Bayesian Networks, Introduction and Practical Applications (final draft)

Wim Wiegerinck, Willem Burgers, Bert Kappen

Abstract In this chapter, we will discuss Bayesian networks, a currently widely accepted modeling class for reasoning with uncertainty. We will take a practical point of view, putting emphasis on modeling and practical applications rather than on mathematical formalities and the advanced algorithms that are used for computation. In general, Bayesian network modeling can be data driven. In this chapter, however, we restrict ourselves to modeling based on domain knowledge only. We will start with a short theoretical introduction to Bayesian networks models and inference. We will describe some of the typical usages of Bayesian network models, e.g. for reasoning and diagnostics; furthermore, we will describe some typical network behaviors such as the explaining away phenomenon, and we will briefly discuss the common approach to network model design by causal modeling. We will illustrate these matters by a detailed modeling and application of a toy model for medical diagnosis. Next, we will discuss two real-world applications. In particular we will discuss the modeling process in some details. With these examples we also aim to illustrate that the modeling power of Bayesian networks goes further than suggested by the common textbook toy applications. The first application that we will discuss is for victim identification by kinship analysis based on DNA profiles. The distinguishing feature in this application is that Bayesian networks are generated and computed on-the-fly, based on case information. The second one is

Wim Wiegerinck
SNN Adaptive Intelligence, Donders Institute for Brain, Cognition and Behaviour, Radboud University Nijmegen, Geert Grooteplein 21, 6525 EZ Nijmegen, The Netherlands e-mail: w.wiegerinck@science.ru.nl

Willem Burgers
SNN Adaptive Intelligence, Donders Institute for Brain, Cognition and Behaviour, Radboud University Nijmegen, Geert Grooteplein 21, 6525 EZ Nijmegen, The Netherlands e-mail: w.burgers@science.ru.nl

Bert Kappen
SNN Adaptive Intelligence, Donders Institute for Brain, Cognition and Behaviour, Radboud University Nijmegen, Geert Grooteplein 21, 6525 EZ Nijmegen, The Netherlands e-mail: b.kappen@science.ru.nl

an application for petrophysical decision support to determine the mineral content of a well based on borehole measurements. This model illustrates the possibility to model with continuous variables and nonlinear relations.

1 Introduction

In modeling intelligent systems for real world applications, one inevitably has to deal with uncertainty. This uncertainty is due to the impossibility to model all the different conditions and exceptions that can underlie a finite set of observations. Probability theory provides the mathematically consistent framework to quantify and to compute with uncertainty. In principle, a probabilistic model assigns a probability to each of its possible states. In models for real world applications, the number of states is so large that a sparse model representation is inevitable. A general class with a representation that allows modeling with many variables are the Bayesian networks [26, 18, 7].

Bayesian networks are nowadays well established as a modeling tool for expert systems in domains with uncertainty [27]. Reasons are their powerful but conceptually transparent representation for probabilistic models in terms of a network. Their graphical representation, showing the conditional independencies between variables, is easy to understand for humans. On the other hand, since a Bayesian network uniquely defines a joint probability model, inference — drawing conclusions based on observations — is based on the solid rules of probability calculus. This implies that the mathematical consistency and correctness of inference are guaranteed. In other words, all assumptions in the method are contained in model, i.e., the definition of variables, the graphical structure, and the parameters. The method has no hidden assumptions in the inference rules. This is unlike other types of reasoning systems such as e.g., Certainty Factors (CFs) that were used in e.g., MYCIN — a medical expert system developed in the early 1970s [29]. In the CF framework, the model is specified in terms of a number of if-then-else rules with certainty factors. The CF framework provides prescriptions how to invert and/or combine these if-then-else rules to do inference. These prescriptions contain implicit conditional independence assumptions which are not immediately clear from the model specification and have consequences in their application [15].

Probabilistic inference is the problem of computing the posterior probabilities of unobserved model variables given the observations of other model variables. For instance in a model for medical diagnoses, given that the patient has complaints x and y, what is the probability that he/she has disease z? Inference in a probabilistic model involves summations or integrals over possible states in the model. In a realistic application the number of states to sum over can be very large. In the medical example, the sum is typically over all combinations of unobserved factors that could influence the disease probability, such as different patient conditions, risk factors, but also alternative explanations for the complaints, etc. In general these computations are intractable. Fortunately, in Bayesian networks with a sparse graphical

structure and with variables that can assume a small number of states, efficient inference algorithms exists such as the junction tree algorithm [18, 7].

The specification of a Bayesian network can be described in two parts, a qualitative and a quantitative part. The qualitative part is the graph structure of the network. The quantitative part consists of specification of the conditional probability tables or distributions. Ideally both specifications are inferred from data [19]. In practice, however, data is often insufficient even for the quantitative part of the specification. The alternative is to do the specification of both parts by hand, in collaboration with domain experts. Many Bayesian networks are created in this way. Furthermore, Bayesian networks are often developed with the use of software packages such as Hugin (www.hugin.com), Netica (www.norsys.com) or BayesBuilder (www.snn.ru.nl). These packages typically contain a graphical user interface (GUI) for modeling and an inference engine based on the junction tree algorithm for computation.

We will discuss in some detail a toy application for respiratory medicine that is modeled and inferred in this way. The main functionality of the application is to list the most probable diseases given the patient-findings (symptoms, patient background revealing risk factors) that are entered. The system is modeled on the basis of hypothetical domain knowledge. Then, it is applied to hypothetical cases illustrating the typical reasoning behavior of Bayesian networks.

Although the networks created in this way can be quite complex, the scope of these software packages obviously has its limitations. In this chapter we discuss two real-world applications in which the standard approach to Bayesian modeling as outlined above was infeasible for different reasons: the need to create models on-the-fly for the data at hand in the first application and the need to model continuous-valued variables in the second one.

The first application is a system to support victim identification by kinship analysis based on DNA profiles (Bonaparte, in collaboration with NFI). Victims should be matched with missing persons in a pedigree of family members. In this application, the model follows from Mendelian laws of genetic inheritance and from principles in DNA profiling. Inference needs some preprocessing but is otherwise reasonably straightforward. The graphical model structure, however, depends on the family structure of the missing person. This structure will differ from case to case and a standard approach with a static network is obviously insufficient. In this application, modeling is implemented in the engine. The application generates Bayesian networks on-the-fly based on case information. Next, it does the required inferences for the matches.

The second model has been developed for an application for petrophysical decision support (in collaboration with SHELL E&P). The main function of this application is to provide a probability distribution of the mineral composition of a potential reservoir based on remote borehole measurements. In this model, variables are continuous valued. One of them represents the volume fractions of 13 minerals, and is therefore a 13-D continuous variable. Any sensible discretization in a standard Bayesian network approach would lead to a blow up of the state space. Due to non-

linearities and constraints, a Bayesian network with linear-Gaussian distributions [3] is also not a solution.

The chapter is organized as follows. First, we will provide a short introduction to Bayesian networks in section 2. In the next section we will discuss in detail modeling basics and the typical application of probabilistic reasoning in the medical toy model. Next, in sections 4 and 5 we will discuss the two real-world applications. In these chapters, we focus on the underlying Bayesian network models and the modeling approaches. We will only briefly discuss the inference methods that were applied whenever they deviate from the standard junction tree approach. In section 6, we will end with discussion and conclusion.

2 Bayesian Networks

In this section, we first give a short and rather informal review of the theory of Bayesian networks (subsection 2.1). Furthermore in subsection 2.2, we briefly discuss Bayesian networks modeling techniques, and in particular the typical practical approach that is taken in many Bayesian network applications.

2.1 Bayesian Network Theory

To introduce notation, we start by considering a joint probability distribution, or probabilistic model, $P(X_1,\ldots,X_n)$ of n stochastic variables $X_1,\ldots,X_n$. Variables X_j can be in state x_j. A state, or value, is a realization of a variable. We use shorthand notation

$$P(x_1,\ldots,x_n) = P(X_1 = x_1,\ldots,X_n = x_n) \tag{1}$$

to denote the probability (in continuous domains: the probability density) of variables X_1 in state x_1, variable X_2 in state x_2 etc.

A Bayesian network is a probabilistic model P on a finite directed acyclic graph (DAG). For each node i in the graph, there is a random variable X_i together with a conditional probability distribution $P(x_i|x_{\pi(i)})$, where $\pi(i)$ are the parents of i in the DAG, see figure 1. The joint probability distribution of the Bayesian network is the product of the conditional probability distributions

$$P(x_1,\ldots,x_n) = \prod_{i=1}^{n} P(x_i|x_{\pi(i)}) \; . \tag{2}$$

Since any DAG can be ordered such that $\pi(i) \subseteq 1,\ldots i-1$ and any joint distribution can be written as

$$P(x_1,\ldots,x_n) = \prod_{i=1}^{n} P(x_i|x_{i-1},\ldots,x_1) \; , \tag{3}$$

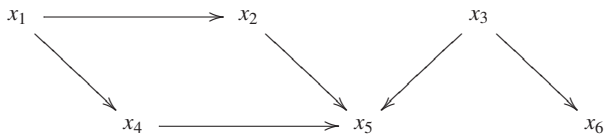

Fig. 1 DAG representing a Bayesian network $P(x_1)P(x_2|x_1)P(x_3)P(x_4|x_1)P(x_5|x_2,x_3,x_4)P(x_6|x_3)$

it can be concluded that a Bayesian network assumes

$$P(x_i|x_{i-1},\ldots,x_1) = P(x_i|x_{\pi(i)}) \,. \tag{4}$$

In other words, the model assumes: given the values of the direct parents of a variable X_i, this variable X_i is independent of all its other predecessing variables in the graph.

Since a Bayesian network is a probabilistic model, one can compute marginal distributions and conditional distributions by applying the standard rules of probability calculus. For instance, in a model with discrete variables, the marginal distribution of variable X_i is given by

$$P(x_i) = \sum_{x_1} \cdots \sum_{x_{i-1}} \sum_{x_{i+1}} \cdots \sum_{x_N} P(x_1,\ldots,x_N) \,. \tag{5}$$

Conditional distributions such as $P(x_i|x_j)$ are obtained by the division of two marginal distributions

$$P(x_i|x_j) = \frac{P(x_i,x_j)}{P(x_j)} \,. \tag{6}$$

The bottleneck in the computation is the sum over combinations of states in (5). The number of combinations is exponential in the number of variables. A straightforward computation of the sum is therefore only feasible in models with a small number of variables. In sparse Bayesian networks with discrete variables, efficient algorithms that exploit the graphical structure, such as the junction tree algorithm [21, 18, 7] can be applied to compute marginal and conditional distributions. In more general models, exact inference is infeasible and approximate methods such as sampling have to be applied [22, 3].

2.2 Bayesian Network Modeling

The construction of a Bayesian network consists of deciding about the domain, what are the variables that are to be modeled, and what are the state spaces of each of the variables. Then the relations between the variables have to be modeled. If these are to be determined by hand (rather than by data), it is a good rule of thumb to

construct a Bayesian network from cause to effect. Start with nodes that represent independent root causes, then model the nodes which they influence, and so on until we end at the leaves, i.e., the nodes that have no direct influence on other nodes. Such a procedure often results in sparse network structures that are understandable for humans [27].

Sometimes this procedure fails, because it is unclear what is cause and what is effect. Is someone's behavior an effect of his environment, or is the environment a reaction on his behavior? In such a case, just avoid the philosophical dispute, and return to the basics of Bayesian networks: a Bayesian network is not a model for causal relations, but a joint probability model. The structure of the network represents the conditional independence assumptions in the model and nothing else.

A related issue is the decision whether two nodes are really (conditionally) independent. Usually, this is a matter of simplifying model assumptions. In the true world, all nodes should be connected. In practice, reasonable (approximate) assumptions are needed to make the model simple enough to handle, but still powerful enough for practical usage.

When the variables, states, and graphical structure is defined, the next step is to determine the conditional probabilities. This means that for each variable x_i, the conditional probabilities $P(x_i|x_{\pi(i)})$ in eqn. (4) have to be determined. In case of a finite number of states per variable, this can be considered as a table of $(|x_i|-1)\times|x_{\pi(i)}|$ entries between 0 and 1, where $|x_i|$ is the number of states of variable x_i and $|x_{\pi(i)}|=\prod_{j\in\pi(i)}|x_j|$ the number of joint states of the parents. The -1 term in the $(|x_i|-1)$ factor is due to the normalization constraint $\sum_{x_i}P(x_i|x_{\pi(i)})=1$ for each parent state. Since the number of entries is linear in the number of states of the variables and exponential in the number of parent variables, a too large state space as well as a too large number of parents in the graph makes modeling practically infeasible.

The entries are often just the result of educated guesses. They may be inferred from data, e.g. by counting frequencies of joint occurences of variables in state x_i and parents in states $x_{\pi(i)}$. For reliable estimates, however, one should have sufficiently many data for each joint state $(x_i,x_{\pi(i)})$. So in this approach one should again be carefull not to take state space and/or number of parents too large. A compromise is to assume a parametrized tables. A popular choice for binary variables is the noisy-OR table [26]. The table parametrization can be considered as an educated guess. The parameters may then be estimated from data.

Often, models are constructed using Bayesian network software such as the earlier mentioned software packages. With the use of a graphical user interface (GUI), nodes can be created. Typically, nodes can assume only values from a finite set. When a node is created, it can be linked to other nodes, under the constraint that there are no directed loops in the network. Finally — or during this process — the table of conditional probabilities are defined, manually or from data as mentioned above. Many Bayesian networks that are found in literature fall into this class, see e.g., www.norsys.com/netlibrary/. In figure 2, a part of the ALARM network as represented in BayesBuilder (www.snn.ru.nl/) is plotted. The ALARM network was originally designed as a network for monitoring patients in intensive care [2]. It con-

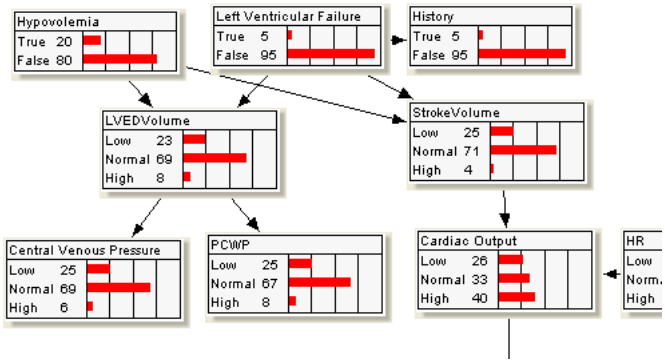

Fig. 2 Screen shot of part of the 'Alarm network' in the BayesBuilder GUI

sists of 37 variables, each with 2, 3, or 4 states. It can be considered as a relatively large member of this class of models. An advantage of the GUI based approach is that a small or medium sized Bayesian network, i.e., with up to a few dozen of variables, where each variable can assume a few states, can be developed quickly, without the need of expertise on Bayesian networks modeling or inference algorithms.

3 An Example Application: Medical Diagnosis

In this section we will consider the a Bayesian network for medical diagnosis of the respiratory system. This is model is inspired on the famous 'ASIA network' described in [21].

3.1 Modeling

We start by considering the the following piece of qualitative 'knowledge':

> The symptom *dyspnoea* (shortness of breath) may be due to the diseases *pneumonia*, *lung cancer*, and/or *bronchitis*. Patients with *pneumonia*, and/or *bronchitis* often have a very nasty *wet coughing*. *Pneumonia*, and/or *lung cancer* are often accompanied by a heavy *chest pain*. *Pneumonia* is often causing a severe *fever*, but this may also be caused by a *common cold*. However, a *common cold* is often recognized by a *runny nose*. Sometimes, *wet coughing*, *chest pain*, and/or *dyspnoea* occurs unexplained, or are due to another cause, without any of these diseases being present. Sometimes diseases co-occur. A *weakened immune-system* (for instance, homeless people, or HIV infected) increases the probability of getting an *pneumonia*. Also, *lung cancer* increases this probability. *Smoking* is a serious risk factor for *bronchitis* and for *lung cancer*.

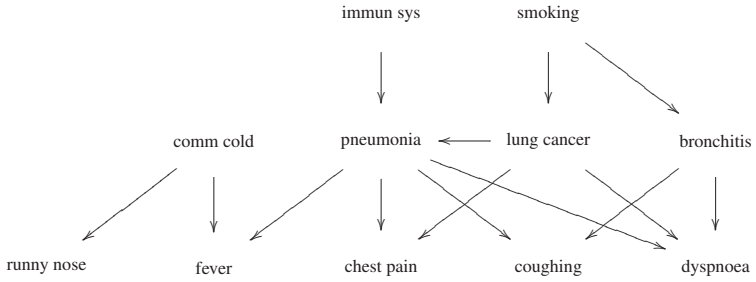

Fig. 3 DAG for the respiratory medicine toy model. See text for details

Now to build a model, we first have to find out which are the variables. In the text above, these are the ones printed in *italics*. In a realistic medical application, one may want to model multi-state variables. For simplicity, however, we take in this example all variables binary (true/false). Note that by modeling diseases as separate variables rather than by mutually exclusive states in a single disease variable, the model allows diseases to co-occur.

The next step is to figure out a sensible graphical structure. In the graphical representation of the model, i.e., in the DAG, all these variables are represented by nodes. The question now is which arrows to draw between the nodes. For this, we will use the principle of causal modeling. We derive these from the 'qualitative knowledge' and some common sense. The general causal modeling assumption in this medical domain is that risk factors 'cause' the diseases, so risk factors will point to diseases, and diseases 'cause' symptoms, so diseases will point to symptoms.

We start by modeling risk factors and diseases. Risk factors are *weakened immune-system* (for *pneumonia*), *smoking* (for *bronchitis* and for *lung cancer*), and *lung cancer* (also for *pneumonia*). The nodes for *weakened immune-system* and *smoking* have no incoming arrows, since there are no explicit causes for these variables in the model. We draw arrows from these node to the diseases for which they are risk factors. Furthermore, we have a node for the disease *common cold*. This node has no incoming arrow, since no risk factor for this variable is modeled.

Next we model the symptoms. The symptom *dyspnoea* may be due to the diseases *pneumonia*, *lung cancer*, and/or *bronchitis*, so we draw an arrow from all these diseases to *dyspnoea*. In a similar way, we can draw arrows from *pneumonia*, and *bronchitis* to *wet coughing*; arrows from *pneumonia*, and *lung cancer* to *chest pain*; arrows from *pneumonia* and *common cold* to *fever*; and an arrow from *common cold* to *runny nose*. This completes the DAG, which can be found in figure 3. (In the figures and in some of the text in the remainder of the section we abbreviated some of the variable names, e.g. we used *immun sys* instrad of *weakened immune-system*, etc.)

P(immun syst)
0.05

P(smoking)
0.3

P(common cold)
0.35

P(lung cancer)	smoking)
0.1	true
0.01	false

P(bronchitis	smoking)
0.3	true
0.01	false

P(runny nose	common cold)
0.9	true
0.01	false

P(pneumonia	immun syst,	lung cancer)
0.3	true	true
0.3	true	false
0.05	false	true
0.001	false	false

P(fever	pneumonia,	common cold)
0.9	true	true
0.9	true	false
0.2	false	true
0.01	false	false

P(cough	pneumonia,	bronchitis)
0.9	true	true
0.9	true	false
0.9	false	true
0.1	false	false

P(chest pain	pneumonia,	bronchitis)
0.9	true	true
0.9	true	false
0.9	false	true
0.1	false	false

P(dyspnoea	bronchitis,	lung cancer,	pneumonia)
0.8	true	true	true
0.8	true	true	false
0.8	true	false	true
0.8	true	false	false
0.5	false	true	true
0.5	false	true	false
0.5	false	false	true
0.1	false	false	false

Fig. 4 Conditional probability tables parametrizing the respiratory medicine toy model. The numbers in the nodes represent the marginal probabilities of the variables in state 'true'. See text for details

The next step is the quantitative part, i.e., the determination of the conditional probability tables. The numbers that we enter are rather arbitrary guesses and we do not pretend them to be anyhow realistic. In determining the conditional probabilities, we used some modeling assumptions such as that the probability of a symptom in the presence of an additional causing diseases is at least as high as the probability of that symptom in the absence of that disease. The tables as presented in figure 4. In these tables, the left column represents the probability values in the true state, $P(variablename) \equiv P(variablename = true)$, so $P(variablename = false) = 1 - P(variablename)$. The other columns indicate the joint states of the parent variables.

3.2 Reasoning

Now that the model is defined, we can use it for reasoning, e.g. by entering observational evidence into the system and doing inference, i.e. computing conditional probabilities given this evidence. To do the computation, we have modeled the system in BayesBuilder. In figure 5 we show a screen shot of the Bayesian network as modeled in BayesBuilder. The program uses the junction tree inference algorithm to compute the marginal node probabilities and displays them on screen. The marginal

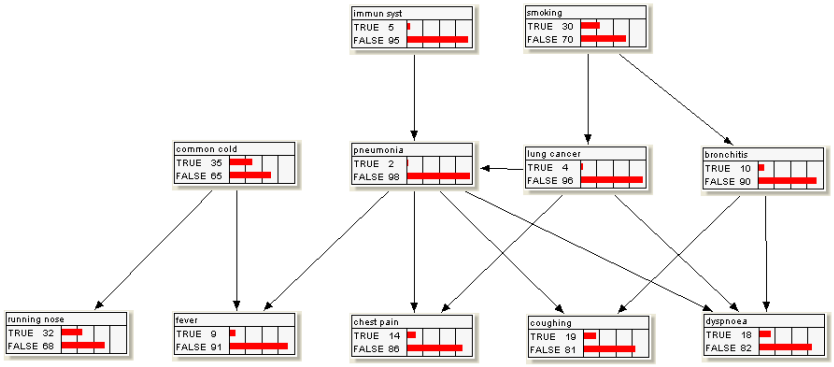

Fig. 5 Screen shot of the respiratory medicine toy model in the BayesBuilder GUI. Red bars present marginal node probabilities

node probabilities are the probability distributions of each of the variables in the absence of any additional evidence. In the program, evidence can be entered by clicking on a state of the variable. This procedure is sometimes called 'clamping'. The node probabilities will then be conditioned on the clamped evidence. With this, we can easily explore how the models reasons.

3.2.1 Knowledge representation

Bayesian networks may serve as a rich knowledge base. This is illustrated by considering a number of hypothetical medical guidelines and comparing these with Bayesian network inference results. These results will also serve to comment on some of the typical behavior in Bayesian networks.

1. In case of high *fever* in absence of a *runny nose*, one should consider *pneumonia*.

   Inference We clamp $fever = true$ and $runny\ nose = false$ and look at the conditional probabilities of the four diseases. We see that in particular the probability of pneumonia is increased from about 2% to 45%. See figure 6.

   Comment There are two causes in the model for *fever*, namely has parents *pneumonia* and *common cold*. However, the absence of a*common cold* makes *common cold* less likely. This makes the other explaining cause *pneumonia* more likely.

2. *Lung cancer* is often found in patients with *chest pain*, *dyspnoea*, no *fever*, and usually no *wet coughing*.

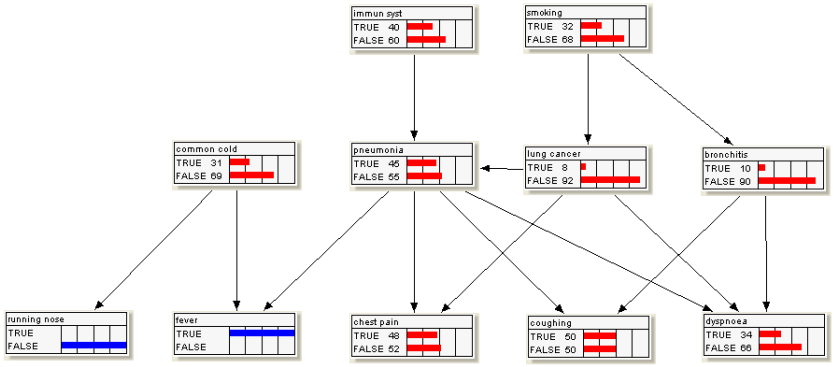

Fig. 6 Model in the state representing medical guideline 1, see main text. Red bars present conditional node probabilities, conditioned on the evidence (bleu bars)

Inference We clamp $chest\,pain = true$, $dyspnoea = true$, $fever = false$, and $coughing = false$ We see that probability of *lung cancer* is raised 0.57. Even if we set $coughing = true$, the probability is still as high as 0.47.

Comment *Chest pain* and *dyspnoea* can both be caused by *lung cancer*. However, *chest pain* for example, can also be caused by *pneumonia*. The absence of in particular *fever* makes *pneumonia* less likely and therefore *lung cancer* more likely. To a lesser extend this holds for absence of *coughing* and *bronchitis*.

3. *Bronchitis* and *lung cancer* are often accompanied, e.g patients with *bronchitis* often develop a *lung cancer* or vice versa. However, these diseases have no known causal relation, i.e., *bronchitis* is not a cause of *lung cancer*, and *lung cancer* is not a cause of *bronchitis*.

   Inference According to the model, $P(lung\ cancer|bronchitis = true) = 0.09$ and $P(bronchitis|lung\ cancer = true) = 0.25$. Both probabilities are more than twice the marginal probabilities (see figure 3).

   Comment Both diseases have the same common cause: *smoking*. If the state of smoking is observed, the correlation is broken.

3.2.2 Diagnostic reasoning

We can apply the system for diagnosis. the idea is to enter the patient observations, i.e. symptoms and risk factors into the system. Then diagnosis (i.e. finding the cause(s) of the symptoms) is done by Bayesian inference. In the following, we

present some hypothetical cases, present the inference results and comment on the reasoning by the network.

1. Mr. Appelflap calls. He lives with his wife and two children in a nice little house in the suburb. You know him well and you have good reasons to assume that he has no risk of a weakened immune system. Mr. Appelflap complains about high fever and a nasty wet cough (although he is a non-smoker). In addition, he sounds rather nasal. What is the diagnosis?

   Inference We clamp the risk factors *immun sys* $=$ *false*, *smoking* $=$ *false* and the symptoms *fever* $=$ *true*, *runny nose* $=$ *true*. We find all disease probabilities very small, except *common cold*, which is almost certainly true.

   Comment Due to the absence of risk factors, the prior probabilities of the other diseases that could explain the symptoms is very small compared to the prior probability of *common cold*. Since *common cold* also explains all the symptoms, that disease takes all the probability of the other causes. This phenomenon is called 'explaining away': pattern of reasoning in which the confirmation of one cause (*common cold*, with a high prior probability and confirmed by *runny nose*) of an observed event (*fever*) reduces the need to invoke alternative causes (*pneumonia* as an explanation of *fever*).

2. The salvation army calls. An unknown person (looking not very well) has arrived in their shelter for homeless people. This person has high fever, a nasty wet cough (and a runny nose.) What is the diagnosis?

   Inference We suspect a weakened immune system, so the system we clamp the risk factor *immun sys* $=$ *true*. As in the previous case, the symptoms are *fever* $=$ *true*,
   mboxrunnynose $=$ *true*. However, now we not only find *common cold* with a high probability ($P = 0.98$), but also *pneumonia* ($P = 0.91$).

   Comment Due to the fact that with a *weakened immune system*, the prior probability of *pneumonia* is almost as high as the prior probability of *common cold*. Therefore the conclusion is very different from the previous cas. Note that for this diagnosis, it is important that diseases can co-occur in the model.

3. A patient suffers from a recurrent pneumonia. This patient is a heavy smoker but otherwise leads a 'normal', healthy live, so you may assume there is no risk of a weakened immune system. What is your advice?

   Inference We clamp *immun sys* $=$ *false*, *smoking* $=$ *true*, and *pneumonia* $=$ *true*. As a result, we see that there is a high probability of *lung cancer*.

   Comment The reason is that due to *smoking*, the prior of disease is increased. More importantly, however, is that *weakened immune system* is excluded as cause of the *pneumonia*, so that *lung cancer* remains as the most likely explanation of the cause of the recurrent *pneumonia*.

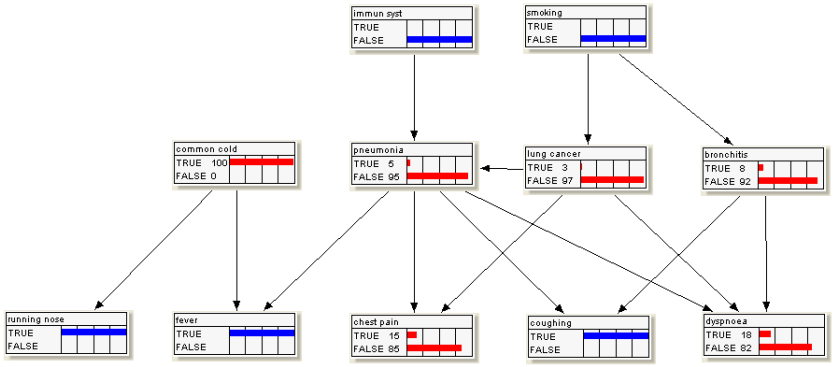

Fig. 7 Diagnosing mr. Appelflap. Primary diagnosis: *common cold*. See main text

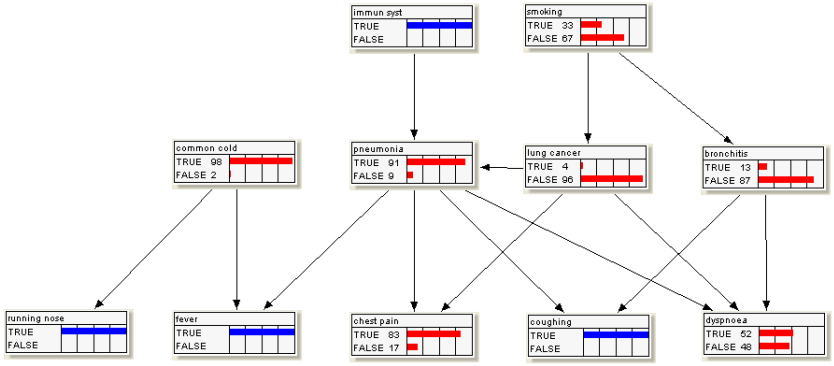

Fig. 8 Salvation army case. Primary diagnosis: *pneumonia*. See main text

3.3 Discussion

With the toy model, we aimed to illustrate the basic principles of Bayesian network modeling. With the inference examples, we have aimed to demonstrate some of typical reasoning capabilities of Bayesian networks. One features of Bayesian networks that distinguish them from e.g. conventional feedforward neural networks is that reasoning is in arbitrary direction, and with arbitrary evidence. Missing data or observations are dealt with in a natural way by probabilistic inference. In many applications, as well as in the examples in this section, the inference question is to compute conditional node probabilities. These are not the only quantities that one could compute in a Bayesian networks. Other examples are are correlations be-

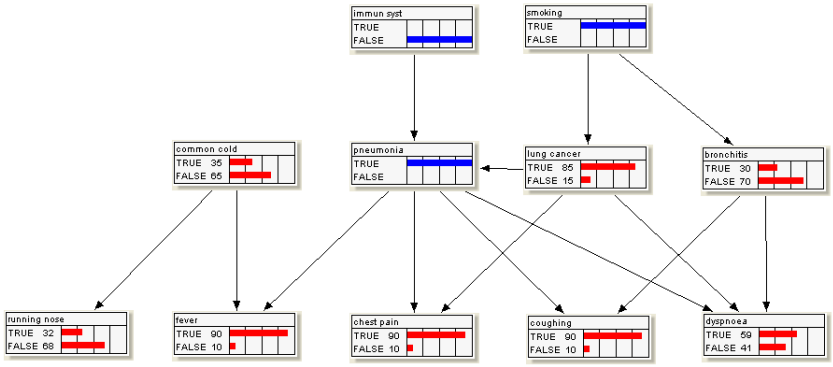

Fig. 9 Recurrent pneumonia case. Primary diagnosis: *lung cancer*. See main text

tween variables, the probability of the joint state of the the nodes, or the entropy of a conditional distribution. Applications of the latter two will be discussed in the next sections.

In the next sections we will discuss two Bayesian networks for real world applications The modeling principles are basically the same as in the toy model described in this section. There are some differences, however. In the first model, the network consists of a few types of nodes that have simple and well defined relations among each other. However, for each different case in the application, a different network has to be generated. It does not make sense for this application to try to build these networks beforehand in a GUI. In the second one the complexity is more in the variables themselves than in the network structure. Dedicated software has been written for both modeling and inference.

4 Bonaparte: a Bayesian Network for Disaster Victim Identification

Society is increasingly aware of the possibility of a mass disaster. Recent examples are the WTC attacks, the tsunami, and various airplane crashes. In such an event, the recovery and identification of the remains of the victims is of great importance, both for humanitarian as well as legal reasons. Disaster victim identification (DVI), i.e., the identification of victims of a mass disaster, is greatly facilitated by the advent of modern DNA technology. In forensic laboratories, DNA profiles can be recorded from small samples of body remains which may otherwise be unidentifiable. The identification task is the match of the unidentified victim with a reported missing person. This is often complicated by the fact that the match has to be made in an indirect way. This is the case when there is no reliable reference material of the

missing person. In such a case, DNA profiles can be taken from relatives. Since their profiles are statistically related to the profile of the missing person (first degree family members share about 50% of their DNA) an indirect match can be made.

In cases with one victim, identification is a reasonable straightforward task for forensic researchers. In the case of a few victims, the puzzle to match the victims and the missing persons is often still doable by hand, using a spread sheet, or with software tools available on the internet [10]. However, large scale DVI is infeasible in this way and an automated routine is almost indispensable for forensic institutes that need to be prepared for DVI.

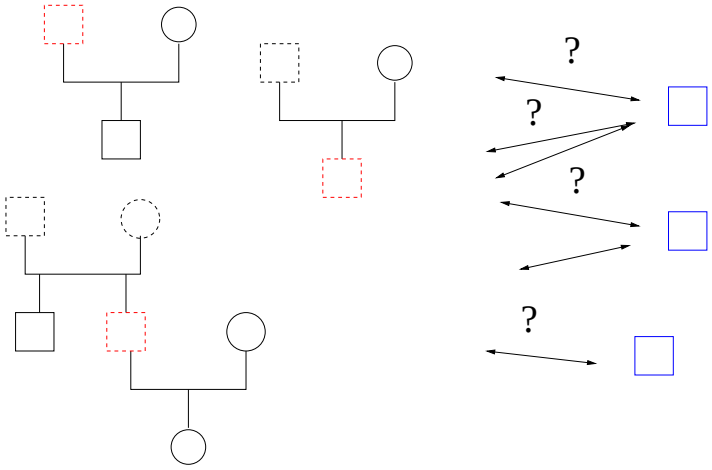

Fig. 10 The matching problem. Match the unidentified victims (blue, right) with reported missing persons (red, left) based on DNA profiles of victims and relatives of missing persons. DNA profiles are available from individuals represented by solid squares (males) and circles (females).

Bayesian networks are very well suited to model the statistical relations of genetic material of relatives in a pedigree [12]. They can directly be applied in kinship analysis with any type of pedigree of relatives of the missing persons. An additional advantage of a Bayesian network approach is that it makes the analysis tool more transparent and flexible, allowing to incorporate other factors that play a role — such as measurement error probability, missing data, statistics of more advanced genetic markers etc.

Recently, we have developed software for DVI, called Bonaparte. This development is in collaboration with NFI (Netherlands Forensic Institute). The computational engine of Bonaparte uses automatically generated Bayesian networks and Bayesian inference methods, enabling to correctly do kinship analysis on the basis of DNA profiles combined with pedigree information. It is designed to handle large scale events, with hundreds of victims and missing persons. In addition, it has graphical user interface, including a pedigree editor, for forensic analysts. Data-interfaces to other laboratory systems (e.g., for the DNA-data input) will also be implemented.

In the remainder of this section we will describe the Bayesian model approach that has been taken in the development of the application. We formulate the computational task, which is the computation of the likelihood ratio of two hypotheses. The main ingredient is a probabilistic model P of DNA profiles. Before discussing the model, we will first provide a brief introduction to DNA profiles. In the last part of the section we describe how P is modeled as a Bayesian network, and how the likelihood ratio is computed.

4.1 Likelihood Ratio of Two Hypotheses

Assume we have a pedigree with an individual MP who is missing (the Missing Person). In this pedigree, there are some family members that have provided DNA material, yielding the profiles. Furthermore there is an Unidentified Individual UI, whose DNA is also profiled. The question is, is $UI = MP$? To proceed, we assume that we have a probabilistic model P for DNA evidence of family members in a pedigree. To compute the probability of this event, we need hypotheses to compare. The common choice is to formulate two hypotheses. The first is the hypothesis H_1 that indeed $UI = MP$. The alternative hypothesis H_0 is that UI is an unrelated person U. In both hypotheses we have two pedigrees: the first pedigree has MP and family members FAM as members. The second one has only U as member. To compare the hypotheses, we compute the likelihoods of the evidence from the DNA profiles under the two hypotheses,

- Under H_p, we assume that $MP = UI$. In this case, MP is observed and U is unobserved. The evidence is $E = \{DNA_{MP} + DNA_{FAM}\}$.
- Under H_d, we assume that $U = UI$. In this case, U is observed and MP is observed. The evidence is $E = \{DNA_U + DNA_{FAM}\}$.

Under the model P, the likelihood ratio of the two hypotheses is

$$LR = \frac{P(E|H_p)}{P(E|H_d)} \,. \tag{7}$$

If in addition a prior odds $P(H_p)/P(H_d)$ is given, the posterior odds $P(H_p|E)/P(H_d|E)$ follows directly from multiplication of the prior odds and likelihood ratio,

$$\frac{P(H_p|E)}{P(H_d|E)} = \frac{P(E|H_p)P(H_p)}{P(E|H_d)P(H_d)} \,. \tag{8}$$

4.2 DNA Profiles

In this subsection we provide a brief introduction on DNA profiles for kinship analysis. A comprehensive treatise can be found in e.g. [6]. In humans, DNA found in the

nucleus of the cell is packed on chromosomes. A normal human cell has 46 chromosomes, which can be organized in 23 pairs. From each pair of chromosomes, one copy is inherited from father and the other copy is inherited from mother. In 22 pairs, chromosomes are homologous, i.e., they have practically the same length and contain in general the same genes (functional functional elements of DNA). These are called the autosomal chromosomes. The remaining chromosome is the sex-chromosome. Males have an X and a Y chromosome. Females have two X chromosomes.

More than 99% of the DNA of any two humans of the general population is identical. Most DNA is therefore not useful for identification. However, there are well specified locations on chromosomes where there is variation in DNA among individuals. Such a variation is called a genetic marker. In genetics, the specified locations are called loci. A single location is a locus.

In forensic research, the short tandem repeat (STR) markers are currently most used. The reason is that they can be reliable determined from small amounts of body tissue. Another advantage is that they have a low mutation rate, which is important for kinship analysis. STR markers is a class of variations that occur when a pattern of two or more nucleotides is repeated. For example,

$$(CATG)_3 = CATGCATGCATG\,. \tag{9}$$

The number of repeats x (which is 3 in the example) is the variation among the population. Sometimes, there is a fractional repeat, e.g. $CATGCATGCATGCA$, this would be encoded with repeat number $x = 3.2$, since there are three repeats and two additional nucleotides. The possible values of x and their frequencies are well documented for the loci used in forensic research. These ranges and frequencies vary between loci. To some extend they vary among subpopulations of humans. The STR loci are standardized. The NFI uses CODIS (Combined DNA Index System) standard with 13 specific core STR loci, each on different autosomal chromosomes.

The collection of markers yields the DNA profile. Since chromosomes exist in pairs, a profile will consist of pairs of markers. For example in the CODIS standard, a full DNA profile will consist of 13 pairs (the following notation is not common standard)

$$\bar{\mathbf{x}} = ({}^{1}x^{1},{}^{1}x^{2}),({}^{2}x^{1},{}^{2}x^{2}),\ldots,({}^{13}x^{1},{}^{13}x^{2})\,, \tag{10}$$

in which each ${}^{\mu}x^{s}$ is a number of repeats at a well defined locus μ. However, since chromosomes exists in pairs, there will be two alleles ${}^{\mu}x^{1}$ and ${}^{\mu}x^{2}$ for each location, one paternal — on the chromosome inherited from father — and one maternal. Unfortunately, current DNA analysis methods cannot identify the phase of the alleles, i.e., whether an allele is paternal or maternal. This means that $({}^{\mu}x^{1},{}^{\mu}x^{2})$ cannot be distinguished from $({}^{\mu}x^{2},{}^{\mu}x^{1})$. In order to make the notation unique, we order the observed alleles of a locus such that ${}^{\mu}x^{1} \leq {}^{\mu}x^{2}$.

Chromosomes are inherited from parents. Each parent passes one copy of each pair of chromosomes to the child. For autosomal chromosomes there is no (known) preference which one is transmitted to the child. There is also no (known) correlation between the transmission of chromosomes from different pairs. Since chromo-

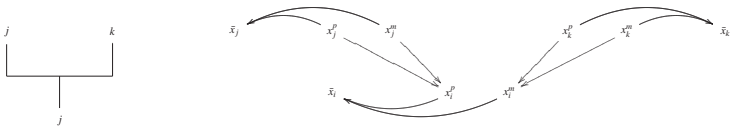

Fig. 11 A basic pedigree with father, mother, and child. Squares represent males, circles represent females. Right: corresponding Bayesian network. Grey nodes are observables. x_j^p and x_j^m represents paternal and maternal allele of individual j. See text.

somes are inherited from parents, alleles are inherited from parents as well. However, there is a small probability that an allele is changed or mutated. This mutation probability is about 0.1%.

Finally in the DNA analysis, sometimes failures occur in the DNA analysis method and an allele at a certain locus drops out. In such a case the observation is $({}^{\mu}x^1, F)$, in which "F" is a wild card.

4.3 A Bayesian Network for Kinship Analysis

In this subsection we will describe the building blocks of a Bayesian network to model probabilities of DNA profiles of individuals in a pedigree. First we observe that inheritance and observation of alleles at different loci are independent. So for each locus we can make an independent model P_μ. In the model description below, we will consider a model for a single locus, and we will suppress the μ dependency for notational convenience.

4.3.1 Allele Probabilities

We will consider pedigrees with individuals i. In a pedigree, each individual i has two parents, a father $f(i)$ and a mother $m(i)$. An exception is when a individual is a founder. In that case it has no parents in the pedigree.

Statistical relations between DNA profiles and alleles of family members can be constructed from the pedigree, combined with models for allele transmission . On the given locus, each individual i has a paternal allele x_i^f and an maternal allele x_i^m. f and m stands for 'father' and 'mother'. The pair of alleles is denoted as $x_i = (x_i^f, x_i^m)$. Sometimes we use superscript s which can have values $\{f, m\}$. So each allele in the pedigree is indexed by (i,s), where i runs over individuals and s over phases (f,m). The alleles can assume N values, where N as well as the allele values depend on the locus.

An allele from a founder is called 'founder allele'. So a founder in the pedigree has two founder alleles. The simplest model for founder alleles is to assume that they are independent, and each follow a distribution $P(a)$ of population frequencies.

This distribution is assumed to be given. In general $P(a)$ will depend on the locus. More advanced models have been proposed in which founder alleles are correlated. For instance, one could assume that founders in a pedigree come from a single but unknown subpopulation [1]. This model assumption yield corrections to the outcomes in models without correlations between founders. A drawback is that these models may lead to a severe increase in required memory and computation time. In this chapter we will restrict ourself to models with independent founder alleles.

If an individual i has its parents in the pedigree the allele distribution of an individual given the alleles of its parents are as follows,

$$P(x_i|x_{f(i)},x_{m(i)}) = P(x_i^f|x_{f(i)})P(x_i^m|x_{m(i)}) \,, \tag{11}$$

where

$$P(x_i^f|x_{f(i)}) = \frac{1}{2}\sum_{s=f,m} P(x_i^f|x_{f(i)}^s) \,, \tag{12}$$

$$P(x_i^m|x_{m(i)}) = \frac{1}{2}\sum_{s=f,m} P(x_i^m|x_{m(i)}^s) \,. \tag{13}$$

To explain (12) in words: individual i obtains its paternal allele x_i^f from its father $f(i)$. However, there is a 50% chance that this allele is the *paternal* allele $x_{f(i)}^f$ of father $f(i)$ and a 50% chance that it is his *maternal* allele $x_{f(i)}^m$. A similar explanation applies to (13).

The probabilities $P(x_i^f|x_{f(i)}^s)$ and $P(x_i^m|x_{m(i)}^s)$ are given by a mutation model $P(a|b)$, which encodes the probability that allele of the child is a while the allele on the parental chromosome that is transmitted is b. The precise mutation mechanisms for the different STR markers are not known. There is evidence that mutations from father to child are in general about 10 times as probable as mutations from mother to child. Gender of each individual is assumed to be known, but for notational convenience we suppress dependency of parent gender. In general, mutation tends to decrease with the difference in repeat numbers $|a-b|$. Mutation is also locus dependent [4].

Several mutation models have been proposed, see e.g. [8]. As we will see later, however, the inclusion of a detailed mutation model may lead to a severe increase in required memory and computation time. Since mutations are very rare, one could ask if there is any practical relevance in a detailed mutation model. The simplest mutation model is of course to assume the absence of mutations, $P(a|b) = \delta_{a,b}$. Such model enhances efficient inference. However, any mutation in any single locus would lead to a 100% rejection of the match, even if there is a 100% match in the remaining markers. Mutation models are important to get some model tolerance against such case. The simplest non-trivial mutation model is a uniform mutation model with mutation rate μ (not to be confused with the locus index μ),

$$P(a|a) = 1-\mu \ , \tag{14}$$
$$P(a|b) = \mu/(N-1) \quad \text{if } a \neq b \ . \tag{15}$$

Mutation rate may depend on locus and gender.

An advantage of this model is that the required memory and computation time increases only slightly compared to the mutation free model. Note that the population frequency is in general not invariant under this model: the mutation makes the frequency more flat. One could argue that this is a realistic property that introduces diversity in the population. In practical applications in the model, however, the same population frequency is assumed to apply to founders in different generations in a pedigree. This implies that if more unobserved references are included in the pedigree to model ancestors of an individual, the likelihood ratio will (slightly) change. In other words, formally equivalent pedigrees will give (slightly) different likelihood ratios.

4.3.2 Observations

Observations are denoted as $\bar{x}_i$, or $\bar{x}$ if we do not refer to an individual. The parental origin of an allele can not be observed, so alleles $x^f = a, x^m = b$ yields the same observation as $x^f = b, x^m = a$. We adopt the convention to write the smallest allele first in the observation: $\bar{x} = (a,b) \Leftrightarrow a \leq b$. In the case of an allele loss, we write $\bar{x} = (x,F)$ where F stands for a wild card. We assume that the event of an allele loss can be observed (e.g. via the peak hight [6]). This event is modeled by L. With $L = 1$ there is allele loss, and there will be a wild card ?. A full observation is coded as $L = 0$. The case of loss of two alleles is not modeled, since in that case we simply have no observation.

The observation model is now straightforwardly written down. Without allele loss ($L = 0$), alleles y results in an observation $\bar{y}$. This is modeled by the deterministic table

$$P(\bar{x}|y, L=0) = \begin{cases} 1 & \text{if } \bar{x} = \bar{y} \ , \\ 0 & \text{otherwise.} \end{cases} \tag{16}$$

Note that for a given y there is only one $\bar{x}$ with $\bar{x} = \bar{y}$.

With allele loss ($L = 1$), we have

$$P(\bar{x} = (a,F)|(a,b), L=1) = 1/2 \qquad \text{if } a \neq b \tag{17}$$
$$P(\bar{x} = (b,F)|(a,b), L=1) = 1/2 \qquad \text{if } a \neq b \tag{18}$$
$$P(\bar{x} = (a,F)|(a,a), L=1) = 1 \ . \tag{19}$$

I.e., if one allele is lost, the alleles (a,b) leads to an observation a (then b is lost), or to an observation b (then a is lost). Both events have 50% probability. If both alleles are the same, so the pair is (a,a), then of course a is observed with 100% probability.

4.4 Inference

By multiplying all allele priors, transmission probabilities and observation models, a Bayesian network of alleles x and DNA profiles of individuals $\bar{x}$ in a given pedigree is obtained. Assume that the pedigree consists of a set of individuals $\mathcal{I} = 1,\ldots,K$ with a subset of founders $\mathcal{F}$, and assume that allele losses L_j are given, then this probability reads

$$P(\{\bar{x},x\}_{\mathcal{I}}) = \prod_j P(\bar{x}_j|x_j,L_j) \prod_{i\in\mathcal{I}\setminus\mathcal{F}} P(x_i|x_{f(i)},x_{m(i)}) \prod_{i\in\mathcal{F}} P(x_i)\ . \tag{20}$$

Under this model the likelihood of a given set DNA profiles can now be computed. If we have observations $\bar{x}_j$ from a subset of individuals $j \in \mathcal{O}$, the likelihood of the observations in this pedigree is the marginal distribution $P(\{\bar{x}\}_{\mathcal{O}})$, which is the marginal probability

$$P(\{\bar{x}\}_{\mathcal{O}}) = \sum_{x_1}\ldots\sum_{x_K} \prod_{j\in\mathcal{O}} P(\bar{x}_j|x_j,L_j) \prod_{i\in\mathcal{I}\setminus\mathcal{F}} P(x_i|x_{f(i)},x_{m(i)}) \prod_{i\in\mathcal{F}} P(x_i)\ . \tag{21}$$

This computation involves the sum over all states of allele pairs x_i of all individuals.

In general, the allele-state space can be prohibitively large. This would make even the junction tree algorithm infeasible if it would straightforwardly be applied. Fortunately, a significant reduction in memory requirement can be achieved by "value abstraction": if the observed alleles in the pedigree are all in a subset A of M different allele values, we can abstract from all unobserved allele values and consider them as a single state z. If an allele is z, it means that it has a value that is not in the set of observed values A. We now have a system in which states can assume only $M+1$ values which is generally a lot smaller than N, the number of a priori possible allele values. This procedure is called value abstraction [14]. The procedure is applicable if for any $a \in A$, $L \in \{0,1\}$, and $b_1,b_2,b_3,b_4 \notin A$, the following equalities hold

$$P(a|b_1) = P(a|b_2) \tag{22}$$
$$P(\bar{x}|a,b_1,L) = P(\bar{x}|a,b_2,L) \tag{23}$$
$$P(\bar{x}|b_1,a,L) = P(\bar{x}|b_2,a,L) \tag{24}$$
$$P(\bar{x}|b_1,b_2,L) = P(\bar{x}|b_3,b_4,L) \tag{25}$$

If these equalities hold, then we can replace $P(a|b)$ by $P(a|z)$ and $P(\bar{x}|a,b)$ by $P(\bar{x}|a,z)$ etc. in the abstracted state representation. The conditional probability of z then follows from

$$P(z|x) = 1 - \sum_{a\in A} P(a|x) \tag{26}$$

for all x in $A \cup z$. One can also easily check that the observation probabilities satisfy the condition. The uniform mutation model satisfies condition (22) since $P(a|b) =$

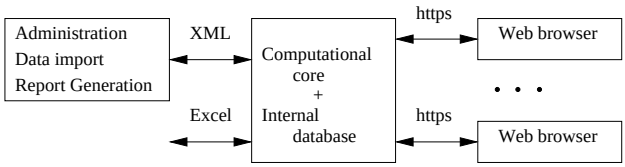

Fig. 12 Bonaparte's basic architecture

$\mu/(N-1)$ for any $a \in A$ and any $b \notin A$. Note that condition (22) does not necessarily holds for a general mutation model, so value abstraction could then not be applied.

Using value abstraction as a preprocessing step, a junction tree-based algorithm can straightforwardly applied to compute the desired likelihood. In this way, likelihoods and likelihood ratios are computed for all loci, and reported to the user.

4.5 The application

Bonaparte has been designed to facilitate large scale matching. The application has a multi-user client-server based architecture, see fig. 12. Its computational core and the internal database runs on a server. All match results are stored in internal database. Rewind to any point in back in time is possible. Via an XML and secure https interfaces, the server connects to other systems. Users can login via a web-browser so that no additional software is needed on the clients. The current version Bonaparte is now under user-validation. A live demo version will be made available on www.dnadvi.nl. The application is currently being deployed by the Netherlands Forensic Institute NFI. On 12 May 2010, Afriqiyah Airways Flight 8U771 crashed on landing near Tripoli International Airport. There were 103 victims. One child survived the crash. A large number of victims were blood-relatives. The Bonaparte program has been successfully used for the matching analysis to identify the victims. The program has two advantages compared to NFI's previous approach. Firstly, due to fully automated processing, the identification process has been significantly accelerated. Secondly, unlike the previous approach, the program does not need reference samples from first degree relatives since it processes whole pedigree information. For this accident, this was important since in some cases parents with children crashed together and for some individuals, no reference samples from living first degree relatives were available. Bonaparte could do the identification well with samples from relatives of higher degree.

4.6 Summary

Bonaparte is an application of Bayesian networks for victim identification by kinship analysis based on DNA profiles. The Bayesian networks are used to model statistical relations between DNA profiles of different individuals in a pedigree. By Bayesian inference, likelihood ratios and posterior odds of hypotheses are computed, which are the quantities of interest for the forensic researcher. The probabilistic relations between variables are based on first principles of genetics. A feature of this application is the automatic, on-the-fly derivation of models from data, i.e., the pedigree structure of a family of a missing person. The approach is related to the idea of modeling with templates, which is discussed in e.g. [20].

5 A Petrophysical Decision Support System

Oil and gas reservoirs are located in the earth's crust at depths of several kilometers, and when located offshore, in water depths of a few meters to a few kilometers. Consequently, the gathering of critical information such as the presence and type of hydrocarbons, size of the reservoir and the physical properties of the reservoir such as the porosity of the rock and the permeability is a key activity in the oil and gas industry.

Pre-development methods to gather information on the nature of the reservoirs range from gravimetric, 2D and 3D seismic to the drilling of exploration and appraisal boreholes. Additional information is obtained while a field is developed through data acquisition in new development wells drilled to produce hydrocarbons, time-lapse seismic surveys and in-well monitoring of how the actual production of hydrocarbons affects physical properties such as the pressure and temperature. The purpose of information gathering is to decide which reservoirs can be developed economically, and how to adapt the means of development best to the particular nature of a reservoir.

The early measurements acquired in exploration, appraisal and development boreholes are a crucial component of the information gathering process. These measurements are typically obtained from tools on the end of a wireline that are lowered into the borehole to measure the rock and fluid properties of the formation. Their is a vast range of possible measurement tools [28]. Some options are very expensive and may even risk other data acquisition options. In general acquiring all possible data imposes too great an economic burden on the exploration, appraisal and development. Hence data acquisition options must be exercised carefully bearing in mind the learnings of already acquired data and general hydrocarbon field knowledge. Also important is a clear understanding of what data can and cannot be acquired later and the consequences of having an incorrect understanding of the nature of a reservoir on the effectiveness of its development.

Making the right data acquisition decisions, as well as the best interpretation of information obtained in boreholes forms one of the principle tasks of petrophysi-

cists. The efficiency of a petrophysicist executing her/his task is substantially influenced by the ability to gauge her/his experience to the issues at hand. Efficiency is hampered when a petrophysicists experience level is not yet fully sufficient and by the rather common circumstance that decisions to acquire particular types of information or not must be made in a rush, at high costs and shortly after receiving other information that impact on that very same decision. Mistakes are not entirely uncommon and almost always painful. In some cases, non essential data is obtained at the expense of extremely high cost, or essential data is not obtained at all; causing development mistakes that can jeopardize the amount of hydrocarbon recoverable from a reservoir and induce significant cost increases.

The overall effectiveness of petrophysicists is expected to improve using a decision support system (DSS). In practice a DSS can increase the petrophysicists' awareness of low probability but high impact cases and alleviate some of the operational decision pressure.

In cooperation with Shell E&P, SNN has developed a DSS tool based on a Bayesian network and an efficient sampler for inference. The main tasks of the application is the estimation of compositional volume fractions in a reservoir on the basis of measurement data. In addition it provides insight in the effect of additional measurements. Besides an implementation of the model and the inference, the tool contains graphical user interface in which the user can take different views on the sampled probability distribution and on the effect of additional measurements.

In the remainder of this section, we will describe the Bayesian network approach for the DSS tool. We focus on our modeling and inference approach. More details are described in the full paper [5].

5.1 Probabilistic modeling

The primary aim of the model is to estimate the compositional volume fractions of a reservoir on the basis of borehole measurements. Due to incomplete knowledge, limited amount of measurements, and noise in the measurements, there will be uncertainty in the volume fractions. We will use Bayesian inference to deal with this uncertainty.

The starting point is a model for the probability distribution $P(\mathbf{v},\mathbf{m})$ of the compositional volume fractions $\mathbf{v}$ and borehole measurements $\mathbf{m}$. A causal argument *"The composition is given by the (unknown) volume fractions, and the volume fractions determine the distribution measurement outcomes of each of the tools"* leads us to a Bayesian network formulation of the probabilistic model,

$$P(\mathbf{v},\mathbf{m}) = \prod_{i=1}^{Z} P(m_i|\mathbf{v})P(\mathbf{v}) \; . \tag{27}$$

In this model, $P(\mathbf{v})$ is the so-called *prior*, the prior probability distribution of volume fractions before having seen any data. In principle, the prior encodes the generic ge-

ological and petrophysical knowledge and beliefs [30]. The factor $\prod_{i=1}^{Z} P(m_i|\mathbf{v})$ is the *observation model*. The observation model relates volume fractions $\mathbf{v}$ to measurement outcomes m_i of each of the Z tools i. The observation model assumes that *given* the underlying volume fractions, measurement outcomes of the different tools are independent. Each term in the observation model gives the probability density of observing outcome m_i for tool i given that the composition is $\mathbf{v}$. Now given a set of measurement outcomes $\mathbf{m}^o$ of a subset *Obs* of tools, the probability distribution of the volume fractions can be updated in a principled way by applying *Bayes' rule*,

$$P(\mathbf{v}|\mathbf{m}^o) = \frac{\prod_{i\in Obs} P(m_i^o|\mathbf{v})P(\mathbf{v})}{P(\mathbf{m}^o)}\,. \tag{28}$$

The updated distribution is called the *posterior* distribution. The constant in the denominator $P(\mathbf{m}^o) = \int_{\mathbf{v}} \prod_{i\in Obs} P(m_i^o|\mathbf{v})P(\mathbf{v})\mathrm{d}\mathbf{v}$ is called the *evidence*.

In our model, $\mathbf{v}$ is a 13 dimensional vector. Each component represents the volume fraction of one of 13 most common minerals and fluids (water, calcite, quartz, oil, etc.). So each component is bounded between zero and one. The components sum up to one. In other words, the volume fractions are confined to a simplex $\mathbb{S}^K = \{\mathbf{v}|0 \leq v_j \leq 1, \sum_k v_k = 1\}$. There are some additional physical constraints on the distribution of $\mathbf{v}$, for instance that the total amount of fluids should not exceed 40% of the total formation. The presence of more fluids would cause a collapse of the formation.

Each tool measurement gives a one-dimensional continuous value. The relation between composition and measurement outcome is well understood. Based on the physics of the tools, petrophysicists have expressed these relations in terms of deterministic functions $f_j(\mathbf{v})$ that provide the idealized noiseless measurement outcomes of tool j given the composition $\mathbf{v}$ [30]. In general, the functions f_j are nonlinear. For most tools, the noise process is also reasonably well understood — and can be described by either a Gaussian (additive noise) or a log-Gaussian (multiplicative noise) distribution.

A straightforward approach to model a Bayesian network would be to discretize the variables and create conditional probability tables for priors and conditional distributions. However, due to the dimensionality of the volume fraction vector, any reasonable discretization would result in an infeasible large state space of this variable. We therefore decided to remain in the continuous domain.

The remainder of this section describes the prior and observation model, as well as the approximate inference method to obtain the posterior.

5.2 The prior and the observation model

The model has two ingredients: the prior of the volume fractions $P(\mathbf{v})$ and the observation model $P(m_j|\mathbf{v})$.

There is not much detailed domain knowledge available about the prior distribution. Therefore we decided to model the prior using conveniently parametrized family of distributions. In our case, $\mathbf{v} \in \mathbb{S}^K$, this lead to the Dirichlet distribution [22, 3]

$$Dir(\mathbf{v}|\alpha,\mu) \propto \prod_{j=1}^{K} v_j^{\alpha\mu_j-1} \delta\left(1-\sum_{i=1}^{K} v_i\right) . \tag{29}$$

The two parameters $\alpha \in \mathbb{R}_+$ (precision) and $\mu \in \mathbb{S}^K$ (vector of means) can be used to fine-tune the prior to our liking. The delta function — which ensures that the simplex constraint holds — is put here for clarity, but is in fact redundant if the model is constraint to $\mathbf{v} \in \mathbb{S}^K$. Additional information, e.g. the fact that the amount of fluids may not exceed 40% of the volume fraction can be incorporated by multiplying the prior by a likelihood term $\Phi(\mathbf{v})$ expressing this fact. The resulting prior is of the form

$$P(\mathbf{v}) \propto \Phi(\mathbf{v}) Dir(\mathbf{v}|\alpha,\mu) . \tag{30}$$

The other ingredient in the Bayesian network are the observation models. For most tools, the noise process is reasonably well understood and can be reasonably well described by either a Gaussian (additive noise) or a log-Gaussian (multiplicative noise) distribution. In the model, measurements are modeled as a deterministic tool function plus noise,

$$m_j = f_j(\mathbf{v}) + \xi_j , \tag{31}$$

in which the functions f_j are the deterministic tool functions provided by domain experts. For tools where the noise is multiplicative, a log transform is applied to the tool functions f_j and the measurement outcomes m_j. A detailed description of these functions is beyond the scope of this paper. The noises ξ_j are Gaussian and have a tool specific variance σ_j^2. These variances have been provided by domain experts. So, the observational probability models can be written as

$$P(m_i|\mathbf{v}) \propto \exp\left(-\frac{(m_j - f_j(\mathbf{v}))^2}{2\sigma_j^2}\right) . \tag{32}$$

5.3 Bayesian Inference

The next step is given a set of observations $\{m_i^o\}$, $i \in \text{Obs}$, to compute the posterior distribution. If we were able to find an expression for the evidence term, i.e., for the marginal distribution of the observations $P(\mathbf{m}^o) = \int_{\mathbf{v}} \prod_{i\in\text{Obs}} P(m_i^o|\mathbf{v}) P(\mathbf{v}) \mathrm{d}\mathbf{v}$ then the posterior distribution (28) could be written in closed form and readily evaluated. Unfortunately $P(\mathbf{m}^o)$ is intractable and a closed-form expression does not exist. In order to obtain the desired compositional estimates we therefore have to resort to approximate inference methods. Pilot studies indicated that sampling methods gave the best performance.

The goal of any sampling procedure is to obtain a set of N samples $\{x_i\}$ that come from a given (but maybe intractable) distribution π. Using these samples we can approximate expectation values $\langle A \rangle$ of a function $A(x)$ according to

$$\langle A \rangle = \int_x A(x)\pi(x)\mathrm{d}x \approx \frac{1}{N}\sum_{i=1}^{N} A(x_i) \; . \tag{33}$$

For instance, if we take $A(x) = x$, the approximation of the mean $\langle x \rangle$ is the sample mean $\frac{1}{N}\sum_{i=1}^{N} x_i$.

An important class of sampling methods are the so-called Markov Chain Monte Carlo (MCMC) methods [22, 3]. In MCMC sampling a Markov chain is defined that has an equilibrium distribution π, in such a way that (33) gives a good approximation when applied to a sufficiently long chain $x_1, x_2, \ldots, x_N$. To make the chain independent of the initial state x_0, a burn-in period is often taken into account. This means that one ignores the first $M \ll N$ samples that come from intermediate distributions and begins storing the samples once the system has reached the equilibrium distribution π.

In our application we use the hybrid Monte Carlo (HMC) sampling algorithm [11, 22]. HMC is a powerful class of MCMC methods that are designed for problems with continuous state spaces, such as we consider in this section. HMC can in principle be applied to any noise model with a continuous probability density, so there is no restriction to Gaussian noise models. HMC uses Hamiltonian dynamics in combination with a Metropolis [23] acceptance procedure to find regions of higher probability. This leads to a more efficient sampler than a sampler that relies on random walk for phase space exploration. HMC also tends to mix more rapidly than the standard Metropolis Hastings algorithm. For details of the algorithm we refer to the literature [11, 22].

In our case, $\pi(\mathbf{v})$ is the posterior distribution $p(\mathbf{v}|m_i^o)$ in (28). The HMC sampler generates samples $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_N$ from this posterior distribution. Each of the N samples is a full K-dimensional vector of volume fractions constraint on $\mathbb{S}^K$. The number of samples is of the order of $N = 10^5$, which takes a few seconds on a standard PC. Figure 13 shows an example of a chain of 10 000 states generated by the sampler. For visual clarity, only two components of the vectors are plotted (quartz and dolomite). The plot illustrates the multivariate character of the method: for example, the traces shows that the volume fractions of the two minerals tend to be mutually exclusive: either 20% quartz, or 20% dolomite but generally not both. From the traces, all kind of statistics can be derived. As an example, the resulting one dimensional marginal distributions of the mineral volume fractions are plotted.

The performance of the method relies heavily on the quality of the sampler. Therefore we looked at the ability of the system to estimate the composition of a (synthetic) reservoir and the ability to reproduce the results. For this purpose, we set the composition to a certain value $\mathbf{v}^*$. We apply the observation model to generate measurements $\mathbf{m}^o$. Then we run HMC to obtain samples from the posterior $P(\mathbf{v}|\mathbf{m}^o)$. Consistency is assessed by comparing results of different runs to each other and by comparing them with the "ground truth" $\mathbf{v}^*$. Results of simulations

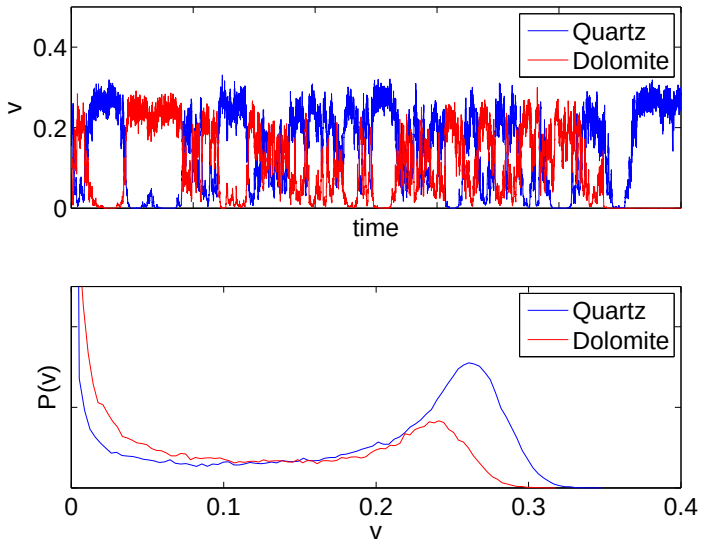

Fig. 13 Diagrams for quartz and dolomite. Top: time traces (10 000 time steps) of the volume fractions of quartz and dolomite. Bottom: Resulting marginal probability distributions of both fractions.

confirm that the sampler generates reproducible results, consistent with the underlying compositional vector [5]. In these simulations, we took the observation model to generate measurement data (the generating model) equal to the observation model that is used to compute the posterior (the inference model). We also performed simulations where they are different, in particular in their assumed variance. We found that the sampler is robust to cases where the variance of the generating model is smaller than the variance of the inference model. In the cases where the variance of the generating model is bigger, we found that the method is robust up to differences of a factor 10. After that we found that the sampler suffered severely from local minima, leading to irreproducible results.

5.4 Decision Support

Suppose that we have obtained a subset of measurement outcomes $\mathbf{m}^o$, yielding a distribution $P(\mathbf{v}|\mathbf{m}^o)$. One may subsequently ask the question which tool t should be deployed next in order to gain as much information as possible?

When asking this question, one is often interested in a specific subset of minerals and fluids. Here we assume this interest is actually in one specific component u. The question then reduces to selecting the most informative tool(s) t for a given mineral u.

We define the informativeness of a tool as the expected decrease of uncertainty in the distribution of v_u after obtaining a measurement with that tool. Usually, entropy is taken as a measure for uncertainty [22], so a measure of informativeness is the expected entropy of the distribution of v_u after measurement with tool t,

$$\begin{aligned}\langle \mathrm{H}_{u,t}|\mathbf{m}^o\rangle \equiv -\int P(m_t|\mathbf{m}^o)\int P(v_u|m_t,\mathbf{m}^o)\\ \times\log\left(P(v_u|m_t,\mathbf{m}^o)\right)\mathrm{d}v_u\mathrm{d}m_t\ .\end{aligned} \tag{34}$$

Note that the information of a tool depends on the earlier measurement results since the probabilities in (34) are conditioned on $\mathbf{m}^o$.

The most informative tool for mineral u is now identified as that tool t^* which yields in expectation the lowest entropy in the posterior distribution of v_u:

$$t^*_{u|\mathbf{m}^o} = \operatorname*{argmin}_t \langle \mathrm{H}_{u,t}|\mathbf{m}^o\rangle$$

In order to compute the expected conditional entropy using HMC sampling methods, we first rewrite the expected conditional entropy (34) in terms of quantities that are conditioned only on the measurement outcomes $\mathbf{m}^o$,

$$\begin{aligned}\langle \mathrm{H}_{u,t}|\mathbf{m}^o\rangle = -\int\int P(v_u,m_t|\mathbf{m}^o)\\ \times\log\left(P(v_u,m_t|\mathbf{m}^o)\right)\mathrm{d}v_u\mathrm{d}m_t\\ +\int P(m_t|\mathbf{m}^o)\int\log\left(P(m_t|\mathbf{m}^o)\right)\mathrm{d}m_t\ .\end{aligned} \tag{35}$$

Now the HMC run yields a set $V=\{v_1^j,v_2^j,\ldots,v_K^j\}$ of compositional samples (conditioned on $\mathbf{m}^o$). We augment these by a set $M=\{m_1^j=f_1(\mathbf{v}^j)+\xi_1^j,\ \ldots,\ m_Z^j=f_Z(\mathbf{v}^j)+\xi_Z^j\}$ of synthetic tool values generated from these samples (which are indexed by j) by applying equation (31). Subsequently, discretized joint probabilities $P(v_u,m_t|\mathbf{m}^o)$ are obtained via a two-dimensional binning procedure over v_u and m_t for each of the potential tools t. The binned versions of $P(v_u,m_t|\mathbf{m}^o)$ (and $P(m_t|\mathbf{m}^o)$) can be directly used to approximate the expected conditional entropy using a discretized version of equation (35).

The outcome of our implementation of the decision support tool is a ranking of tools according to the expected entropies of their posterior distributions. In this way, the user can select a tool based on a trade-off between expected information and other factors, such as deployment costs and feasibility.

5.5 The Application

The application is implemented in C++ as a stand alone version with a graphical user interface running on a Windows PC. The application has been validated by

petrophysical domain experts from Shell E&P. The further use by Shell of this application is beyond the scope of this chapter.

5.6 Summary

This chapter described a Bayesian network application for petrophysical decision support. The observation models are based on the physics of the measurement tools. The physical variables in this application are continuous-valued. A naive Bayesian network approach with discretized values would fail. We remained in the continuous domain and used the hybrid Monte Carlo algorithm for inference.

6 Discussion

Human decision makers are often confronted with highly complex domains. They have to deal with various sources of information and various sources of uncertainty. The quality of the decision is strongly influenced by the decision makers experience to correctly interpret the data at hand. Computerized decision support can help to improve the effectiveness of the decision maker by enhancing awareness and alerting the user to uncommon situations that may have high impact. Rationalizing the decision process may alleviate some of the decision pressure.

Bayesian networks are widely accepted as a principled methodology for modeling complex domains with uncertainty, in which different sources of information are to be combined, as needed in intelligent decision support systems. We have discussed in detail three applications of Bayesian networks. With these applications, we aimed to illustrate the modeling power of the Bayesian networks and to demonstrate that Bayesian networks can be applied in a wide variety of domains with different types of domain requirements. The medical model is a toy application illustrating the basic modeling approach and the typical reasoning behavior. The forensic and petrophysical models are real world applications, and show that Bayesian network technology can be applied beyond the basic modeling approach.

The chapter should be read as an introduction to Bayesian network modeling. There has been carried out much work in the field of Bayesian networks that is not covered in this chapter, e.g. the work on Bayesian learning [16], dynamical Bayesian networks [24], approximate inference in large, densely connected models [9, 25], templates and structure learning [20], nonparametric approaches [17, 13], etc.

Finally, we would like to stress that the Bayesian network technology is only one side of the model. The other side is the domain knowledge, which is maybe even more important for the model. Therefore Bayesian network modeling always requires a close collaboration with domain experts. And even then, the model is of course only one of many ingredients of an application, such as user-interface, data-

management, user-acceptance etc. which are all essential to make the application a success.

Acknowledgments

The presented work was partly carried out with support from the Intelligent Collaborative Information Systems (ICIS) project, supported by the Dutch Ministry of Economic Affairs, grant BSIK03024. We thank Ender Akay, Kees Albers and Martijn Leisink (SNN), Mirano Spalburg (Shell E & P), Carla van Dongen, Klaas Slooten and Martin Slagter (NFI) for their collaboration.

References

1. Balding, D., Nichols, R.: DNA profile match probability calculation: how to allow for population stratification, relatedness, database selection and single bands. Forensic Science International **64**(2-3), 125–140 (1994)
2. Beinlich, I., Suermondt, H., Chavez, R., Cooper, G., et al.: The ALARM monitoring system: A case study with two probabilistic inference techniques for belief networks. In: Proceedings of the Second European Conference on Artificial Intelligence in Medicine, vol. 256. Berlin: Springer-Verlag (1989)
3. Bishop, C.: Pattern recognition and machine learning. Springer (2006)
4. Brinkmann, B., Klintschar, M., Neuhuber, F., Hühne, J., Rolf, B.: Mutation rate in human microsatellites: influence of the structure and length of the tandem repeat. The American Journal of Human Genetics **62**(6), 1408–1415 (1998)
5. Burgers, W., Wiegerinck, W., Kappen, B., Spalburg, M.: A bayesian petrophysical decision support system for estimation of reservoir compositions. Expert Systems with Applications **37**(12), 7526 – 7532 (2010)
6. Butler, J.: Forensic DNA typing: biology, technology, and genetics of STR markers. Academic Press (2005)
7. Castillo, E., Gutierrez, J.M., Hadi, A.S.: Expert Systems and Probabilistic Network Models. Springer (1997)
8. Dawid, A., Mortera, J., Pascali, V.: Non-fatherhood or mutation? A probabilistic approach to parental exclusion in paternity testing. Forensic science international **124**(1), 55–61 (2001)
9. Doucet, A., Freitas, N.d., Gordon, N. (eds.): Sequential Monte Carlo Methods in Practice. Springer-Verlag, New York (2001)
10. Drábek, J.: Validation of software for calculating the likelihood ratio for parentage and kinship. Forensic Science International: Genetics **3**(2), 112–118 (2009)
11. Duane, S., Kennedy, A., Pendleton, B., Roweth, D.: Hybrid Monte Carlo Algorithm. Phys. Lett. B **195**, 216 (1987)
12. Fishelson, M., Geiger, D.: Exact genetic linkage computations for general pedigrees. Bioinformatics **18**(Suppl 1), S189–S198 (2002)
13. Freno, A., Trentin, E., Gori, M.: Kernel-based hybrid random fields for nonparametric density estimation. In: European Conference on Artificial Intelligence (ECAI), vol. 19, pp. 427–432 (2010)
14. Friedman, N., Geiger, D., Lotner, N.: Likelihood computations using value abstraction. In: Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence, pp. 192–200. Morgan Kaufmann Publishers (2000)
15. Heckerman, D.: Probabilistic interpretations for mycin's certainty factors. In: L. Kanal, J. Lemmer (eds.) Uncertainty in artificial intelligence, pp. 167–96. North Holland (1986)

16. Heckerman, D.: A tutorial on learning with bayesian networks. In: Innovations in Bayesian Networks, *Studies in Computational Intelligence*, vol. 156, pp. 33–82. Springer Berlin / Heidelberg (2008)
17. Hofmann, R., Tresp, V.: Discovering structure in continuous variables using bayesian networks. In: Advances in Neural Information Processing Systems (NIPS), vol. 8, pp. 500–506 (1995)
18. Jensen, F.: An Introduction to Bayesian networks. UCL Press (1996)
19. Jordan, M.: Learning in graphical models. Kluwer Academic Publishers (1998)
20. Koller, D., Friedman, N.: Probabilistic Graphical Models: Principles and Techniques. The MIT Press (2009)
21. Lauritzen, S., Spiegelhalter, D.: Local computations with probabilities on graphical structures and their application to expert systems. Journal of the Royal Statistical Society. Series B (Methodological) pp. 157–224 (1988)
22. MacKay, D.: Information theory, inference and learning algorithms. Cambridge University Press (2003)
23. Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A., Teller, E.: Equation of state calculations by fast computing machines. The journal of chemical physics **21**(6), 1087 (1953)
24. Murphy, K.: "dynamic bayesian networks: Representation, inference and learning". Ph.D. thesis, UC Berkeley (2002)
25. Murphy, K.P., Weiss, Y., Jordan, M.I.: Loopy belief propagation for approximate inference: An empirical study. In: Proceedings of Uncertainty in AI, pp. 467–475 (1999)
26. Pearl, J.: Probabilistic Reasoning in Intelligent systems: Networks of Plausible Inference. Morgan Kaufmann Publishers, Inc. (1988)
27. Russell, S., Norvig, P., Canny, J., Malik, J., Edwards, D.: Artificial intelligence: a modern approach. Prentice Hall (2003)
28. Schlumberger: Log Interpretation Principles/Applications. Schlumberger Limited (1991)
29. Shortliffe, E., Buchanan, B.: A model of inexact reasoning in medicine. Mathematical Biosciences **23**(3-4), 351–379 (1975)
30. Spalburg, M.: Bayesian uncertainty reduction for log evaluation. SPE International (2004). SPE88685