

## 6.2.11 BELIEF NETWORKS/BAYESIAN NETWORKS

Wim Wiegerink<sup>1</sup>, Tom Heskes<sup>2</sup>

### INTRODUCTION

In modeling real world tasks, one inevitably has to deal with uncertainty. This uncertainty is due to the fact that many facts are unknown and or simply ignored and summarized. Suppose that one morning you find out that your grass is wet. Is it due to rain, or is it due to the sprinkler? If there is no other information, you can only talk in terms of probabilities. In a probabilistic model approach, you could try to enumerate the states of all variables (grass: wet or dry; rained: true or false; sprinkler: on or off), and assign probabilities to each combination of states. Ideally, these probabilities will be proportional to the relative frequencies of the occurrence of the combinations of states.

The elegance of the probabilistic approach resides in the fact that the probabilistic model on these three variables is correct, consistent and automatically includes context dependency. For instance, you can use the model to compute the probability that it has rained in the context that the grass is wet. You will find an increase in the probability that it has rained. However, in the context that the grass is wet *and* that the sprinkler has been left on, the probability that it has rained is generally (for sensible choices of the conditional probabilities) lower. In systems that are rule-based rather than based on probability theory context dependency is not fully modeled. In such systems invalid conclusions can be drawn easily. For instance, in a system with context free rules, concatenation of the rule: ‘sprinkler on’ implies ‘wet grass’ with the rule: ‘wet grass’ implies ‘it has rained’, will lead to the incorrect conclusion that ‘sprinkler on’ implies ‘it has rained’.

A drawback of probabilistic models is their computational complexity. In problems with many variables the approach in which all combinations of states are enumerated in the model will lead to huge computational problems. The reason is that the number of combinations of states grows exponentially with the number of variables. Even if one manages to parameterize the probabilities in an efficient way, the problem is still not easily solved: inference (i.e. computing probabilities of variables of interest) requires the summation over all (exponentially many) states of the remaining variables.

Graphical models provide a remedy. They include Bayesian networks (also known as belief networks), Hidden Markov models, Markov fields, naive Bayes and many others. In graphical models, the probability distribution is defined in terms of local quantities, involving only a few variables. In particular, the local structure can be represented by a graph; hence the name graphical model. The local quantities are glued together according to the laws of probability theory, such that they define a unique and consistent global probability distribution.

---

<sup>1</sup> Dr T. Heskes,  
tom@mbfys.kun.nl, SNN,  
Nijmegen University, Nijmegen,  
The Netherlands

<sup>2</sup> Dr W. Wiegerink,  
wimw@mbfys.kun.nl, SNN,  
Nijmegen University, Nijmegen,  
The Netherlands  
<http://www.snn.kun.nl/>

In graphical models the simplifying assumption is that variables that are not directly connected in the graph are (conditionally) independent. Although this may seem a severe restriction, it is in many cases a quite natural and intuitive one. Suppose that we want to extend the wet grass model with a variable representing the neighbor's grass. Now in general, the states of the neighbor's grass and your own grass are dependent. The reason is simply that, if your grass is wet, it probably has rained, and thus your neighbor's grass will also be wet. Natural assumptions, however, are that given the state of 'rained', the states of both grasses are independent, and in the same context that the state of your neighbor's grass is independent of the state of your sprinkler. Note that if you do not know the state of 'rained', the fact that your sprinkler has been on may imply a reduced probability of 'rained', thus a reduced probability of your neighbor's grass being wet. Again, this is an example of the context dependency of probabilistic models, referred to as 'explaining away' (see also the example below). The (conditional) independencies in graphical models do not only simplify the representation of these models, they are also exploited by efficient inference algorithms. This facilitates the practical usage of graphical models. It should be stressed that these inference algorithms are exact according to the laws of probability theory. In large, complex networks, however, even these algorithms may become computationally too demanding. In such cases, approximate inference methods such as stochastic sampling are needed.

A Bayesian network is a particular type of graphical model, frequently used in applications of artificial intelligence for building probabilistic expert systems. An appealing feature of Bayesian networks is that their graphical structure can often be loosely interpreted as the result of direct causal relations between variables. In domains with lots of causal relations, such as medical diagnosis (diseases cause symptoms), human experts are usually able to express their domain knowledge in the graphical structure of the network. The parameters of the network are the conditional probabilities of effects given the state of their direct causes. Bayes' rule

$$P(\text{cause}|\text{effect}) = \frac{P(\text{effect}|\text{cause})P(\text{cause})}{P(\text{effect})} ,$$

arises when we want to reason from symptom (effect) to disease (cause) and thus have to 'invert' the probabilities.

The exact values of the conditional probabilities are often more difficult for human experts to determine. Fortunately, these conditional probabilities are relatively easy to estimate from data: in its simplest version learning in Bayesian networks (given their graphical structure) amounts to straightforward frequency counting. If the experts are not able to specify the graphical structure, there is the real challenge of learning the structure of Bayesian networks from data.

## BAYESIAN NETWORKS IN MORE DETAILS

### Bayesian networks and probability theory

The mathematics of Bayesian networks is most easily explained through an example. So let us consider the wet-grass example with four variables  $R$  (Rained),  $S$  (Sprinkler being on),  $G$  (Grass wet) and  $N$  (Neighbors grass wet). Each variable can be in two states, true or false. The joint probability distribution  $P(R, S, G, N)$  is a table with 16 entries. The table is normalized, i.e.

$$\sum_{R=\{t,f\}, \dots, N=\{t,f\}} P(R, S, G, N) = 1$$

With this table we can compute any variable of interest, e.g.  $P(R=f | N=t)$ , the probability that it has not rained given that my neighbor's grass is wet, reads

$$P(R=f | N=t) = \frac{P(R=f, N=t)}{P(N=t)} = \frac{\sum_{S=\{t,f\}, G=\{t,f\}} P(R=f, S, G, N=t)}{\sum_{R=\{t,f\}, S=\{t,f\}, G=\{t,f\}} P(R, S, G, N=t)}$$

Now according to probability theory, we can write any joint probability distribution as a product of conditional distributions:

$$P(R, S, G, N) = P(R)P(S|R)P(G|R, S)P(N|R, S, G).$$

In a Bayesian network, conditional independencies are assumed. For instance, we may assume for our model that being given (only) the state of Rained does not influence probability of the Sprinkler being on (an independency not yet discussed in the introductory section),

$$P(S|R) = P(S)$$

and given the state of Rained, the additional knowledge of the state of sprinkler and or the state of your grass will not influence the probability of the neighbor's grass being wet,

$$P(N|R, S, G) = P(N|R)$$

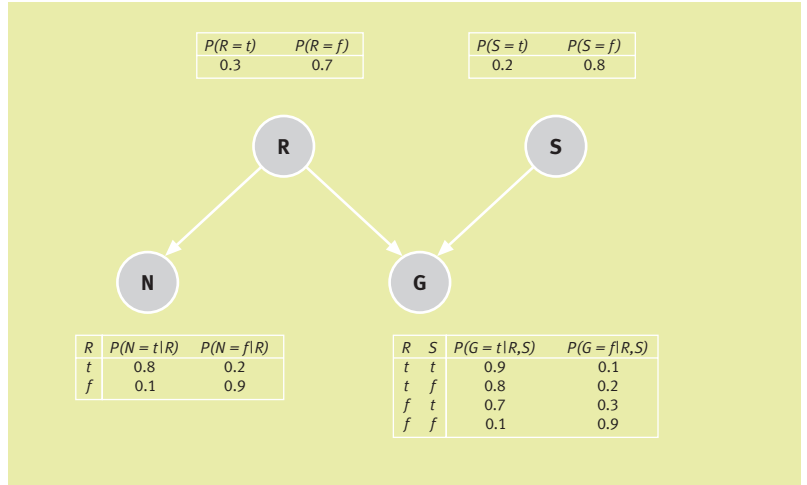
With these model assumptions (which may be completely wrong, but that is another issue), the joint probability in our Bayesian network is of the form

$$P(R, S, G, N) = P(R)P(S)P(G|R, S)P(N|R)$$

In Figure 1, the corresponding (directed acyclic) graph or DAG is visualized. Each variable is represented as a node. Each combination of a node and its incoming

**Figure 1**

Graphical representation and conditional probability tables for the 'wet grass' model.



arrows corresponds to a conditional probability distribution. For example, the node  $G$  with incoming arrows from  $R$  and  $S$  corresponds to the distribution  $P(G|R,S)$ . It can be shown that there is a 1-1 correspondence between DAGs and factorizations of the joint distribution into conditionals. If there is no conditional independence assumption, the graph is fully connected. Independence assumptions lead to deletion of arrows.

Furthermore, it is clear that if we define the conditional probabilities on the right hand side of equation (1), the joint probability is fully specified. The conditional probability distributions that we need to complete the definition of our model are given in Figure 1.

Our Bayesian network is now ready for carrying out inference. For instance, we can compute the two marginal probabilities of the grasses  $G$  and  $N$  being wet:

$$P(G=t) = \sum_{R=\{t,f\}, S=\{t,f\}} P(R)P(S)P(G=t|R,S) = 0.4$$

$$P(N=t) = \sum_{R=\{t,f\}} P(R)P(N=t|R) = 0.31$$

If we know that it has rained we can compute how these probabilities change through conditioning:

$$P(G=t|R=t) = \sum_{S=\{t,f\}} P(S)P(G=t|R=t,S) = 0.82$$

$$P(N=t|R=t) = \frac{P(S=t, G=t)}{P(G=t)} = 0.8$$

So, given that it has rained, the probabilities of both grasses being wet increase, as they should. On the other hand, if it is found that your grass is wet,

we can compute the probability that it has rained, as well as the probability that the sprinkler has been on.

$$P(R=t|G=t) = \frac{P(R=t, G=t)}{P(G=t)} = 0.615 ,$$

$$P(S=t|G=t) = \frac{P(S=t, G=t)}{P(G=t)} = 0.38 .$$

So, both probabilities are higher than their a priori probabilities  $P(R=t) = 0.3$  and  $P(S=t) = 0.2$ , respectively. In other words, the probabilities of both causes  $R$  and  $S$  increase, if their effect  $G$  is found to be true.

If it is observed now that the neighbor's grass is also wet, we obtain the probabilities

$$P(R=t|G=t, N=t) = \frac{P(R=t, G=t, N=t)}{P(G=t, N=t)} = 0.9274 ,$$

$$P(S=t|G=t, N=t) = \frac{P(S=t, G=t, N=t)}{P(G=t, N=t)} = 0.2498 .$$

Clearly, the possible cause  $R$  becomes more likely since additional evidence  $N=t$  is found. Since the increased probability of  $R$  provides an explanation of  $G=t$ , we no longer need the explanation provided by the other cause, which therefore receives a lower probability. It is said that  $S$  is 'explained away'.

### Bayesian networks and causal modeling?

A very important modeling issue in Bayesian networks is the direction of the arrows. It is instructive to consider what the consequences in the model are, if we reverse the incoming arrows to  $G$ . One could say: (1) wet grass indicates rain, therefore there should be an arrow from  $G$  to  $R$  and (2) wet grass indicates that the sprinkler has been on, therefore there should be an arrow from  $G$  to  $R$ . This model would lead to similar insensible conclusions as the concatenation of deterministic rules in the introductory section: if it has rained, the grass is probably wet, and this leads to an increased probability of the sprinkler being on.

In a similar way, it is instructive to consider the consequences of reversing the outgoing arrows from  $R$ , i.e. such that arrows point from  $N$  to  $R$  and from  $G$  to  $R$ . Given that it has rained, the probabilities of  $N$  and  $G$  both increase. Now suppose that you next observe that the neighbor's grass is wet. Then according to the model, the neighbor's wet grass explains the rain, and therefore the probability of your grass being wet decreases! This does definitely not correspond to common sense and therefore we deem the original model, with arrows from cause to effect, more sensible than the one with reversed arrows.

As we have seen, in general it is a good rule of thumb to construct a Bayesian

network from cause to effect. You start with nodes that represent independent root causes, then model the nodes they influence, and so on until you end at the leaves, i.e. the nodes that have no direct influence on other nodes. For this procedure, it is often useful to have a ‘story’ in mind.

Sometimes this procedure fails, because it is too difficult to tell what is cause and what is effect. Is someone’s behavior a result of his environment, or is the environment a reaction to his behavior? In such a case, you should 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.

In practice it is often difficult to decide whether two nodes are really (conditionally) independent. Usually, this is a matter of simplifying model assumptions. In the case of the wet grass model, one could easily argue that  $N$  and  $G$  are still dependent, even if we know that it has not rained, e.g. due to humidity or other weather conditions that increase the probability of both grasses being wet simultaneously. 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.

### SOFTWARE AND FURTHER READING

SNN Nijmegen has developed a freeware tool for building and computing with Bayesian networks, called BayesBuilder, available for both Windows and Linux platforms, which is included on the CD. Its most recent version can be downloaded from the site <http://www.snn.kun.nl/nijmegen/bayesbuilder.html>. The reference work on Bayesian networks is the book by [Pearl, 1988]. [Cowell, 1999] is more up to date and somewhat less technical. The tutorial by [Heckerman, 1998] specifically treats learning in Bayesian networks. The collection ‘Learning in Graphical Models’ as a whole gives an impression of the state of the art and current challenges. Lots of information can be found on the internet, with <http://www.auai.org>, the homepage of the Association for Uncertainty in Artificial Intelligence, as a good starting point. See also Section 2.3.2, Decision support for medical diagnosis.

### REFERENCES

- Cowell, R., A. Dawid, S. Lauritzen, D. Spiegelhalter. (1999). Probabilistic Networks and Expert Systems. Springer Verlag, Berlin
- Heckerman, D. (1998). A Tutorial on Learning with Bayesian Networks. In: M. Jordan (ed.). Learning in Graphical Models **89** of NATO ASI, series D. Behavioural and Social Sciences: 301-354. Kluwer
- Pearl, J. (1988). Probabilistic Reasoning in Intelligent systems: Networks of Plausible Inference. Morgan Kaufmann, San Francisco, CA