, but there are diseases that can be missed. The QMR uses five assumptions and CPCS does not use them all. The most important one is that diseases and findings arerepresented by binary variables in QMR and it is possible to use multivalued representations of both diseases and manifestations of disease. So on this point CPCS is more accurate then QMR. Another big difference is that CPCS is automatically generated and QMR is not.

3.4 MUNIN

3.4.1 Summary

MUscle and Nerve Inference Network (MUNIN) is the fourth network I looked at. In earlier publications it was proposed that a causal network contains the information necessary for a unified approach to three of the main tasks of a medical expert system: diagnosing, planning of data acquisition, and explanation of the systems reasoning. At the beginning of a diagnostic session the disease node is initialized with a priori probabilities corresponding to the observed frequencies of the diseases in patients referred for electromyography (EMG) examinations.

The number of diseases is restricted to three, each with two to four states, corresponding to gradations and/or different varieties of the diseases. In addition the patient may be in one of the states "normal" or other, giving a total of eleven different "disease" states. An algorithm for propagation of evidence in causal networks was developed by Kim and Pearl (1983). The algorithm was adapted to this network and supplemented by a method for coherent initialization of probabilities. Later in the development they switched to the junction-tree algorithm developed by Lauritzen and Spiegelhalter (1988), because this algorithm preformed better.

Beyond the already mentioned reductions on the number of diseases, the prototype is also restricted in other ways: multiple simultaneous diseases are not considered and the network does not handle measurement of nerve signals, which are as important as measurement of muscle signals. Furthermore, it only considers findings from one muscle.

The diagnostic task consists of adjusting the probabilities in all nodes as the findings are entered into the findings nodes. A finding entered into a findings node is indicated by a broken horizontal 100% bar. The network correctly indicates a large probability for moderate to severe axonal neuropathy, it generates distributions for the pathophysiological nodes that are consistent with "moderate chronic axonal neuropathy" and offers predictions of the outcomes of the remaining findings, should the physician chose to perform the appropriate EMG-tests [21][22].

A network for the interpretation of EMG finding has been constructed. We expect a network of this type to be an important building block in an expert system for EMG. Although the network is small and in its current form has only limited functions, it has allowed us to reach a number of conclusions:

1. With present algorithms for propagation of evidence in causal probabilistic networks, probabilistic inference is a feasible approach. Since the computation time of the algorithms is increasing approximately

linearly with the number of states in the network, we expect that probabilistic inference can also be used in networks considerably larger than the current network.

2. The shift from nodes with only two states (yes, no) to nodes with multiple states has given a conceptual simplicity that makes knowledge acquisition and verification easier. It also makes the knowledge representation very compact.

3. The use of "deep knowledge" in the form of models has reduced the almost intractable problem of estimating thousands of probabilities to the much more tractable problem of adjusting a much smaller number of model parameters. The models have the added virtue that they can be explained through pathophysiological reasoning similar to the reasoning done by an expert.

4. Lack of knowledge in the system and conflicting evidence is handled in a simple and consistent way by adding the state "other" to some of the nodes. This way the network can signal, when it reaches the limits of its knowledge.

3.4.2 Structure

The MUNIN structure is different from the structure in section 3.1: Structure of a medical Bayesian network. The diseases, symptoms and mechanism is the same. The difference with this network is that it does not include the environmental conditions. In figure 3.8 we can see the structure of the MUNIN network.

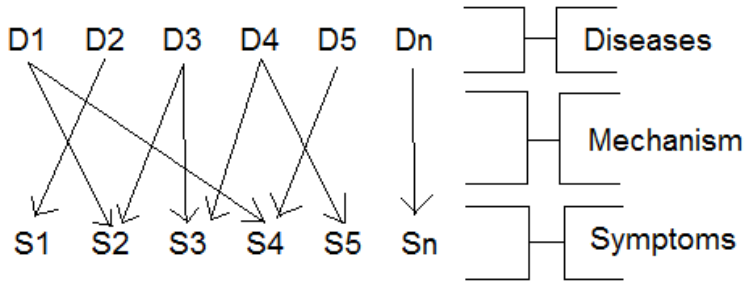

Figure 3.8: Network structure MUNIN

3.4.3 Conclusion

Technical

1. The accuracy of the network drops because it does not handle measurement of nerve signals.

2. Multiple simultaneous diseases are not considered which makes the model less accurate.

3. Because it uses only signals from one muscle it's less accurate

Non-technical

1. Because it only focuses on one type of disease this will limit the results of the application.

2. Because it uses only signals from one muscle it is possible that more then one test is needed.

3. As a medical diagnosis aid we have to deal with the legal liability issues of misdiagnoses.

3.5 TREAT

3.5.1 Summary

TREAT is a decision support system for antibiotic treatment in inpatients with common bacterial infections. It was tested in a randomized controlled trial in three countries and shown to improve the percentage of appropriate empirical antibiotic treatments, while at the same time reducing hospital stay and the use of broad-spectrum antibiotics. TREAT is based on a causal probabilistic network and uses a cost–benefit model for antibiotic treatment, including costs assigned to future resistance. In the present review we discuss the advantages of using causal probabilistic models for prediction and decision support, and the various decisions that were taken in the TREAT project.

TREAT was calibrated and installed in three locations: Rabin Medical Center, Beilinson Campus, Petah-Tiqva, Israel (six wards of medicine); Gemelli Hospital in Rome, Italy (three wards of infectious diseases); and Freiburg University Hospital, Freiburg, Germany (six wards of medicine). In the cohort studies included 1203 patients, of whom 350 had an identified bacterial pathogen. TREAT recommended appropriate antibiotic treatment for 70% of patients, vs.. 57% actually prescribed by physicians (P= 0.0001). The cost of antibiotics prescribed by TREAT was lower by almost 50% compared to those actually prescribed by the physicians. The results were similar at the three medical centers [23].

The CPN was built from distinct modules, each module representing one site of infection. Fig. 3.9 depicts a general model of a site of infection. **Pathogen1 to Pathogen_n** represent the potential pathogens of infections

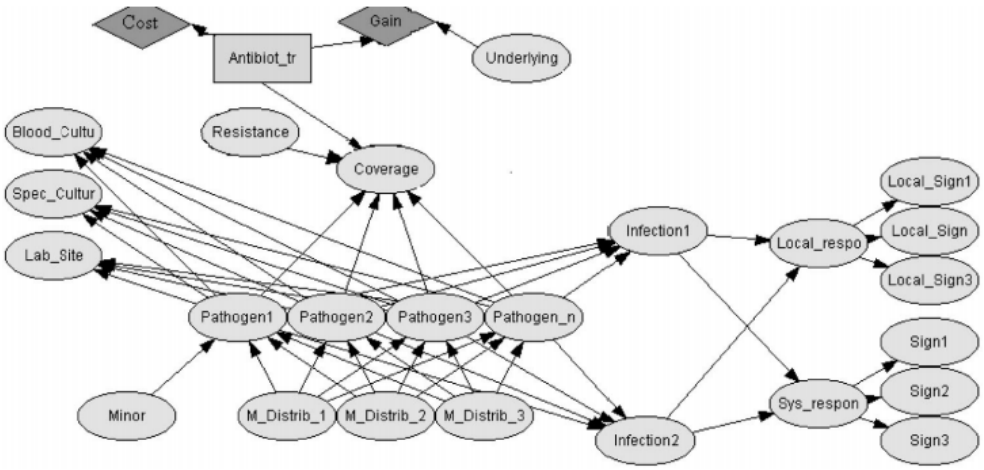

Figure 3.9: A general scheme for a site-of-infection network

at the given site. The states of the **Pathogen** nodes are severity states, with a risk of mortality associated with each state.

The probability of an infection caused by a **Pathogen** is determined by its prevalence in major patient-groups (**M_Distrib_1 to M_Distrib_3**). We selected a factor as defining a major patient-group if it emerged as a strong and independent predictor for infection and distribution of pathogens on statistical analysis of our databases, if, according to present knowledge, it has a clear patho-physiological contribution to the risk of infection at this site, and if the data on the prevalence of infection and distribution of severity-states and pathogens are available. Several factors qualified as minor distribution factors(**Minor**), i.e., factors that change the likelihood of one without affecting the overall risk for infection.

Any of the pathogens can cause an infection and infections can manifest as different patterns (**Infection1 and Infection2**). Infection will cause a local response (**Local_respo**), specific to each site of infection, and the local response will manifest as local signs and symptoms, e.g., cough and pains on inspiration caused by pneumonia (**Local_sign1 to Local_sign3**). It will also cause a systemic response (**Sys_respon**) common to all sites of infection and manifesting as generalized signs and symptoms (**Sign1 to Sign3**), such as fever, rapid pulse, and hypotension. A pathogen causing an infection will grow in local specimens (**e.g., urine or sputum, Spec_cultur**) and in the blood (**Blood_cultu**). It can cause other changes, detectable by tests and specific for the site [24].

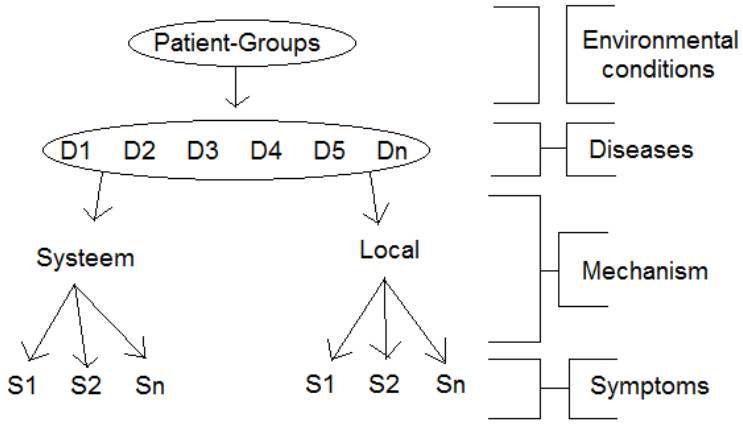

Figure 3.10: Network structure TREAT

3.5.2 Structure

The TREAT structure is much like the structure in section 3.1: Structure of a medical Bayesian network. The diseases are the same as in that structure. The symptoms are different, because we still have all the symptoms, but they are split into two groups of symptoms that occur at the infection site and symptoms that show up in the entire system. The environmental conditions are in this network patient groups that change the chances of getting infected. The mechanism is different, because it is split up into two part just like the symptoms. If we take the two parts to getter we will get the normal mechanism. In figure 3.10 we can see the structure of the TREAT network.

3.5.3 Conclusion

Technical

1. As you can see in the summary there is a 70% accuracy. This means it is still wrong 30% of the time, but if we compare this to the prescription of the physicians that are wrong 43

Non-technical

1. Because it only focuses on one type of disease this will limited the results of the application.

3.6 General conclusion

In this chapter we have considered the structure of five different medical Baysian networks. With all of them we looked at technical and non-technical problems or difficulties these systems have. If we look at all the technical conclusions we can see that marginal independence of diseases this limiteds the accuracy. Also conditional independence limiteds the accucay in four of the five systems. Lastly we also have findings only as manifestations of disease this means that the environmental conditions are not used in fout of the five networks. If we then look at the non-technical problems with these systems we will see that the biggest problem is that they do not cover the whole of medicine which limits the results of these systems. The second problem is that these systems have to deal with the legal liability issues of misdiagnosis. This problem can be solved by, for example by requesting the user to accept limited liability before the systeem can be consulled. Another possibly is if it is used by the hospital to always let a doctor make the last decision. If we look at the structure of the network we can see the technical problems with these systems. We can see that the mechanism in all these networks is always pathophysiology. Also we can see that most of them miss the environmental conditions in there structure. These models are smaller then the system we are looking at, but this networks are already very complex. So we have to look at ways reducing the complexity if we want to develop a large-scale network.

Chapter 4

Towards Large Diagnostic Bayesian Network Models

In Chapter 3 we considered large systems that were developed in the past. Here we can also discussed the technical and non-technical problems associated to these systems. In this chapter I discuss four different ways these systems might be developed despite the obstacles mentioned above. The first way is to split up the network. This will make the network easier to develop and less complex, but will lose accuracy. There are different ways that one can split these networks. The first three solutions I discuss are splitting on type of disease, medical specialty and body parts. A further possibility is to change the structure of the network in to a bipartite graph. This solution will make it easier, because we only have two subsets that have no connections within the subsets. This solution makes it less complex, but also less accurate and here we have to find a solution for using environmental conditions.

4.1 Splitting-up the network

Developing a network for the whole of medicine is a gigantic task, because we have to deal with thousands of diseases. A possible way to solve this problem is developing a smaller and simpler network that only cover parts of the whole domain. In the related work a number of already developed networks have been discused most of these are networks that deal with one type of disease. Another way to split a network is to look at the hospital departments or body parts. Choosing to split up this network will make it easier to eliminate the problems in these Bayesian networks. It is not possible to have directed cycles in a Bayesian network, but it is possible to have cycles if we don't look at the directed part of this network. The more of these cycles there are in a network the more complex this network is this in turn will make it more difficult for the algorithm to calculate the

correct diseases in a short time. For an example of these cycles see figure 4.1. It is possible to remove this problem by combing different symptoms in one node or by removing one of the arcs. The complexity of this network is then determined by the tree width (the number of symptoms or diseases that are in one node). If we split a network we can eliminate certain cycles before we even have to combining the different diseases. This will make this network less complex and therefore easier to compute the correct disease in a respectable time.

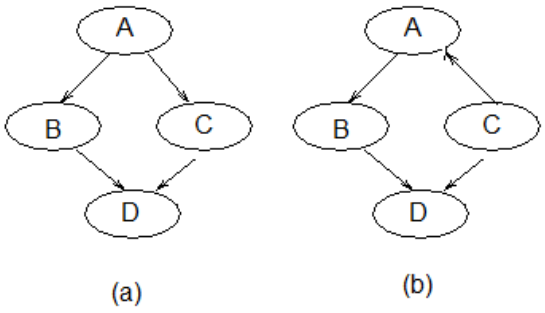

Figure 4.1: (a) is a cyclic that is allowed and can make it diffecult to use inference. (b) is a directed cyclic so these are not allow in a Baysian network.

Now that we have eliminated some of the cycles we still have to deal with the second problem, which is that there are too many parents to a node will make it more difficult to calculate. The QMR-DT network tried to solve this problem by using a noisy-OR gate. This will take all the incoming arcs and calculate them one at a time see figure 4.2 and 4.3 for a example of the noisy-OR gate. If we split the network it's very likely that some of the incoming arcs to nodes with very many parents will be removed, because they belong to a different group. This will simplify the network even more, which makes the algorithms faster.

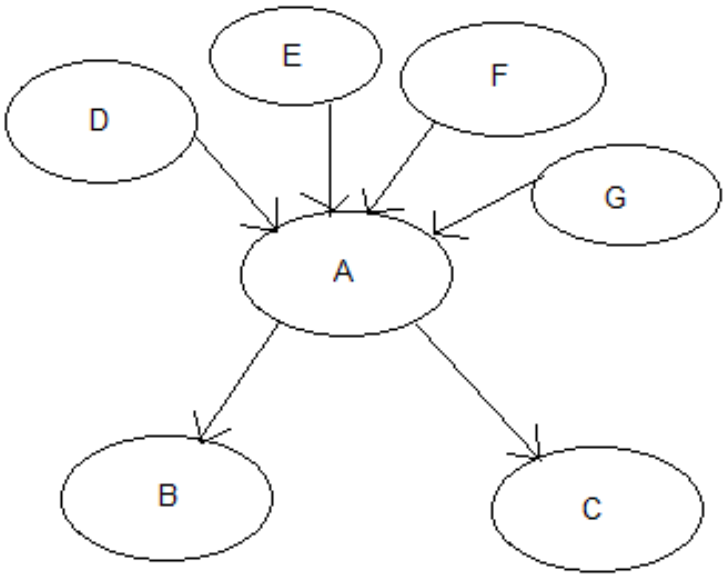

Figure 4.2: Example of small network with a node with many parents

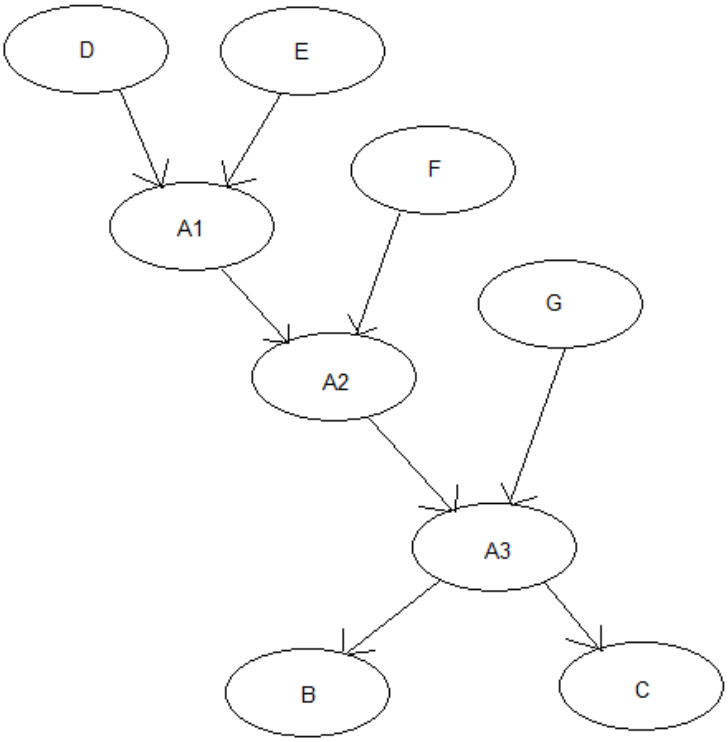

Figure 4.3: Where we can see the use of the noisy-OR gate. As you can see we have split up the incoming arcs to A and handle them separately. As you see this will result in more nodes

So if we look at the complete picture of a network when we use splitting, we will first limit the complexities by removing cycles and arcs. If we want to simplify the network even more we can combine symptoms and disease to remove even more cycles and arcs between the nodes. Because of the split we will also reduce the tree if we use combining because there are less nodes to combine. This in turn will reduce the complexity of the network.

4.1.1 Types of diseases

Splitting-up the network in to these different types of diseases will also cause loss of accuracy. The reason for the loss of this accuracy is the removal of arcs between the diseases and symptoms that are in different types of diseases, but will still have an influence on each other. If we build a network with for example 10 nodes and 12 arcs. See figure 4.4. If we want to split this network, that is because the diseases fall into different categories. If we look at figure 4.5 we can see that nodes A, B, C, D and E fall under the first type of disease and F, G, H, I and J under the second type of disease. As you can see, in order to split this network we have to eliminate three arcs

D-F, D-G and E-G, because they cross over to the other type of disease. Since we have eliminate these three arcs we have also influenced the chance in this network, because if we know that D is true then this would change the chance of F and G.

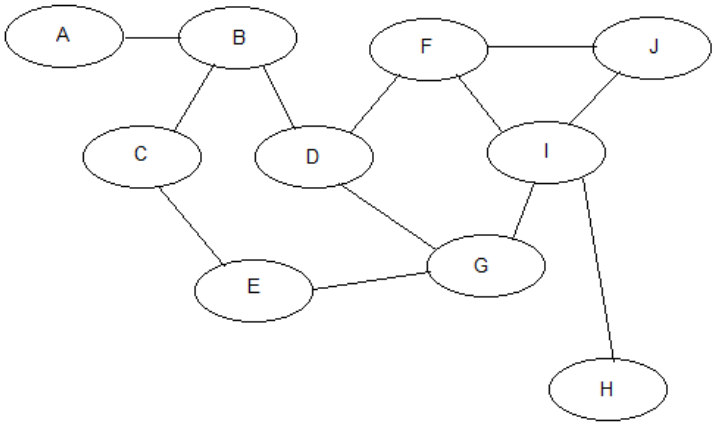

Figure 4.4: Example of small network

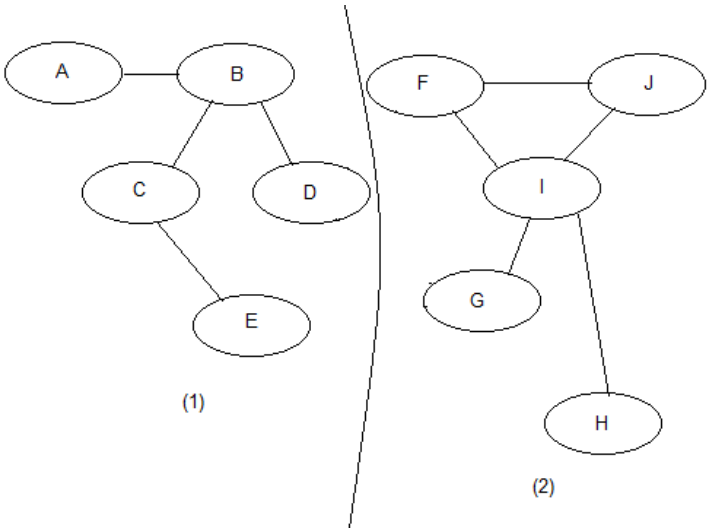

Figure 4.5: Example of a split network

We can also change the complexities and the accuracy of these networks by making some assumptions will building this network:

The first possibility is choosing to develop a network that will pick a type of disease. In this network you still need all the symptoms in the network.

The big change here is the diseases are all changed in their types. This type of network is better for patients, because here they get results instead of a Medical specialty or hospital department like in the second solution.

By using only the types it's also easier to build. The network becomes smaller, so this will save time in defining the node probability tables for each node of the graph and building the graph structure itself. Because the network is simpler, it's also easier to develop a means of easily identifying and constructing the components that form the foundations of a Bayesian network.

Secondly it's possible to make the assumption that all symptoms are binary, because severity is important in diseases and can change the probability of having a disease. In this network were not determining the disease so it does not matter what the severity of the symptoms are, because it will still the system will still pick the same type of disease. So using this assumption in this network won't change the accuracy of the complete network, but it will make it simpler. Because there are less value need which makes it easier to develop this network.

Because the network is now less complex we can also choose to add symptoms that are no manifestations of diseases. This means adding historical findings as possible conditions that have influence on the chances of having a certain type of disease. This will add complexity to the network, but it will make the network more accurate. The complexity of adding these historical data can be changed, because there are many historical findings, but you can limit them to the most common or the most influential findings. Most networks already developed don't use these conditions so this could be a great addition to this network.

We make the assumption that the different disease types will also be developed in to separate networks. Then networks of these different types of diseases will use far less symptoms, because not all symptoms are linked to these diseases. This means that we can make these networks a little more complex by not using binary values but normative values for some symptoms. We can limit the complexity of adding the severity and combining different nodes to reduce effects of combinatorial explosion. This is done by using the diverging and converging. If for example we have a graph with nodes A, B, C and D with arcs B-A, C-A and D-A and each node has four states. Were the nodes C and D can have a common synthetic node. It's possible to use that node to limited the conditional probability table, because without using the synthetic node there are $4^4 = 256$ probability values and with the synthetic node there are $4^3 + 4^3 = 64$ probability values for A.

The problem of reduced accuracy, because some types of disease share symptoms, can be countermanded by choosing not one but multiple types of disease networks. This will take more time, but the accuracy will increase. To make sure there is less time wasted on unnecessary calculations we can use the number of symptoms and/or the probabilities. If we use symptoms and probabilities at the same time we can for instance say that if the different between the highest probability and any other probabilities is 5% and we have 5 symptoms we will use those types of diseases in the next calculations.

It might be possible to develop a complete system that uses these networks. Imagine this system as a program that asks what kind of symptoms the patient has. After that the types of disease network is used to determine one of the types of disease this patient has. If the system remembers what the symptoms were that were used to find the type of disease it can use these to ask for more details, where this is possible, and so still use the accuracy of using severity in the symptoms, but speed up the calculations by limiting the symptoms and diseases that have to be calculated.

These ten types of diseases can for example be used to split the complete medical network in the smaller parts that you can use.

1. Cancer
2. Viral infections
3. Bacterial infections
4. Autoimmune diseases
5. Heart disease
6. Digestive diseases
7. Thyroid diseases
8. Blood diseases
9. Neurodegenerative diseases
10. Sexually transmitted diseases

4.1.2 Medical specialty

The second possibility is choosing to develop a network that will pick a department of the hospital. In this network you still need all the symptoms in this network. The big change here is the diseases are all changed in the departments of the hospital. This is a network more for the hospital than

for a patient, because if we make the same assumptions as with the types of disease then networks for every department of the hospital have to be developed. This way doctors of different department's only use their own departments Bayesian network. This will result in less time needed to find the disease, because only the diseases in that department have to be used in those networks.

The two biggest problems with using this split are overlapping diseases and the networks are going to do parts of the doctor's work. The first problem is that, when you are in a hospital, it is possible to get treated in multiple departments. This means that there are overlapping diseases between these departments; this is not a problem for the separate networks for every department, but it is a problem for the first network that picks the department. This means that it will take longer to get to the right disease. Also the idea of reducing the accuracy by using probabilities and number of symptoms is problematic, because this wil onlyl result in longer searches. The second problem is mostly about using and accepting the program and networks in their routines. In most companies with big changes in the use of IT there is commonly some resistance against these changes. It is also possible that doctors won't use, it because they think they can do it better than the Bayesian networks.

4.1.3 Body parts

A third possibility is to develop a network that will be split according to the location of the disease in the body. In this network you still need all the symptoms. The big change here is that diseases are all are grouped according to their location. This can cause some problems, because it's possible that the disease is located in a different part of the body from the symptoms of this disease. It might be possible to avoid this problem by not asking this as the first question. This way it's a little more narrowed down before you get the information on the body part. Also you have to consider diseases that are not located in one part of the body or located in the complete body.

4.2 Medical bipartite graph

Yet another possibility is choosing to switch to another graph like the bipartite graph. In this network you will still need all the symptoms and diseases. The big change here is the symptoms are put into a disjoint set and the diseases in to another disjoint set. There are three big challenges in making a bipartite graph for complex networks which are logarithmic average distance, high clustering and power law degree distribution. If the network does not have these properties it will be very complex to make a

bipartite graph of this network.

A complete medical network is a very complex network. If we look at some of these complex networks we will see that they already have a bipartite graph structure. For example if we look at movies we will see that a link between actors and the movies in which they feature will we see a naturally occurring bipartite graph structure. This also happens with symptoms and the diseases with which they are linked. These will make it easier to build a network, because we can follow the naturally occurring structure. When building this system the designer also has to consider minimum length of the paths joining the disease and the symptom and to keep the overview easier; you would want to design a graph with a minimal of crossing lines. For developing a network with the minimal of crossing lines we have programs.

In a medical graph we could use symptom groups like nervous system symptoms, eye symptoms and heart symptoms. We can then make a group containing the diseases and a group containing the symptoms. Splitting the symptoms in groups will also make building the network easier. There are certain data that are not symptoms, but do affect different diseases these data or conditions can be modeled into this type of network. There are also findings or conditions for example age that have an effect on symptoms. These findings and conditions can't be used in the network itself. However, it is possible to chance the probability distribution in the subset of symptoms. This will improve the accuracy of this network

An advantage of using this type of network is that we can use causal independence with less problems, because this will only cause problems if two symptoms operate a through common pathway; this is impossible in a bipartite graph structure. Causal independence is what maintains the mechanisms by which symptoms operate independently of one another and independently of any other events that may cause the symptom to occur, such as the influence of other symptoms or data. Using causal independence makes the network simpler and easier to track back if there are problems, but will result in a less accurate network.

Besides causal independence we can also have to make assumptions about marginal independence and conditional independence. Marginal independence means that the diseases are not connected to each other, meaning there are no arcs between the diseases. If we look at the structure of a bipartite graph we will see that it is impossible to make a connection between diseases, because there are no arcs possible within the set of diseases. This will decrease the accuracy of this network, but will make it possible to calculate the disease faster. The loss in accuracy with this assumption has less impact than the time you win in with calculations, because there are very

few diseases that have influence on other diseases.

Conditional independence of data can be seen in these networks because there are no arcs in the set of symptoms. This is one of the advantages of using this type of graph, because conditional independence of findings will make the network simpler. Since we are talking about observed symptoms we can say that this will not jeopardise the accuracy of this network, because a first observation will not change if we observe a second symptom.

Chapter 5

Conclusions

The aim of this study was to find different ways of developing a diagnostic system that covers the whole of medicine. We have seen various systems that have been developed in the past and the problems that developers faced when developing these network. It is safe to say that networks that are more specialized can be developed with a very good accuracy. These networks can be used in newly developed devices and applications that are used to help patients with their diseases. Examples of these applications are computerized ECG analysis, automated arterial blood gas interpretation and automated protein electrophoresis reports. We have also looked at splitting the network or using a bipartite network as possible solutions for the current problems. What could the future of large-scale diagnostic systems look like?

It is very likely that large-scale diagnostic systems will be developed in the future. These days many companies are working on the development of such networks. So it is very likely that one of these companies will also succeed, but before this can happen a number of major challenges remain to be met before large-scale diagnostic systems can be used successfully. We have extensively discussed some of these problems in chapter 4. Of course, other problems may also occur like knowledge base maintenance is not always up-to-date. This however a critical requirement to determine the validity of these large-scale diagnostic systems. This means that we have to take into account the occurrence of new diseases and symptoms and the change of the probability of having a disease given certain symptoms.

Another issue that can determine the success of large-scale diagnostic systems is the environment. The smaller systems that are focused on one disease or some diseases have a higher chance of getting adopted into the community for which they are intended, while doctors in general medical, for whom the large-scale systems are intended, may not experience the need for diagnostic assistance on a frequent enough basis to justify purchase of

one or more such systems. A possible direction for future research is to look at automated hospital information systems in combination with these large-scale diagnostic systems. This way the patient data is provide by the hospital information system and this means that the doctor does not have to manually enter all of a patient data in order to obtain a result from these large-scale diagnostic systems. However, it is not so easy to transfer the information about a patient from a hospital information system to these large-scale diagnostic systems. So we can say that there are some problems left to be resolved that could create excellent subjects for future research.

Bibliography

[1] Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[2] Bayesian network. `http://en.wikipedia.org/wiki/Bayesian_network`, May 2013.

[3] Directed acyclic graph. `http://en.wikipedia.org/wiki/Directed_acyclic_graph`, May 2013.

[4] Bipartite graph. `http://en.wikipedia.org/wiki/Bipartite_graph`, June 2013.

[5] Judea Pearl. The causal foundations of structural equation modeling. Technical report, DTIC Document, 2012.

[6] Remco Ronaldus Bouckaert. *Bayesian Belief Networks: from Construction to Inference*. PhD thesis, Universiteit Utrecht, Faculteit Wiskunde en Informatica, 1995.

[7] Carsten Riggelsen. Induction of bayesian networks with a priori domain knowledge. Master's thesis, University of Utrecht, 2002.

[8] Chain rule. `http://en.wikipedia.org/wiki/Chain_rule_(probability)`, May 2013.

[9] Marginal distribution. `http://en.wikipedia.org/wiki/Marginal_distribution`, June 2013.

[10] Probability mass function. `http://en.wikipedia.org/wiki/Probability_mass_function`, June 2013.

[11] Conditional probability distribution. `http://en.wikipedia.org/wiki/Conditional_probability_distribution`, June 2013.

[12] Bayes' theorem. `http://en.wikipedia.org/wiki/Bayes'_theorem`, June 2013.

[13] Rasa Jurgelenaite and Peter JF Lucas. Exploiting causal independence in large bayesian networks. *Knowledge-Based Systems*, 18(4):153–162, 2005.

[14] Malcolm Pradhan, Gregory Provan, Blackford Middleton, and Max Henrion. Knowledge engineering for large belief networks. In *Proceedings of the Tenth international conference on Uncertainty in artificial intelligence*, pages 484–490. Morgan Kaufmann Publishers Inc., 1994.

[15] David Heckerman, Dan Geiger, and David M Chickering. Learning bayesian networks: The combination of knowledge and statistical data. Technical Report 3, 1995.

[16] Clive R Hollin and Great Britain. *Pathfinder programmes in the Probation Service: A retrospective analysis*. Home Office London, 2004.

[17] David Earl Heckerman, Eric J Horvitz, and Bharat N Nathwani. Toward normative expert systems: The pathfinder project. Technical report, National Libary of Medicine, 1990.

[18] Michael A Shwe, B Middleton, DE Heckerman, M Henrion, EJ Horvitz, HP Lehmann, and GF Cooper. Probabilistic diagnosis using a reformulation of the internist-1/qmr knowledge base. *Methods of information in Medicine*, 30(4):241–255, 1991.

[19] Tommi S. Jaakkola and Micheal I. Jordan. Variational probabilistic inference and the qmr-dt network. *Journal of artificial Intelligence Research*, pages 291–322, 1999.

[20] Michael Shwe, Blackford Middleton, David Heckerman, Max Henrion, Eric Horvitz, Harold Lehmann, and Gregory Cooper. A probabilistic reformulation of the quick medical reference system. In *Proceedings of the Annual Symposium on Computer Application in Medical Care*, page 790. American Medical Informatics Association, 1990.

[21] Steen Andreassen, Marianne Woldbye, Bjørn Falck, and Stig K. Andersen. Munin - a causal probabilistic network for interpretation of electromyographic findings. pages 366–372.

[22] Gregory F Cooper. The computational complexity of probabilistic inference using bayesian belief networks. *Artificial intelligence*, 42(2):393–405, 1990.

[23] Leonard Leibovici, Mical Paul, Anders D Nielsen, Evelina Tacconelli, and Steen Andreassen. The treat project: decision support and prediction using causal probabilistic networks. *International journal of antimicrobial agents*, 30:93–102, 2007.

[24] Leonard Leibovici, Michal Fishman, Henrik C Schonheyder, Christian Riekehr, Brian Kristensen, Ilana Shraga, and Steen Andreassen. A causal probabilistic network for optimal treatment of bacterial infections. *Knowledge and Data Engineering, IEEE Transactions on*, 12(4):517–528, 2000.