Object-Oriented Modelling by Bayesian Networks

Based on the HUGIN Training Course by Anders L. Madsen

Prof. Dr. Galia Weidl

Basics of Decision-Making

TH Aschaffenburg

WS2022/2023

Outline

- Bayesian networks
- Object oriented Bayesian networks
  - Network classes
  - Instance nodes
  - Interface
  - Restrictions, scope, and typing
- Dynamic models

Bayesian Networks

The graphical structure of a Bayesian network is flat

- It is a compact model representation of a problem domain
- May sometimes be a large network with many nodes and edges
- Often only few nodes are of interest

A Bayesian network can often naturally be split up into a number of network fragments

- It becomes a hierarchical structure of network fragments

Object-Oriented Modelling

Hugin provides a simple, but powerful, mechanism for object-oriented modelling:

- Hierarchical model construction
- Easy specification of nets with identical subnets
- Top-down and bottom-up modelling naturally supported
- Model reuse naturally supported

Some key concepts:

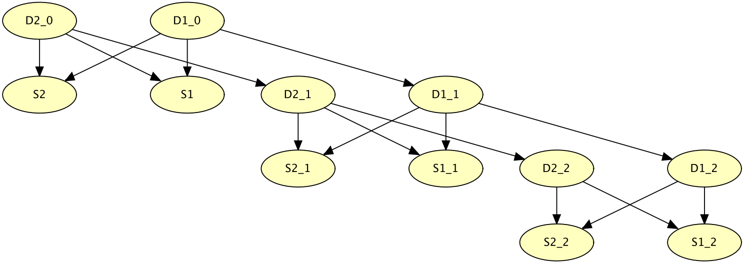
- Instance nodes
- Interface nodes
- private nodes
- Abstraction, encapsulation, information hiding, and inheritance

Example: Car Accident (OOBN versus flat)

file:///C:/Program%20Files/Hugin%20Expert/HUGIN%209.2%20(x64)/Htmlhelp/pages/Tutorials/ObjectOrientedBayesianNetwork/IntroductionToObjectOrientedNetworks.html

Object-Oriented Modeling

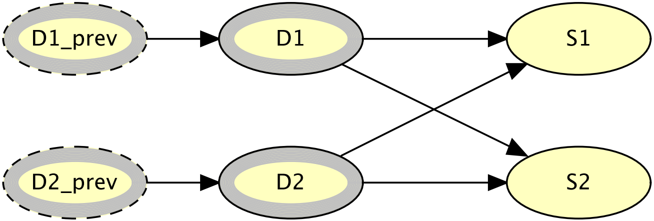
We would like to model the progression of two diseases D_1 and D_2 over three time steps. Assume there are two symptoms S_1 and S_2 which indicate manifestations of D_1 and D_2.



- The network subnets are identical for all time slices
- Specifying this as a normal Bayesian network makes specification tedious and modifications cumbersome

Object-Oriented Modeling

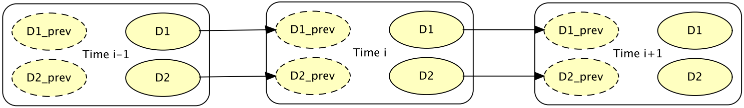
Using an object-oriented approach a network class with proper interface is defined



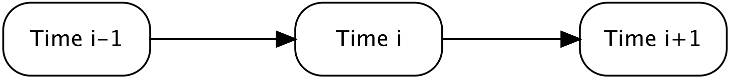
- Letting each subnet be an instance of a "network class" makes specification and modifications much easier

Object-Oriented Modeling

Three instances of the network class are connected to reflect progression of the diseases over time



An instance node may be collapsed to hide its interface



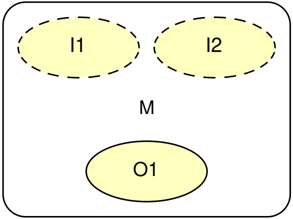
Socks

I have washed two pairs of socks in the washing machine. The washing has been rather hard on them, so they are now difficult to distinguish. However, it is important for me to pair them correctly. To classify the socks I have pattern and color. There are two types of patterns and two different colors.

Object Oriented Bayesian Networks

A network class is characterized by its

- Name
- Interface
- Hidden part



An BN can be viewed as a hierarchical tree structure over a set of subnets supporting:

- Abstraction, encapsulation, information hiding, and inheritance

Classes and Instance Nodes

An instance node M is a named and self-contained description of a network class

An instance node consists of three different kinds of nodes:

- Input, hidden, and output
- Internal and external, public and private nodes

The properties of instance nodes are

- They can be encapsulated within other network classes
- Only loaded network classes are available
- No local changes

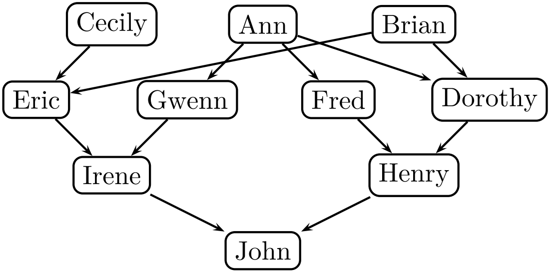
Stud Farm

Consider a studfarm with 10 horses: Cecily has unknown mare and sire, John has mare Irene and sire Henry, Henry has mare Dorothy and sire Fred, Irene has mare Gwenn and sire Eric, Gween has mare Ann and unknown sire, Eric has mare Cecily and sire Brian, Fred has mare Ann and unknown sire, Brian has unknown mare and sire, Dorothy has mare Ann and sire Brian, and Ann has unknown mare and sire.

A sick horse has genotype aa, a carrier of the disease has genotype aA, and a non-carrier has genotype AA. $\mathrm{P}(\mathrm{aa}, \mathrm{aA}, \mathrm{AA}) = (0.04, 0.32, 0.64)$

What is the probability of each horse being sick/a carrier/a non-carrier once we learn that John is sick ?

Stud Farm Pedigree



A sick horse has genotype aa, a carrier of the disease has genotype aA, and a non-carrier has genotype AA

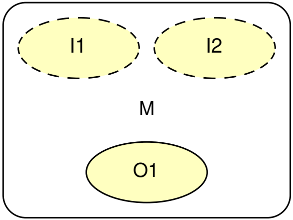
$\mathrm{P}(\mathrm{aa}, \mathrm{aA}, \mathrm{AA}) = (0.04, 0.32, 0.64)$

What is the probability of each horse being sick/a carrier/a non-carrier once we learn that John is sick ?

Interface of an Instance Node

An instance node M consists of input I(M), output O(M), and hidden nodes H(M)

- The interface is $I(M) \cup O(M)$



The properties of the interface are

- No instance nodes
- No edges are explicitly shown
- The sets of public and private nodes are disjoint, as are the sets of input and output nodes

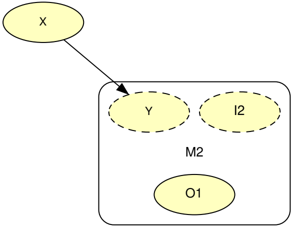
Input Nodes

An input node of an instance node is a reference to a node outside the internal scope of the instance node

- A parent $X \in M_1$ of an input node $Y \in I(M_2)$ specifies that Y is a placeholder for X in the instance node M_2,
- Gives $Z \in H(M_2)$ access to X

The properties of an input node are

- It has a default potential
- It can have at most one parent
- Utility nodes cannot be input



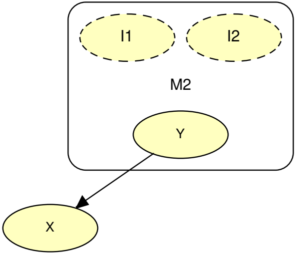
Output Nodes

An output node of an instance node gives access to an internal node outside the internal scope of the instance node

- Used when an internal node has to be parent of external nodes

The properties of output nodes are

- A node $Y \in O(M)$ can only have $X \in V(M)$ as parent
- A utility node cannot be output



Typing

Strong type-checking on discrete input nodes

- Category, kind, and states labels have to be the same,
- The names of the nodes do not need to be the same,

Weak type-checking on continuous input nodes

- The distribution has to be the same

The type of an instance node is the network class the node is an instance of

Syntactical Restrictions

- A node $J \in M_1$ cannot be parent of $I_1, I_2 \in I(M_2)$ where $M_2 \in M_1$ and I_1, I_2 have a common child
- A node $X \in I(M_1)$ cannot be parent of a node $Y \in V(M_2)$, unless $M_1 = M_2$ and $Y \notin I(M_2)$
- A node $X \in M_1$ cannot be parent of node $Y \in O(M_2)$, if $M_2 \in M_1$
- No instance nodes as interface nodes
- The instantiated network must be a DAG

Insemination

After the initial insemination, the farmer monitors the cow for three weeks. At the beginnig of each week the farmer makes a decision on whether to repeat the insemination or wait. The cost of waiting one period is 10 if the cow is not pregnant.

If the cow was pregnant at the last time step, the probability of it still being pregnant is 0.99 and 0.999 if you wait and repeat, respectively.

After monitoring the cow for three periods the insemination is repeated, if the cow is not pregnant.

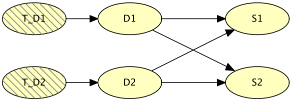
The result of the most recent blood test is known prior to the decision.

What is the strategy of the farmer?

Dynamic Models

Dynamic Bayesian networks are used to model systems that evolve over time

- The state of the system at a single time instant is modelled as a static BBN
- Time slices are linked to create a dynamic model
- Temporal dependencies are modelled using temporal clones
- Transition probability distributions between temporal clones and regular nodes



The temporal clones represent the interface between two successive time slices

Summary

- Bayesian networks
- Object oriented Bayesian networks
  - Network classes
  - Instance nodes
  - Interface
  - Restrictions, scope, and typing
- Dynamic models