BAYESIAN NETWORKS AND INFLUENCE DIAGRAMS

Finn V. Jensen
Department of Mathematics and Computer Science
Aalborg University
Fredrik Bajers Vej 7
DK-9220 Aalborg Ø
Denmark

Summary

Bayesian networks are graphical probabilistic models consisting of variables and cause-effect relations between them. First Bayesian networks are defined and a couple of examples are given. It is illustrated how Bayesian networks are used for various task like calculating updated probabilities for variables given evidens, calculating probabilities for specific configurations of variables, calculating configurations of maximal probability and analysis for conflicting evidens. Influence diagrams are Bayesian networks argumented with special variables for actions. They are used for calculating optimal strategies for sequences of actions. It is finally shown how the problem of deciding between various information sources can be solved in situations with only one set of action options. These and other issues are treated more deeply in Jensen (1996).

1 Introduction

Expert systems are systems which repeatedly give advice to an expert on decision making on a series of cases with a common structure.

A simple view of the tasks of the expert is the triangle in Figure 1. First of all she observes her part of the world to establish the state of it.

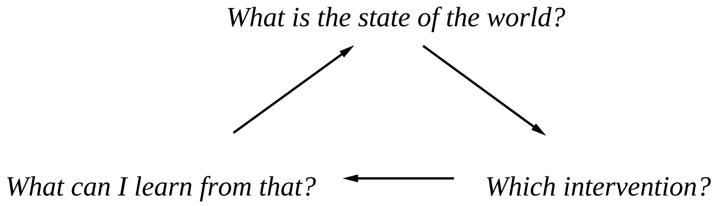

Figure 1: The task-triangle of an expert.

Based on her interpretation of the state of the world, the expert decides for an action. An action is an intervention in her part of the world. To any action the expert has some expectations. Sometimes they come true and sometimes they don't; but in any case she will learn from the results of the actions which may help her interpreting the world in future.

The first expert systems were constructed in the late 60's. The building blocks for the systems were *production rules.* A production rule is of the form

if *condition* **then** (*fact or action*)

where the condition is a logical expression. Though the language is very simple, it turned out to be rather powerful when modelling expert's reasoning, and several impressive rule based expert systems were constructed (for example MYCIN (Shortliffe 1976) and R1 (McDermott 1984).)

Rather soon after their first successes it became clear that rule based systems have their shortcomings. One of the major problems was how to treat uncertainty. Various uncertainty calculi where proposed, but all of them have serious shortcomings with respect to coherent inference.

Normative systems is an alternative to rule based expert systems. Both types of systems deal with repeated decision making, but instead of using a non-coherent uncertainty calculus tailored for rules, they use probability calculus and decision theory. Already in the 60's attempts were made to use probability theory in expert systems (Gorry & Barnett 1968). However, due to the very heavy calculation load required it was given up and considered an intractable task (Gorry 1973).

In the mid 80's the principles got a revival. Work by Pearl (1986) introduced Bayesian networks to expert systems, work by Lauritzen & Spiegelhalter (1988) and Jensen, Olesen & Andersen (1990) gave very efficient calculation methods, and with the MUNIN system (Andreassen, Jensen, Andersen, Falck, Kjærulff, Woldbye, Sørensen, Rosenfalck & Jensen 1989) it was demonstrated that the necessary calculations for very large networks are indeed tractable.

2 Bayesian networks

The basic knowledge when reasoning under uncertainty is whether information on some event influences your belief in other events.

2.1 Wet grass

Mr Homes leaves his house in the morning and notices that his grass is wet. He reasons: "I think it has been raining tonight. Then my neighbour, dr. Watson's grass is most probably wet also". That is, the information that Holmes' grass is wet has an influence on his belief of the status of Watson's grass. Now, assume that Holmes has checked the rain meter, and it was dry. Then he will not reason as above, and information on Holmes' grass has no influence on his belief about Watson's grass.

Next, consider two possible causes for wet grass. Besides rain, Holmes may have forgot to turn off the sprinkler. Assume that Mr. Holmes the next morning again notices that his grass is wet. Holmes' belief of both rain and sprinkler increases. Now he observes that Watson's grass is wet, and he concludes, that it has rained tonight. The next step in the reasoning is hard for machines but natural for human beings, namely explaining away: Holmes' wet grass has been explained by the rain, and thus there is no reason anymore to believe that the sprinkler has been on. Hence Holmes' belief of sprinkler is reduced to (almost) its initial size.

2.2 Causal networks

The situations above can be described by a graph. The events are nodes, and two nodes A and B are connected by a directed link from A to B if A has a causal impact on B. Figure 2 is a graphical model for Holmes' small world of wet grass.

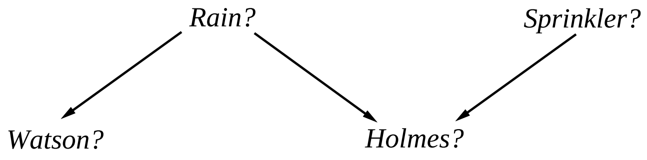

Figure 2: A network model for the wet grass example. Rain and sprinkler are causes of Holmes' grass being wet. Only rain can cause Watson's grass to be wet.

Figure 2 is an example of a causal network. A causal network consists of a set of *variables* and a set of *directed links* between variables. Mathematically the structure is called a directed graph. When talking about the relations in a directed graph we use the wording of family relations: If there is a link from A to B we say that B is a *child* of A, and A is a *parent* of B.

The variables represent events (propositions). In Figure 2 each variable has the states *yes* and *no* reflecting whether a certain event had taken place or not. In general a variable can have any number of states. A variable may for example be the colour of a car (states $blue, green, red, brown$), the number of children in a family (states $0, 1, 2, 3, 4, 5, 6, > 6$), diseases (states *bronchitis, tuberculosis, lung cancer*). Variables may have a countable or a continuous state-set, but in this article we solely consider variables with a finite number of states.

In a causal network a variable represents a set of possible states of affairs. A variable is in exactly one of its states; which one may be unknown to us.

2.3 Probability calculus

So far nothing has been said about the quantitative part of certainty assessment. Various certainty calculi on causal networks exist, but we shall only treat the so called Bayesian calculus, which is *classical probability calculus*.

2.3.1 Conditional probabilities

The basic concept in the Bayesian treatment of certainties in causal networks is *conditional probability.* A conditional probability statement is of the following kind:

> *"Given the event b (and everything else known is irrelevant for a), then the probability of the event a is x"*

The notation for the statement above is $P(a \mid b) = x$.

The *Fundamental Rule* for probability calculus is the following:

$$P(a \mid b)P(b) = P(a, b), \tag{1}$$

where $P(a, b)$ is the probability of the joint event $a \wedge b$.

From 1 follows $P(a \mid b)P(b) = P(b \mid a)P(a)$ and this yields the well known *Bayes' Rule*:

$$P(b \mid a) = \frac{P(a \mid b)P(b)}{P(a)}. \tag{2}$$

Let A be a variable with states $a_1, \ldots, a_n$, then $P(A)$ is a probability distribution over these states:

$$P(A) = (x_1, \ldots, x_n); \qquad x_i \geq 0; \qquad \sum_{i=1}^{n} x_i = 1,$$

where x_i is the probability of A being in state a_i.

From a table $P(A, B)$ of probabilities $P(a_i, b_j)$ the probability distribution $P(A)$ can be calculated. Let a_i be a state of A. There are exactly m different events for which A is in state a_i, namely the mutually exclusive events $(a_i, b_1), \ldots, (a_i, b_m)$. Therefore

$$P(a_i) = \sum_{j=1}^{m} P(a_i, b_j).$$

This calculation is called *marginalization* and we say that the variable B is marginalized out of $P(A, B)$ (resulting in $P(A)$). The notation is

$$P(A) = \sum_{B} P(A, B) \tag{3}$$

Note: Probability calculus does not require that the probabilities are based on theoretical results or frequencies of repeated experiments. Probabilities may also be completely subjective estimates of the certainty of an event.

2.4 Bayesian networks

Causal relations also have a quantitative side, namely their *strength.* This is expressed by attaching numbers to the links.

Let A be a parent of B. Using probability calculus it would be natural to let $P(B \mid A)$ be the strength of the link, where $P(A \mid B)$ is the table. However, if also C is a parent of B, then the two conditional probability tables $P(B \mid A)$ and $P(B \mid C)$ alone do not give any clue on how the impacts from A and B interact. They may co-operate or counteract in various ways. So, we need a specification of $P(B \mid A, C)$.

It may happen that the domain to be modelled contains feed-back cycles. Feed-back cycles are difficult to model quantitatively (this is for example what differential equations are all about), and for causal networks no calculus coping with feed-back cycles has been developed. Therefore we require the network not to contain cycles.

A Bayesian network consists of

A set of *variables* and a set of *directed edges* between variables.

Each variable has a finite set of mutually exclusive states.

The variables together with the directed edges form a *directed acyclic graph* (DAG).[1]

To each variable A with parents $B_1, \ldots, B_n$ is attached a conditional probability table $P(A \mid B_1, \ldots, B_n)$.

Note that if A has no parents then the table reduces to unconditional probabilities $P(A)$.

2.5 Evidence and belief revision

Let U be the universe for a Bayesian network. Evidence is information on the state of the variables of U. For simplicity we will only consider evidence as statements of the kind "The variable X is in state x ". We shall let e denote a set of such statements.

Belief revision consists of calculating the posterior distribution $P(X \mid e)$ for all variables X in U . Various methods for belief revision exist. They do in fact calculate $P(X, e)$ for all X, and then $P(X \mid e)$ is calculated by normalizing $P(X, e)$:

$$P(X \mid e) = \frac{P(X, e)}{P(e)}, \text{ where } P(e) = \sum_X P(X, e)$$

We shall not in this article go into the methods for belief revision (called propagation in Bayesian network). The interested reader is referred to the litterature. Fortunately software exists for editing and running Bayesian networks. In *http://bayes.stat.washington.edu/almond/belief.html* a list of available software can be found.

3 Examples

3.1 Insemination (constructed)

Six weeks after insemination of a cow there are three tests for the result: blood test (BT), urine test (UT), and scanning (Sc). The results of the blood test and the urine test are mediated through the hormonal state (Ho) which is affected by a possible pregnancy (Pr).

A model will be like the one in Figure 3.

---

[1] A directed graph is *acyclic* if there is no directed path $A_1 \rightarrow \cdots \rightarrow A_n$ s. t. $A_1 = A_n$.

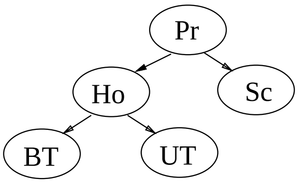

Figure 3: A model for test of pregnancy (Pr). Both the blood test (BT) and the urine test (UT) measure the hormonal state (Ho).

3.2 BOBLO

BOBLO is a system which helps in the verification of parentage for Jersey cattle through blood type identification. The introduction of embryo transplantation technology and the increasing trade of semen and embryos have stressed the importance of proper pedigree registration and therefore there is a need for sophisticated methods for individual identification and parentage control of cattle.

For the blood group determination of cattle, 10 different independent blood group systems are used. These systems control 52 different blood group *factors* which can be measured in a laboratory. In eight of these systems the blood group determination is relatively simple (controlling from one to 4 blood group factors only). However, the systems B- and C- are rather complicated, controlling respectively 26 and 10 of the above-mentioned 52 blood group factors.

Heredity of blood type follows the normal genetic rules, however, the blood groups are attached to sets of loci rather than to single loci of the chromosomes, and instead of allels the term *phenogroup* is used.

So, for each blood group, a Bayesian network for inheritance will be as in Figure 4.

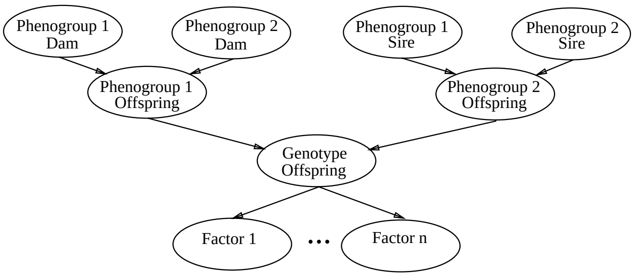

Figure 4: Heredity of blood type. From each parent one out of two phenotypes are chosen. This constitutes the genotype of the offspring, and the genotype determines a set of factors measurable in a laboratory (the phenotype).

If nothing is known of the phenogroups of the parents they are given a prior probability equal to the frequencies of the various phenogroups. Let us for the example suppose that there are three phenogroups f_1, f_2, f_3 with frequencies $(0.58, 0.1, 0.32)$ (this is the situation for the so-called *F-system*).

However, we do not know the parents of the offspring – we only have a stated dam and a stated sire. If the stated parents are the true parents we have no problems, but what if they are not so? Then we will say that the phenogroups of the true parents are distributed as the prior probabilities, that is $(0.58, 0.1, 0.32)$.

So, for modelling the part concerning possible parental errors, we can introduce a node *parental error* with states *both, sire, dam* and *no*, and with prior probabilities to be the frequency of parental errors. This leads to the Bayesian network in Figure 5.

The network model in BOBLO also has a part that models the risks of mistakes in the

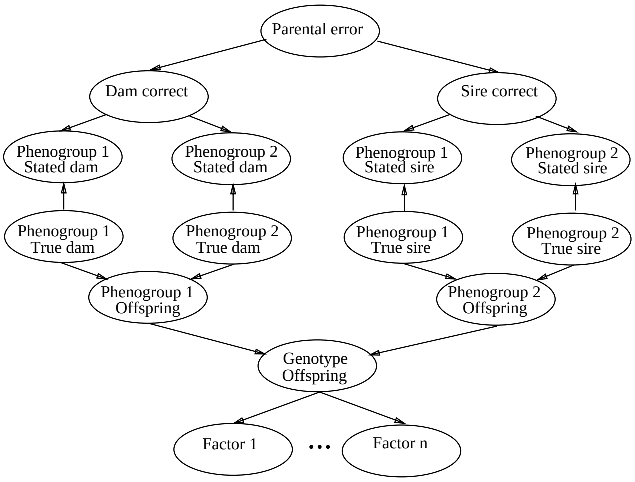

Figure 5: The part of BOBLO modelling parental error. Evidence is entered into the variables *Factor* and *Phenogroup Stated d/s*. Evidence from *Factor* is transmitted to *Parental error* because *Phenogroup stated* has received evidence.

laboratory procedures. For now, assume that evidence on factors are entered directly to the nodes *Factor*. It is assumed that the stated parents are so well known that their genotypes are known, and therefore the state of the variables *Phenogroup Stated d/s* is known.

BOBLO is described in (Rasmussen 1995).

4 Use of Bayesian network models

When you have the facilities for belief revision, you will ask for methods for further expert system facilities, and in this section we shall describe some of these facilities. They have all been implemented as extensions to the HUGIN system (Andersen, Jensen, Olesen & Jensen 1989), and some of them are publicly available. A demo-version of HUGIN is available through *http://www.hugin.dk*.

It should be noted that all features have a cost in terms of time and space. In general, the larger the network the more time and space a propagation requires. Usually the time for a propagation is less than a second, but you may also meet networks where it is practically impossible. Anyhow, for the considerations to come we consider a propagation as the time unit. That is, we will discuss how costly the various features are in terms of number of propagations. As a rule of thumb, you shall avoid tasks requiring more than a thousand propagations.

4.1 Probabilities for specific configurations

Assume that we for some reason in the wet grass example want the probability for both rain and forgotten sprinkler given that Watson's and Holmes' grass is wet. Belief revision only yields $P(\mathit{Rain?} \mid e)$ and $P(\mathit{Sprinkler?} \mid e)$.

This can be achieved through a side effect of the propagation method. As mentioned above, belief revision yields $P(e)$. Next, enter $\mathit{Rain?} = y$ and $\mathit{Sprinkler?} = y$ as virtual evidence. Belief revison then yields the probability of all the entered evidence, and

$$P(\mathit{Rain?} = y, \mathit{Sprinkler} = y \mid e) = \frac{P(\mathit{Rain?} = y, \mathit{Sprinkler} = y, e)}{P(e)}$$

So, to get the probability for a specific configuration only requires one propagation.

4.2 Joint probabilities

Assume that instead of the pobability for a single configuration you want the joint probability distribution for several variables. You may for example in the wet grass example want the joint probability distribution for *Rain?* and *Sprinkler?*.

This can be achieved through some overhead in time. The principle is that you enter the various configurations as virtual evidence and compute their probabilities. This requires one propagation for each configuration. There are smart methods which reduce the number of propagations, but in general the number of propagation is linear in the number of configurations. That is, if you want $P(A, B, C, D)$, where all variables are ternary, then the time for achieving it is in worst case close to the time required for 81 propagations.

4.3 Configuration of maximal probability

Instead of the entire joint distribution for a set of variables, you may only be interested in the configuration of maximal probability. In the wet grass example, you may be interested in the most probable scenario explaining the evidence. This can be achieved through a special belief revision process called *max-propagation*. The complexity of max-propagation is similar to that of normal propagation (also called *sum-propagation*).

In general, let e be the evidence entered, and let V be the remaining set of variables with unknown state.Then one max-propagation yields the most probable configuration in V. However, if you are interested in the most likely configuration of a subset $W \subset V$, then things become more complicated, and there is a risk that the number of required propagations is close to the number of propagations required for computing the joint probability distribution for the subset.

4.4 Data conflict

Findings may be flawed (e.g. red herrings), or findings might originate from a case not covered by the model. If the findings are not directly inconsistent, then a propagation will result in posterior distributions for all variables; so conflicting data can not be detected this way.

This is for example the task which BOBLO solves. In BOBLO, the network is particularly designed for detecting conflicts between the stated parents and the measured factors.

If the possible conflicts are not modelled explicitly, there is another method. Jensen, Chamberlain, Nordahl & Jensen (1991) suggest an index for mismatch between model and findings $x, \ldots, y$:

$$\mathrm{conf}(x, \ldots, y) = \log \frac{P(x) \cdot \ldots \cdot P(y)}{P(x, \ldots, y)}$$

The rationale behind the measure is that findings entered from a coherent case should conform to certain expected patterns, and therefore we would expect $P(x, \ldots, y)$ to be larger than the product of the probabilities for the individual findings. The needed probabilities for calculating the conflict measure are provided directly by the propagation.

5 Influence diagrams

A Bayesian network serves as a model for a part of the world, and the relations in the model reflect causal impact between events. The reason for building these computer models is to use them when taking decisions. That is, the probabilities provided by the network are used to support some kind of decision making. In principle there are two kinds of decisions, namely *test-decisions* and *action-decisions*.

A test-decision is a decision to look for more evidence to be entered into the model, and an action-decision is a decision to change the state of the world. In real life this distinction is not very sharp; tests may have side effects, and by performing a treatment against a disease, evidence on the diagnosis may be acquired. In order to be precise we should say that decisions have two *aspects*, namely a test aspect and an action aspect. The two aspects are handled differently in connection with Bayesian networks, and accordingly we treated them separately. In Section 5.2 we presented methods for test decisions and in this chapter we deal with actions.

We shall treat decision problems in the framework of *utility theory*. The utility of an action may depend on the state of some variables called *determining variables*. For example the utility of a treatment with penicillin is dependent on the type of infection and whether the patient is allergic to penicillin. Let $A = (a_1 \dots a_n)$ be a set of mutually exclusive actions, and let H be the determining variable. What is required in order to specify the problem of deciding between the actions in A is a *utility table* $U(A, H)$ yielding the utility for each configuration of action and determining variable, and the problem is solved by calculating the action which maximizes the expected utility:

$$EU(a) = \sum_H U(a, H) \cdot P(H \mid a)$$

5.1 One set of action options

If you have only one decision to make (see Figure 6), then the calculations are fairly simple: Insert the evidence and the various action options in a Bayesian network and calculate $P(H \mid a)$. This requires one propagation for each action option. Using Bayes' Rule more efficient methods can be constructed (Cooper 1988).

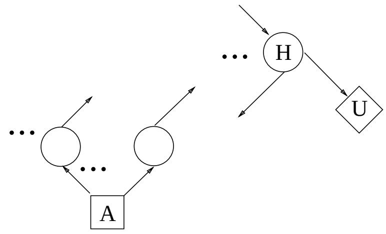

Figure 6: One set of intervening actions.

5.2 Value of information

Whenever decisions under uncertainty are to be made, there is a quest for more information to reduce the uncertainty. However, information is rather seldom cost free, and therefore there is also a need for evaluating on beforehand whether it is worthwhile to consult an information source. Furthermore, if several sources are available there is a need to come up with a strategy for a sequence of data requests.

Consider the insemination example. Before deciding on an action there is a possibility of acquiring information. A scan can be performed at a cost of 40 units; a blood test costs 10 units, a urine test costs 10 units, and a combined blood and urine test costs 15 units (See Figure 7). Should any test be performed, and if so, in which order?

First of all we shall attach a *value* to the various information scenarios. The driving force for evaluating an information scenario is the information on the hypothesis variable Pr. We therefore call this kind of data request situation *hypothesis driven*.

For our example, assume that 6 weeks after the insemination there are two possible actions, *na* (wait another 6 weeks) and *rp* (repeat the insemination). Let $U(Pr, A)$ be a utility table giving the outcome for each combination of action and hypothesis.

The value of any information scenario is a function V of the distribution of Pr. In our case, V is the *expected utility* of performing an optimal action.

$$V(P(Pr)) = \max_{a \in A} \sum_{h \in Pr} U(a, h) P(h)$$

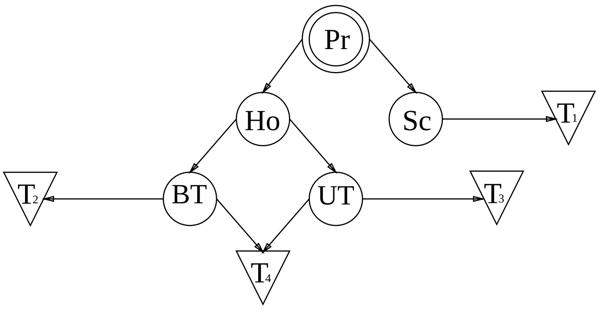

Figure 7: The data request situation in the insemination example. The double circled variable indicates that it is the driving hypothesis variable. A triangle indicates a test yielding the state of its parents.

A proper analysis of the data request situation should consist of an analysis of all possible sequences of tests (including the empty sequence). However, at this place we shall limit ourselves to the *myopic* approach: If you are allowed to consult at most one information source, which option should be chosen?

If test T with cost C_T yields the outcome t, then the value of the new information scenario is

$$V(P(Pr \mid t)) = \max_{a\in A} \sum_{h\in Pr} U(a,h)P(h \mid t)$$

Since the outcome of T is not known we can only calculate the *expected value*

$$EV(T) = \sum_{t\in T} V(P(Pr \mid t)) \cdot P(t)$$

The *expected benefit* of performing test T is

$$EB(T) = EV(T) - V(P(Pr))$$

The *expected profit* is

$$EP(T) = EB(T) - C_T$$

The myopic data request task is to calculate the expected profit for the various tests and to choose the one with maximal expected profit (if positive).

To calculate the expected values in our example, one should determine $P(Pr), P(UT), P(BT), P(Sc), P(BT, UT)$ as well as $P(Pr \mid UT), \ldots, P(Pr \mid BT, UT)$.

5.3 Sequences of actions

Very often you are in a situation where you have to decide on an action A now, knowing that later in time, when more evidence on the system is acquired, you have to take a decision on actions in B. To analyse your present decision problem on A you have to imagine what you will do when deciding on B, and you have to imagine all possible information scenarios.

Bayesian networks can be extended also to represent *symmetric* sequential decision problems. A sequential decision problem is symmetric if, at any given instant of time, the action options and the set of observed variables are independent of previous actions and observations.

The graphical representations are called *Influence Diagrams* (Howard & Matheson 1984). Besides the chance nodes of Bayesian networks, influence diagrams also contain *action nodes* (usually displayed as rectangulars) and *utility nodes* (usually given a diamond shape).

Figure 8 is an example of an influence diagram. It is a simplified version of a model for mildew management in winter wheat ((Jensen 1995)). It consists of weekly time steps, and in Figure 8 we have shown three of them. Each week the mildew attack is observed (*M-obs*) and a decision on a possible treatment is taken (*Treatmt*). During the week the weather has been observed, and now (in week 2) we have to decide on a treatment.

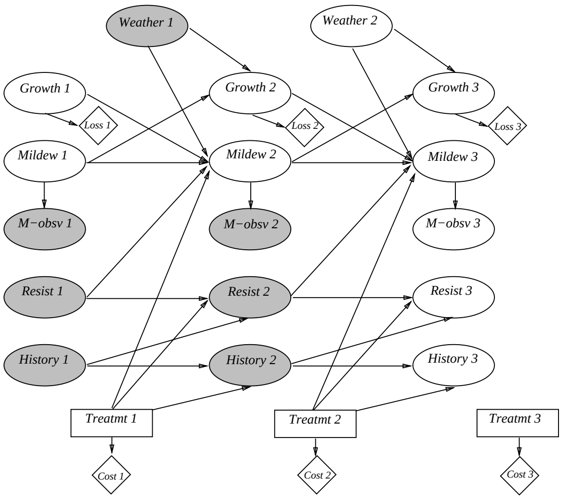

Figure 8: 3 timesteps of a simplified version of an influence diagram for mildew treatment. The grey chance nodes are the ones whose state are known at the time of deciding on *Treatmt 2*.

The links in Figure 8 are causal links. Traditionally, influence diagrams also contain information links. Each action node is given as parents the chance nodes which have been observed since the last decision was made. Usually there is also a link between decision nodes indicating the temporal order. In Figure 8 we have avoided these links. Instead, the influence diagram is read temporarily from left to right, and nodes which become observed are grey. This means that grey nodes at the left of an action variable A are observed at the time of deciding on A.

There are methods for exploiting the structure in an influence diagram to solve the decison problem ((Shachter 1986), (Shenoy 1992)), and Jensen, Jensen & Dittmer (1994) have developped a method which is implemented in HUGIN. However, the problem may be exponential of nature which makes it intractable. The dynamite for the combinatorial explosion is the unobserved chance nodes. If the state of an unobserved node in time step i has an influence on the decision in time step $i + 2$ then the decision problem may explode. In Figure 8 it can be detected graphically: there is a path from *Mildew 1* to *Mildew 3* not containing grey nodes.

Therefore, the model of Figure 8 was changed to the (less correct) model in Figure 9. In Figure 9 it is assumed that the mildew observation is so precise that it can substitute the correct (but unknown) degree of mildew attack. Note that in Figure 9 all paths from time step 1 to time step 3 contain a grey node.

For the structure of Figure 9, the methods for solving influence diagrams boil down to dynamic programming utilizing the efficient updating methods developped for Bayesian networks.

References

Andersen, S. K., Jensen, F. V., Olesen, K. G. & Jensen, F. (1989). HUGIN—a shell for building bayesian belief universes for expert systems, *Technical report*, Institute of Electronic Systems.

Andreassen, S., Jensen, F. V., Andersen, S. K., Falck, B., Kjærulff, U., Woldbye, M., Sørensen, A., Rosenfalck, A. & Jensen, F. (1989). MUNIN - an expert EMG assistant, *in* J. E. Desmedt (ed.), *Computer-Aided Electromyography and Expert Systems*, Elsevier Science Publishers, Amsterdam, chapter 21, pp. 255–277.

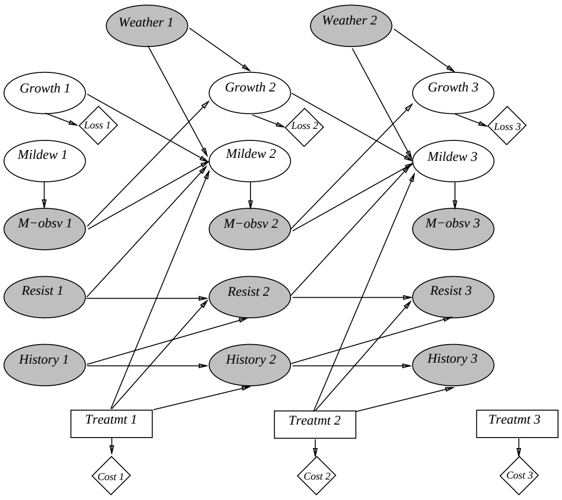

Figure 9: A modified version of Figure 8. In this model you have information blocking between time steps.

Cooper, G. F. (1988). A method for using belief networks as influence diagrams, *Proceedings of the Fourth Workshop on Uncertainty in Artificial Intelligence, Minneapolis*, pp. 55–63.

Gorry, G. (1973). Computer-assisted clinical decision making, *Methods of Information in Medicine* **12**: 45–51.

Gorry, G. & Barnett, G. (1968). Experience with a model of sequential diagnosis, *Computers and Biomedical Research* **1**: 490–507.

Howard, R. A. & Matheson, J. E. (1984). Influence diagrams, *in* R. A. Howard & J. E. Matheson (eds), *Readings on the Principles and Applications of Decision Analysis*, Vol. 2, Strategic Decisions Group, Menlo Park, pp. 719–762.

Jensen, A. L. (1995). A probabilistic model based decision support system for mildew management in winter wheat., *Technical Report DINA Report No. 39*, DINA, Research Centre Foulum, Research Centre Foulum, PB 29, 8830 Tjele, Denmark.

Jensen, F., Jensen, F. V. & Dittmer, S. L. (1994). From influence diagrams to junction trees, *in* R. L. de Mantaras & D. Poole (eds), *Proceedings of the 10th Conference on Uncertainty in Artificial Intelligence*, Morgan Kaufmann, San Fransisco, pp. 367–373.

Jensen, F. V. (1996). *An introduction to Bayesian networks*, University College London Press.

Jensen, F. V., Chamberlain, B., Nordahl, T. & Jensen, F. (1991). Analysis in HUGIN of data conflict, *in* N.-H. Bonnisone et al. (ed.), *Uncertainty in Artificial Intelligence 6*, Elsevier Science Publishers, Amsterdam, pp. 519–528.

Jensen, F. V., Olesen, K. G. & Andersen, S. K. (1990). An algebra of Bayesian belief universes for knowledge-based systems, *Networks* **20**: 637–659.

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

McDermott, J. (1984). R1 re-visited: 4 years in the trenches, *The Artificial Intelligence Magazine* **Fall**: 21–32.

Pearl, J. (1986). Fusion, propagation, and structuring in belief networks, **29**(3): 241–288.

Rasmussen, L. K. (1995). *Bayesian network for blood typing and parentage verification of cattle.*, Dina research report no. 38, Aalborg University, Dept. of Math. and Compu. Science.

Shachter, R. D. (1986). Evaluating influence diagrams, *Operations Research* **34**(6): 871–882.

Shenoy, P. P. (1992). Valuation-based systems for Bayesian decision analysis, *Operations Research* **40**(3): 463–484.

Shortliffe, E. H. (1976). *Computer-based medical consultation: MYCIN*, Elsevier Science Publishers, Amsterdam.