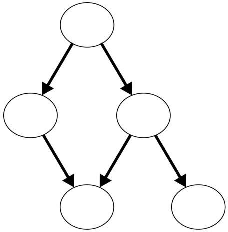
Multiconnected Networks
Variable Elimination
Conditioning
Junction Tree Algorithm
Approximate Inference
Loopy Propagation
Stochastic Simulation
Most probable explanation
Continuous variables
Applications
Information Validation
Reliability Analysis
References

# Bayesian Networks - Inference (Part II)

## Probabilistic Graphical Models

L. Enrique Sucar, INAOE



# Outline

1. Multiconnected Networks
   - Variable Elimination
   - Conditioning
   - Junction Tree Algorithm
2. Approximate Inference
   - Loopy Propagation
   - Stochastic Simulation
3. Most probable explanation
4. Continuous variables
5. Applications
   - Information Validation
   - Reliability Analysis
6. References

# Inference - Multiconnected networks

Multiconnected Networks
Variable Elimination
Conditioning
Junction Tree Algorithm
Approximate Inference
Loopy Propagation
Stochastic Simulation
Most probable explanation
Continuous variables
Applications
Information Validation
Reliability Analysis
References

- There are several classes of algorithms for probabilistic inference on multi conneced BNs:
  - variable elimination,
  - conditioning,
  - junction tree.

# Variable Elimination

Multiconnected Networks
Variable Elimination
Conditioning
Junction Tree Algorithm
Approximate Inference
Loopy Propagation
Stochastic Simulation
Most probable explanation
Continuous variables
Applications
Information Validation
Reliability Analysis
References

- The variable elimination technique is based on the idea of calculating the probability by marginalizing the joint distribution
- It takes advantage of the independence conditions of the BN and the associative and distributive properties of addition and multiplication to do the calculations more efficiently:
  1. Represent the joint distribution as a product of local probabilities according to the network structure
  2. Summations can be carried out only on the subset of terms which are a function of the variables being normalized

Multiconnected Networks
Variable Elimination
Conditioning
Junction Tree Algorithm
Approximate Inference
Loopy Propagation
Stochastic Simulation
Most probable explanation
Continuous variables
Applications
Information Validation
Reliability Analysis
References

# Variable Elimination

- Joint probability distribution of $\mathbf{X} = \{X_1, X_2, ..., X_n\}$
- Posterior probability of a certain variable or subset of variables, $X_H$, given a subset of evidence variables, $X_E$; the remaining variables are $X_R$, such that $\mathbf{X} = \{X_H \cup X_E \cup X_R\}$:

$$P(X_H \mid X_E) = P(X_H, X_E)/P(X_E) \quad (1)$$

- We can obtain both terms via marginalization of the joint distribution:

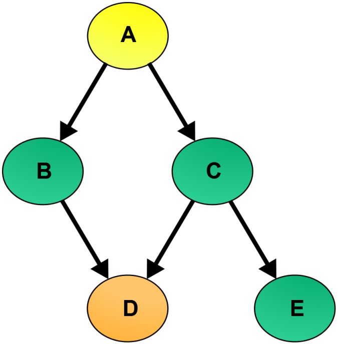
$$P(X_H, X_E) = \sum_{X_R} P(\mathbf{X}) \quad (2)$$

and

$$P(X_E) = \sum_{X_H} P(X_H, X_E) \quad (3)$$

Multiconnected Networks
Variable Elimination
Conditioning
Junction Tree Algorithm
Approximate Inference
Loopy Propagation
Stochastic Simulation
Most probable explanation
Continuous variables
Applications
Information Validation
Reliability Analysis
References

# VE - illustration




- Obtain $P(A \mid D)$ - we need to obtain $P(A, D)$ and $P(D)$.

Multiconnected Networks
Variable Elimination
Conditioning
Junction Tree Algorithm
Approximate Inference
Loopy Propagation
Stochastic Simulation
Most probable explanation
Continuous variables
Applications
Information Validation
Reliability Analysis
References

# VE - calculations

- *Eliminate* $B, C, E$ from the joint distribution, that is:

$$P(A, D) =$$

$$\sum_B \sum_C \sum_E P(A)P(B \mid A)P(C \mid A)P(D \mid B, C)P(E \mid C)$$

- By *distributing* the summations we can arrive to the following equivalent expression:

$$P(A, D) =$$

$$P(A) \sum_B P(B \mid A) \sum_C P(C \mid A)P(D \mid B, C) \sum_E P(E \mid C)$$

- If all variables are binary, this implies a reduction from 32 operations to 9 operations

# VE - example

Multiconnected Networks
Variable Elimination
Conditioning
Junction Tree Algorithm
Approximate Inference
Loopy Propagation
Stochastic Simulation
Most probable explanation
Continuous variables
Applications
Information Validation
Reliability Analysis
References

- Obtain $P(E \mid F = f_1) = P(E, F = f_1)/P(F = f_1)$
- Joint distribution:
  $P(C, E, F, D) = P(C)P(E \mid C)P(F \mid E)P(D \mid E)$

Multiconnected Networks
Variable Elimination
Conditioning
Junction Tree Algorithm
Approximate Inference
Loopy Propagation
Stochastic Simulation
Most probable explanation
Continuous variables
Applications
Information Validation
Reliability Analysis
References

# VE - example

- First calculate $P(E, F)$; by reordering the operations:

$$P(E,F) = \sum_D P(F \mid E) P(D \mid E) \sum_C P(C) P(E \mid C)$$

- Calculation for each value of $E$, given $F = f_1$:

$$P(e_1, f_1) = \sum_D P(f_1 \mid e_1) P(D \mid e_1) \sum_C P(C) P(e_1 \mid C)$$

$$P(e_1, f_1) = \sum_D P(f_1 \mid e_1) P(D \mid e_1) [0.9 \times 0.8 + 0.7 \times 0.2]$$

$$P(e_1, f_1) = \sum_D P(f_1 \mid e_1) P(D \mid e_1) [0.86]$$

$$P(e_1, f_1) = [0.9 \times 0.7 + 0.9 \times 0.3][0.86]$$

$$P(e_1, f_1) = [0.9][0.86] = 0.774$$

# VE - example

Multiconnected Networks
Variable Elimination
Conditioning
Junction Tree Algorithm
Approximate Inference
Loopy Propagation
Stochastic Simulation
Most probable explanation
Continuous variables
Applications
Information Validation
Reliability Analysis
References

- In a similar way we obtain $P(e_2, f_1)$; and then from these values we can calculate $P(f_1) = \sum_E P(E, f_1)$
- Finally, we calculate the posterior probability of $E$ given $f_1$: $P(e_1 \mid f_1) = P(e_1, f_1)/P(f_1)$ and $P(e_2 \mid f_1) = P(e_2, f_1)/P(f_1)$

# Analysis

Multiconnected Networks
Variable Elimination
Conditioning
Junction Tree Algorithm
Approximate Inference
Loopy Propagation
Stochastic Simulation
Most probable explanation
Continuous variables
Applications
Information Validation
Reliability Analysis
References

- The critical aspect of the variable elimination algorithm is to select the appropriate order for eliminating each variable, as this has an important effect on the number of required operations
- The different terms that are generated during the calculations are known as *factors* which are functions over a subset of variables, in the previous example, one of the factors is $f(C,E) = P(C)P(E \mid C)$
- The computational complexity in terms of space and time of the variable elimination algorithm is determined by the size of the factors – is exponential on the number of variables in the factor.

Multiconnected Networks
Variable Elimination
Conditioning
Junction Tree Algorithm
Approximate Inference
Loopy Propagation
Stochastic Simulation
Most probable explanation
Continuous variables
Applications
Information Validation
Reliability Analysis
References

# Elimination Order

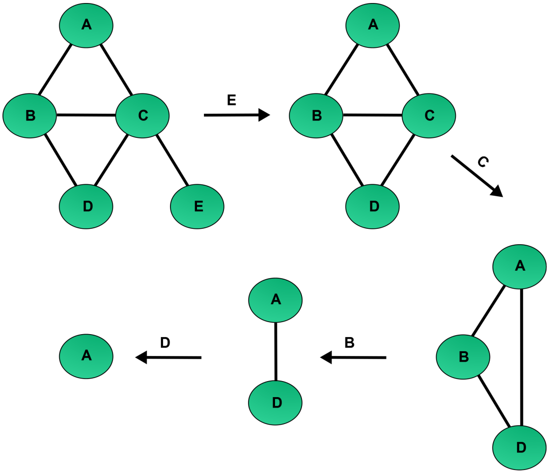
- Finding the *best* order is in general a NP-Hard problem
- There are several heuristics that help to determine a *good* ordering for variable elimination:

  Min-degree: eliminate the variable that leads to the smallest possible factor; which is equivalent to eliminating the variable with the smallest number of neighbors in the current elimination graph.

  Min-fill: eliminate the variable that leads to adding the minimum number of edges to the interaction graph.
- These heuristics can be explained based on the *interaction graph* –an undirected graph that is built during the process of variable elimination

Multiconnected Networks
Variable Elimination
Conditioning
Junction Tree Algorithm
Approximate Inference
Loopy Propagation
Stochastic Simulation
Most probable explanation
Continuous variables
Applications
Information Validation
Reliability Analysis
References

# Interaction graph




- When $X_j$ is eliminated the interaction graph is modified: (i) adding an arc between each pair of neighbors of $X_j$, (ii) deleting variable $X_j$ from the graph

# Analysis

Multiconnected Networks
Variable Elimination
Conditioning
Junction Tree Algorithm
Approximate Inference
Loopy Propagation
Stochastic Simulation
Most probable explanation
Continuous variables
Applications
Information Validation
Reliability Analysis
References

- A disadvantage of variable elimination is that it only obtains the posterior probability of one variable
- To obtain the posterior probability of each non-instantiated variable in a BN, the calculations have to be repeated for each variable
- Next, we describe two algorithms that calculate the posterior probabilities for all variables at the same time

# Conditioning

Multiconnected Networks
Variable Elimination
Conditioning
Junction Tree Algorithm
Approximate Inference
Loopy Propagation
Stochastic Simulation
Most probable explanation
Continuous variables
Applications
Information Validation
Reliability Analysis
References

- The conditioning method is based on the fact that an instantiated variable *blocks* the propagation of the evidence in a Bayesian network
- Can *cut* the graph at an instantiated variable, and this can transform a multi connected graph into a polytree, for which we can apply the probability propagation algorithm
- If these variables are not actually known, we can set them to each of their possible values, and then do probability propagation for each value
- With each propagation we obtain a probability for each unknown variable – the final probability values are obtained as a weighted combination of these probabilities

Multiconnected Networks
Variable Elimination
Conditioning
Junction Tree Algorithm
Approximate Inference
Loopy Propagation
Stochastic Simulation
Most probable explanation
Continuous variables
Applications
Information Validation
Reliability Analysis
References

# Formalization

- Formally, we want to obtain the probability of any variable, $B$, given the evidence $E$, conditioning on variable $A$. By the rule of total probability:

$$P(B \mid E) = \sum_i P(B \mid E, a_i) P(a_i \mid E) \tag{4}$$

- Where:

  $P(B \mid E, a_i)$ is the posterior probability of $B$ which is obtained by probability propagation for each possible value of $A$.

  $P(a_i \mid E)$ is a *weight*.

# Formalization

Multiconnected Networks
Variable Elimination
Conditioning
Junction Tree Algorithm
Approximate Inference
Loopy Propagation
Stochastic Simulation
Most probable explanation
Continuous variables
Applications
Information Validation
Reliability Analysis
References

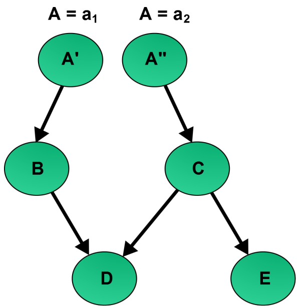
- By applying the Bayes rule we obtain the following equation to estimate the weights:

$$P(a_i \mid E) = \alpha P(a_i) P(E \mid a_i) \tag{5}$$

- The first term, $P(a_i)$, can be obtained by propagating without evidence. The second term, $P(E \mid a_i)$, is calculated by propagation with $A = a_i$ to obtain the probability of the evidence variables. $\alpha$ is a normalizing constant

Multiconnected Networks
Variable Elimination
Conditioning
Junction Tree Algorithm
Approximate Inference
Loopy Propagation
Stochastic Simulation
Most probable explanation
Continuous variables
Applications
Information Validation
Reliability Analysis
References

# Example




- If the evidence is $D$, $E$, then probabilities for the other variables, $A$, $B$, $C$ can be obtained via conditioning

Multiconnected Networks
Variable Elimination
Conditioning
Junction Tree Algorithm
Approximate Inference
Loopy Propagation
Stochastic Simulation
Most probable explanation
Continuous variables
Applications
Information Validation
Reliability Analysis
References

# Example

1. Obtain the prior probability of $A$ (in this case it is already given as it is a root node).
2. Obtain the probability of the evidence nodes $D, E$ for each value of $A$ by propagation in the polytree.
3. Calculate the weights, $P(a_i \mid D, E)$, from (1) and (2) with the Bayes rule.
4. Estimate the probability of $B$ and $C$ for each value of $A$ given the evidence by probability propagation in the polytree.
5. Obtain the posterior probabilities for $B$ and $C$ from (3) and (4) by applying equation 4.

# Analysis

Multiconnected Networks
Variable Elimination
Conditioning
Junction Tree Algorithm
Approximate Inference
Loopy Propagation
Stochastic Simulation
Most probable explanation
Continuous variables
Applications
Information Validation
Reliability Analysis
References

- In general, to transform a multi connected BN to a polytree we need to instantiate $m$ variables
- Thus, propagation must be performed for all the combinations of values (cross product) of the instantiated variables
- If each variable has $k$ values, then the number of propagations is $k^m$

Multiconnected Networks
Variable Elimination
Conditioning
Junction Tree Algorithm
Approximate Inference
Loopy Propagation
Stochastic Simulation
Most probable explanation
Continuous variables
Applications
Information Validation
Reliability Analysis
References

# Junction Tree

- The junction tree method is based on a transformation of the BN to a junction tree, where each node in this tree is a group or cluster of variables
- Probabilistic inference is performed over this new representation, via propagation over the junction tree
- The probability of a variable is obtained by marginalization over the "junction" (clique)

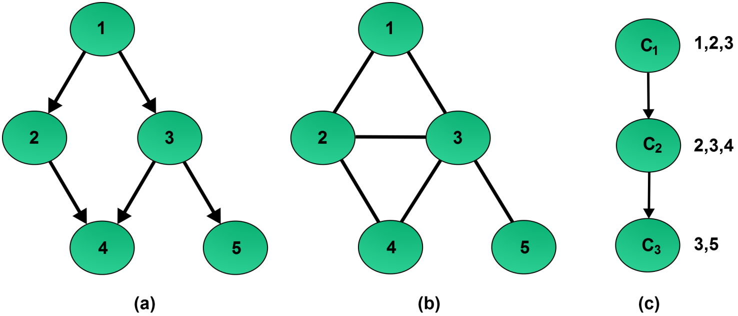
Multiconnected Networks
Variable Elimination
Conditioning
Junction Tree Algorithm
Approximate Inference
Loopy Propagation
Stochastic Simulation
Most probable explanation
Continuous variables
Applications
Information Validation
Reliability Analysis
References

# Transformation

1. Eliminate the directionality of the arcs.
2. Order the nodes in the graph (based on *maximum cardinality*).
3. Moralize the graph (add an arc between pairs of nodes with common children).
4. If necessary add additional arcs to make the graph *triangulated*.
5. Obtain the *cliques* of the graph (subsets of nodes that are fully connected and are not subsets of other fully connected sets).
6. Build a junction tree in which each node is a clique and its parent is any node that contains all common previous variables according to the ordering.

Multiconnected Networks
Variable Elimination
Conditioning
Junction Tree Algorithm
Approximate Inference
Loopy Propagation
Stochastic Simulation
Most probable explanation
Continuous variables
Applications
Information Validation
Reliability Analysis
References

# Transformation - example



# Inference

Multiconnected Networks
Variable Elimination
Conditioning
Junction Tree Algorithm
Approximate Inference
Loopy Propagation
Stochastic Simulation
Most probable explanation
Continuous variables
Applications
Information Validation
Reliability Analysis
References

- Once the junction tree is built, inference is based on probability propagation over the junction tree
- Initially the joint probability (potential) of each macro node is obtained, and given some evidence, this is propagated to obtain the posterior probability of each junction
- The individual probability of each variable is obtained from the joint probability of the appropriate junction via marginalization

Multiconnected Networks
Variable Elimination
Conditioning
Junction Tree Algorithm
Approximate Inference
Loopy Propagation
Stochastic Simulation
Most probable explanation
Continuous variables
Applications
Information Validation
Reliability Analysis
References

# Preprocessing

In the preprocessing phase the potentials of each clique are obtained following the next steps:

1. Determine the set of variables for each clique, $C_i$.
2. Determine the set of variables that are common with the previous (parent) clique, $S_i$.
3. Determine the variables that are in $C_i$ but not in $S_i$: $R_i = C_i - S_i$.
4. Calculate the potential of each clique, $clq_i$, as the product of the corresponding CPTs: $\psi(clq_i) = \prod_j P(X_j \mid Pa(X_j))$; where $X_j$ are the variables in $clq_i$.

Multiconnected Networks
Variable Elimination
Conditioning
Junction Tree Algorithm
Approximate Inference
Loopy Propagation
Stochastic Simulation
Most probable explanation
Continuous variables
Applications
Information Validation
Reliability Analysis
References

# Preprocessing - example

Cliques: $clq_1 = \{1,2,3\}$, $clq_2 = \{2,3,4\}$, $clq_3 = \{3,5\}$.
Then the preprocessing phase is:

$C$: $C_1 = \{1,2,3\}$, $C_2 = \{2,3,4\}$, $C_3 = \{3,5\}$.

$S$: $S_1 = \emptyset$, $S_2 = \{2,3\}$, $S_3 = \{3\}$.

$R$: $R_1 = \{1,2,3\}$, $R_2 = \{4\}$, $R_3 = \{5\}$.

Potentials: $\psi(clq_1) = P(1)P(2 \mid 1)P(3 \mid 2)$,
$\psi(clq_2) = P(4 \mid 3,2)$, $\psi(clq_3) = P(5 \mid 3)$.

Multiconnected Networks
Variable Elimination
Conditioning
Junction Tree Algorithm
Approximate Inference
Loopy Propagation
Stochastic Simulation
Most probable explanation
Continuous variables
Applications
Information Validation
Reliability Analysis
References

# Propagation

- The propagation phase proceeds in a similar way to belief propagation for trees, by propagating $\lambda$ messages bottom-up and $\pi$ messages top-down
- **Bottom-Up Propagation**
  1. Calculate the $\lambda$ message to send to the parent clique: $\lambda(C_i) = \sum_R \psi(C_i)$.
  2. Update the potential of each clique with the $\lambda$ messages of its sons: $\psi(C_j)' = \lambda(C_i)\psi(C_j)$.
  3. Repeat the previous two steps until reaching the root clique.
  4. When reaching the root node obtain $P'(C_r) = \psi(C_r)'$.

# Propagation

Multiconnected Networks
Variable Elimination
Conditioning
Junction Tree Algorithm
Approximate Inference
Loopy Propagation
Stochastic Simulation
Most probable explanation
Continuous variables
Applications
Information Validation
Reliability Analysis
References

- **Top-Down Propagation**
  1. Calculate the $\pi$ message to send to each child node $i$: $\pi(C_i) = \sum_{C_j - S_i} P'(C_j)$.
  2. Update the potential of each clique when receiving the $\pi$ message of its parent: $P'(C_i) = \pi(Ci)\psi(C_i)'$.
  3. Repeat the previous two steps until reaching the leaf nodes in the junction tree.
- When there is evidence, the potentials for each clique are updated based on the evidence, and the same propagation procedure is followed
- Finally, the marginal posterior probabilities of each variable are obtained from the clique potentials via marginalization: $P(X) = \sum_{C_i - X} P'(C_i)$

Multiconnected Networks
Variable Elimination
Conditioning
Junction Tree Algorithm
Approximate Inference
Loopy Propagation
Stochastic Simulation
Most probable explanation
Continuous variables
Applications
Information Validation
Reliability Analysis
References

# Complexity analysis

- In the worst case, probabilistic inference for Bayesian networks is NP-Hard
- The time and space complexity is determined by what is known as the *tree-width* - a tree-structured BN (maximum one parent per variable) has a tree-width of one. A polytree with at most $k$ parents per node has a tree-width of $k$
- In general, the tree-width is determined by how *dense* the topology of the network is, and this has affects: (i) the size of the largest factor in the variable elimination algorithm; (ii) the number of variables that need to be instantiated in the conditioning algorithm, (iii) the size of the largest clique in the junction tree algorithm

# Loopy Propagation

Multiconnected Networks
Variable Elimination
Conditioning
Junction Tree Algorithm
Approximate Inference
Loopy Propagation
Stochastic Simulation
Most probable explanation
Continuous variables
Applications
Information Validation
Reliability Analysis
References

- This is simply the application of the probability propagation algorithm for multi connected networks
- Although in this case the conditions for this algorithm are not satisfied, and it only provides an approximate solution
- Given that the BN is not singly connected, as the messages are propagated, these can *loop* through the network

Multiconnected Networks
Variable Elimination
Conditioning
Junction Tree Algorithm
Approximate Inference
Loopy Propagation
Stochastic Simulation
Most probable explanation
Continuous variables
Applications
Information Validation
Reliability Analysis
References

# Procedure

1. Initialize the $\lambda$ and $\pi$ values for all nodes to random values.
2. Repeat until convergence or a maximum number of iterations:
    1. Do probability propagation according to the algorithm for singly connected networks.
    2. Calculate the posterior probability for each variable.

# Convergence

Multiconnected Networks
Variable Elimination
Conditioning
Junction Tree Algorithm
Approximate Inference
Loopy Propagation
Stochastic Simulation
Most probable explanation
Continuous variables
Applications
Information Validation
Reliability Analysis
References

- The algorithm converges when the difference between the posterior probabilities for all variables of the current and previous iterations is below a certain threshold
- It has been found empirically that for certain structures this algorithm converges to the true posterior probabilities; however, for other structures it does not converge
- An important application of loopy belief propagation is in “Turbo Codes”; which is a popular error detection and correction scheme used in data communications

# Stochastic simulation

Multiconnected Networks
Variable Elimination
Conditioning
Junction Tree Algorithm
Approximate Inference
Loopy Propagation
Stochastic Simulation
Most probable explanation
Continuous variables
Applications
Information Validation
Reliability Analysis
References

- Stochastic simulation algorithms consist in *simulating* the BN several times, where each simulation gives a sample value for all non-instantiated variables
- These values are chosen randomly according to the conditional probability of each variable
- This process is repeated $N$ times, and the posterior probability of each variable is approximated in terms of the frequency of each value in the sample space

Multiconnected Networks
Variable Elimination
Conditioning
Junction Tree Algorithm
Approximate Inference
Loopy Propagation
Stochastic Simulation
Most probable explanation
Continuous variables
Applications
Information Validation
Reliability Analysis
References

# Logic Sampling

- Logic sampling is a basic stochastic simulation algorithm that generates samples according to the following procedure:
  1. Generate sample values for the root nodes of the BN according to their prior probabilities. That is, a random value is generated for each root variable $X$, following a distribution according to $P(X)$.
  2. Generate samples for the next *layer*, that is the sons of the already sampled nodes, according to their conditional probabilities, $P(Y \mid Pa(Y))$, where $Pa(Y)$ are the parents of $Y$.
  3. Repeat (2) until all the leaf nodes are reached.
- The previous procedure is repeated $N$ times to generate $N$ samples. The probability of each variable is estimated as the fraction of times (frequency) that a value occurs in the $N$ samples, that is, $P(X = x_i) \sim No(x_i)/N$

Multiconnected Networks
Variable Elimination
Conditioning
Junction Tree Algorithm
Approximate Inference
Loopy Propagation
Stochastic Simulation
Most probable explanation
Continuous variables
Applications
Information Validation
Reliability Analysis
References

# Example

10 samples generated by logic sampling (assuming all variables are binary):

| variables | A | B | C | D | E |
|---|---|---|---|---|---|
| $sample_1$ | T | F | F | F | T |
| $sample_2$ | F | T | T | F | F |
| $sample_3$ | T | F | F | T | F |
| $sample_4$ | F | F | T | F | T |
| $sample_5$ | T | F | T | T | F |
| $sample_6$ | F | F | F | F | T |
| $sample_7$ | F | T | T | T | F |
| $sample_8$ | F | F | F | F | F |
| $sample_9$ | F | F | F | T | F |
| $sample_{10}$ | T | T | T | T | F |

# Probabilities - no evidence

Multiconnected Networks
Variable Elimination
Conditioning
Junction Tree Algorithm
Approximate Inference
Loopy Propagation
Stochastic Simulation
Most probable explanation
Continuous variables
Applications
Information Validation
Reliability Analysis
References

If there is no evidence, then given these samples, the marginal probabilities are estimated as follows:

- $P(A = T) = 4/10 = 0.4$
- $P(B = T) = 3/10 = 0.3$
- $P(C = T) = 5/10 = 0.5$
- $P(D = T) = 5/10 = 0.5$
- $P(E = T) = 3/10 = 0.3$

# Probabilities - with evidence

Multiconnected Networks
Variable Elimination
Conditioning
Junction Tree Algorithm
Approximate Inference
Loopy Propagation
Stochastic Simulation
Most probable explanation
Continuous variables
Applications
Information Validation
Reliability Analysis
References

In the case where there is evidence with $D = T$, we eliminate all the samples where $D = F$, and estimate the posterior probabilities from the remaining 5 samples:

- $P(A = T \mid D = T) = 3/5 = 0.6$
- $P(B = T \mid D = T) = 2/5 = 0.4$
- $P(C = T \mid D = T) = 3/5 = 0.6$
- $P(E = T \mid D = T) = 1/5 = 0.2$

# Likelihood Weighting

Multiconnected Networks
Variable Elimination
Conditioning
Junction Tree Algorithm
Approximate Inference
Loopy Propagation
Stochastic Simulation
Most probable explanation
Continuous variables
Applications
Information Validation
Reliability Analysis
References

- A disadvantage of logic sampling when evidence exists is that many samples have to be discarded
- Likelihood weighting generates samples in the same way as logic sampling, however when there is evidence the non-consistent samples are not discarded
- Each sample is given a weight according to the weight of the evidence for this sample

Multiconnected Networks
Variable Elimination
Conditioning
Junction Tree Algorithm
Approximate Inference
Loopy Propagation
Stochastic Simulation
Most probable explanation
Continuous variables
Applications
Information Validation
Reliability Analysis
References

# Weighting

- Given a sample $s$ and the evidence variables $\mathbf{E} = \{E_1, ..., E_m\}$, the weight of sample $s$ is estimated as:

$$W(\mathbf{E} \mid s) = P(E_1)P(E_2)...P(E_m) \tag{6}$$

where $P(E_i)$ is the probability of the evidence variable $E_i$ for that sample

- The posterior probability for each variable $X$ taking value $x_i$ is estimated by dividing the sum of the weights $W_i(X = x_i)$ for each sample where $X = x_i$ by the total weight for all the samples:

$$P(X = x_i) \sim \sum_i W_i(X = xi) / \sum_i W_i \tag{7}$$

# MPE

Multiconnected Networks
Variable Elimination
Conditioning
Junction Tree Algorithm
Approximate Inference
Loopy Propagation
Stochastic Simulation
Most probable explanation
Continuous variables
Applications
Information Validation
Reliability Analysis
References

- The *most probable explanation* (MPE) or *abduction* problem consists in determining the most probable values for a subset of variables (explanation subset) in a BN given some evidence
- Two variants: *total abduction* and *partial abduction*
- In the total abduction problem, the explanation subset is the set of all non-instantiated variables
- In partial abduction, the explanation subset is a proper subset of the non-instantiated variables

# Formally

Multiconnected Networks
Variable Elimination
Conditioning
Junction Tree Algorithm
Approximate Inference
Loopy Propagation
Stochastic Simulation
Most probable explanation
Continuous variables
Applications
Information Validation
Reliability Analysis
References

- Consider the set of variables $\mathbf{X} = \{X_E, X_R, X_H\}$, where $X_E$ is the subset of instantiated variables; then we can formalize the MPE problems as follows:

  Total abduction: $ArgMax_{X_H, X_R} P(X_H, X_R \mid X_E)$.
  Partial abduction: $ArgMax_{X_H} P(X_H \mid X_E)$.

# Solution

Multiconnected Networks
Variable Elimination
Conditioning
Junction Tree Algorithm
Approximate Inference
Loopy Propagation
Stochastic Simulation
Most probable explanation
Continuous variables
Applications
Information Validation
Reliability Analysis
References

- One way to solve the MPE problem is based on a modified version of the variable elimination algorithm
- Total abduction, we substitute the summations by maximizations:

$$max_{X_H, X_R} P(X_H, X_R \mid X_E)$$

- Partial abduction, we sum over the variables that are not in the explanation subset and maximize over the explanation subset:

$$max_{X_H} \sum_{X_R} P(X_H, X_R \mid X_E)$$

# Continuous variables

Multiconnected Networks
Variable Elimination
Conditioning
Junction Tree Algorithm
Approximate Inference
Loopy Propagation
Stochastic Simulation
Most probable explanation
Continuous variables
Applications
Information Validation
Reliability Analysis
References

- When dealing with continuos variables, one option is to discretize them; however, this could result in a loss of information or in an unnecessary increase in computational requirements
- Another alternative is to operate directly on the continuous distributions
- Probabilistic inference techniques have been developed for some distribution families, in particular for Gaussian variables

Multiconnected Networks
Variable Elimination
Conditioning
Junction Tree Algorithm
Approximate Inference
Loopy Propagation
Stochastic Simulation
Most probable explanation
Continuous variables
Applications
Information Validation
Reliability Analysis
References

# Gaussian variables

The basic algorithm makes the following assumptions:

1. The structure of the network is a polytree.
2. All the sources of uncertainty are Gaussians and uncorrelated.
3. There is a linear relationship between each variable and its parents:

$$X = b_1 U_1 + b_2 U_2 + ... + b_n U_n + W_X$$

Where $U_i$ are parents of variable $X$, $b_i$ are constant coefficients and $W_X$ represents Gaussian noise with a zero mean.

Multiconnected Networks
Variable Elimination
Conditioning
Junction Tree Algorithm
Approximate Inference
Loopy Propagation
Stochastic Simulation
Most probable explanation
Continuous variables
Applications
Information Validation
Reliability Analysis
References

# Inference

- The inference procedure is analogous to belief propagation in discrete BNs, but instead of propagating probabilities, it propagates means and standard deviations
- The posterior probability of a variable can be written as:

$$P(X \mid E) = N(\mu_X, \sigma_X)$$

  Where $\mu_X$ and $\sigma_X$ are the mean and standard deviation of $X$ given the evidence $E$, respectively.
- We calculate the mean and standard deviation for each variable via a propagation algorithm

Multiconnected Networks
Variable Elimination
Conditioning
Junction Tree Algorithm
Approximate Inference
Loopy Propagation
Stochastic Simulation
Most probable explanation
Continuous variables
Applications
Information Validation
Reliability Analysis
References

# Propagation

- Each variable sends to its parent variable $i$:

$$\mu_i^- = (1/b_i)[\mu_\lambda - \sum_{k \neq i} b_k \mu_k^+] \tag{8}$$

$$\sigma_i^- = (1/b_i^2)[\sigma_\lambda - \sum_{k \neq i} b_k^2 \sigma_k^+] \tag{9}$$

- Each variable sends to its child node $j$:

$$\mu_j^+ = \frac{\sum_{k \neq j} \mu_k^- / \sigma_k + \mu_\pi / \sigma_\pi]}{\sum_{k \neq} 1/\sigma_k^- + \mu_\pi / \sigma_\pi} \tag{10}$$

$$\sigma_j^+ = [\sum_{k \neq j} 1/\sigma_k^- + 1/\sigma_\pi]^{-1} \tag{11}$$

# Propagation

Multiconnected Networks
Variable Elimination
Conditioning
Junction Tree Algorithm
Approximate Inference
Loopy Propagation
Stochastic Simulation
Most probable explanation
Continuous variables
Applications
Information Validation
Reliability Analysis
References

- Each variable integrates the messages it receives from its sons and parents via the following equations:

$$\mu_\pi = \sum_i b_i \mu_i^+ \tag{12}$$

$$\sigma_\pi = \sum_i b_i^2 \sigma_i^+ \tag{13}$$

$$\mu_\lambda = \sigma_\lambda \sum_j \mu_j^- / \sigma_j^- \tag{14}$$

$$\sigma_\lambda = [\sum_j 1/\sigma_j^-]^{-1} \tag{15}$$

Multiconnected Networks
Variable Elimination
Conditioning
Junction Tree Algorithm
Approximate Inference
Loopy Propagation
Stochastic Simulation
Most probable explanation
Continuous variables
Applications
Information Validation
Reliability Analysis
References

- Finally, each variable obtains its mean and standard deviation by combining the information from its parent and children nodes:

$$\mu_X = \frac{\sigma_\pi \mu_\lambda + \sigma_\lambda \mu_\pi}{\sigma_\pi + \sigma_\lambda} \tag{16}$$

$$\sigma_X = \frac{\sigma_\pi \sigma_\lambda}{\sigma_\pi + \sigma_\lambda} \tag{17}$$

- Propagation for other distributions is more difficult, as they do not have the same properties of the Gaussian; in particular, the product of Gaussians is also a Gaussian

# Information validation

Multiconnected Networks
Variable Elimination
Conditioning
Junction Tree Algorithm
Approximate Inference
Loopy Propagation
Stochastic Simulation
Most probable explanation
Continuous variables
Applications
Information Validation
Reliability Analysis
References

- Many systems use information to make decisions; if this information is erroneous it could lead to non-optimal decisions
- In many applications there are different sources of information, i.e. sensors, which are not independent; the information from one source gives us clues about the other sources
- If we can represent these dependencies between the different sources, then we can use it to detect possible errors and avoid erroneous decisions

Multiconnected Networks
Variable Elimination
Conditioning
Junction Tree Algorithm
Approximate Inference
Loopy Propagation
Stochastic Simulation
Most probable explanation
Continuous variables
Applications
Information Validation
Reliability Analysis
References

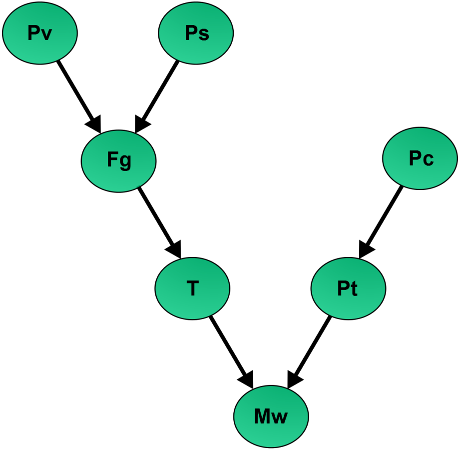
# Algorithm

- The algorithm starts by building a model of the dependencies between sources of information (variables) represented as a Bayesian network
- The validation is done in two phases. In the first phase, potential faults are detected by comparing the actual value with the one predicted from the related variables
- In the second phase, the real faults are isolated by constructing an additional Bayesian network based on the Markov blanket property

Multiconnected Networks
Variable Elimination
Conditioning
Junction Tree Algorithm
Approximate Inference
Loopy Propagation
Stochastic Simulation
Most probable explanation
Continuous variables
Applications
Information Validation
Reliability Analysis
References

# Fault detection

- Build a probabilistic model relating all the variables in the application domain
- Example - gas turbine:



# Fault detection

Multiconnected Networks
Variable Elimination
Conditioning
Junction Tree Algorithm
Approximate Inference
Loopy Propagation
Stochastic Simulation
Most probable explanation
Continuous variables
Applications
Information Validation
Reliability Analysis
References

- Suppose it is required to validate the temperature measurements in the turbine
- By reading the values of the rest of the sensors, and applying probability propagation, it is possible to calculate a posterior probability distribution of the temperature given all the evidence, i.e., $P(T \mid Mw, P, Fg, Pc, Pv, Ps)$
- If the real observed value coincides with a *valid* value – that has a *high* probability, then the sensor is considered correct; otherwise it is considered faulty

# Fault detection

Multiconnected Networks
Variable Elimination
Conditioning
Junction Tree Algorithm
Approximate Inference
Loopy Propagation
Stochastic Simulation
Most probable explanation
Continuous variables
Applications
Information Validation
Reliability Analysis
References

- If a validation of a single sensor is made using a faulty sensor, then a faulty validation can be expected
- In the example above, what happens if $T$ is validated using a faulty $Mw$ sensor?
- By applying this validation procedure, we may only detect a faulty condition, but we are not able to identify which is the real faulty sensor – *apparent fault*
- An isolation stage is needed

# Fault isolation

Multiconnected Networks
Variable Elimination
Conditioning
Junction Tree Algorithm
Approximate Inference
Loopy Propagation
Stochastic Simulation
Most probable explanation
Continuous variables
Applications
Information Validation
Reliability Analysis
References

- The isolation phase is based on the *Markov Blanket* (MB) property
- The *Extended Markov Blanket* of a node $X$ ($EMB(X)$) as the set of sensors formed by the sensor itself plus its MB
- For example, $EMB(Fg) = \{Fg, Pv, Ps, T\}$
- Utilizing this property, if a fault exists in one of the sensors, it will be revealed in all of the sensors in its EMB. On the contrary, if a fault exists outside a sensors' EMB, it will not affect the estimation of that sensor
- The EMB is used to create a *fault isolation* module that distinguishes the *real faults* from the apparent faults

Multiconnected Networks
Variable Elimination
Conditioning
Junction Tree Algorithm
Approximate Inference
Loopy Propagation
Stochastic Simulation
Most probable explanation
Continuous variables
Applications
Information Validation
Reliability Analysis
References

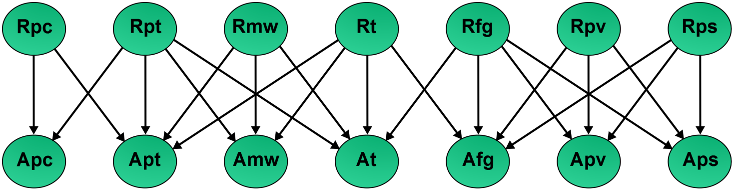
# Fault isolation theory

1. If $S = \phi$ there are no faults.
2. If $S$ is equal to the EMB of a sensor $X$, and there is no other EMB which is a subset of $S$, then there is a *single real fault* in $X$.
3. If $S$ is equal to the EMB of a sensor $X$, and there are one or more EMBs which are subsets of $S$, then there is a real fault in $X$, and possibly, real faults in the sensors whose EMBs are subsets of $S$. In this case, there are possibly *multiple indistinguishable* real faults.
4. If $S$ is equal to the union of several EMBs and the combination is unique, then there are *multiple distinguishable* real faults in all the sensors whose EMB are in $S$.
5. If none of the above cases is satisfied, then there are multiple faults but they can not be distinguished

Multiconnected Networks
Variable Elimination
Conditioning
Junction Tree Algorithm
Approximate Inference
Loopy Propagation
Stochastic Simulation
Most probable explanation
Continuous variables
Applications
Information Validation
Reliability Analysis
References

# Isolation network

- The isolation network is formed by two levels:
    - The root nodes represent the real faults, where there is one per sensor or variable
    - The lower level is formed by one node representing the apparent fault for each variable. Notice that the arcs are defined by the EMB of each variable



# Reliability analysis

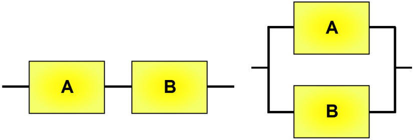
Multiconnected Networks
Variable Elimination
Conditioning
Junction Tree Algorithm
Approximate Inference
Loopy Propagation
Stochastic Simulation
Most probable explanation
Continuous variables
Applications
Information Validation
Reliability Analysis
References

- In the reliability analysis of a complex system, a common approach is to divide the system into smaller elements, units, subsystems, or components
- This subdivision generates a "block diagram" that is similar to the description of the system in operation
- For each element, the *failure rate* is specified, and based on these, the reliability of the complete system is obtained
- Traditional techniques assume that faults are independent

# Reliability modeling with BN

Multiconnected Networks
Variable Elimination
Conditioning
Junction Tree Algorithm
Approximate Inference
Loopy Propagation
Stochastic Simulation
Most probable explanation
Continuous variables
Applications
Information Validation
Reliability Analysis
References

- In a block diagram representation there are two basic structures: serial and parallel components
- A serial structure implies that the two components should operate correctly for the system to function
- In parallel structures, it is sufficient for one of the components to operate for the system to function

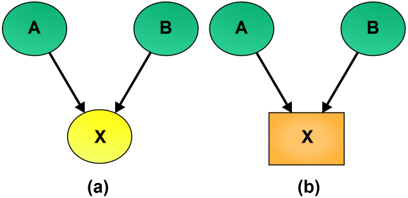



(a) (b)

# Basic structures

Multiconnected Networks
Variable Elimination
Conditioning
Junction Tree Algorithm
Approximate Inference
Loopy Propagation
Stochastic Simulation
Most probable explanation
Continuous variables
Applications
Information Validation
Reliability Analysis
References

- The basic series and parallel block diagrams can be represented with a Bayesian network
- The structure is the same in both cases, the difference is the conditional probability matrix



# CPTs

Multiconnected Networks
Variable Elimination
Conditioning
Junction Tree Algorithm
Approximate Inference
Loopy Propagation
Stochastic Simulation
Most probable explanation
Continuous variables
Applications
Information Validation
Reliability Analysis
References

- Series (AND):

| X | $A, B$ | $A, \neg B$ | $\neg A, B$ | $\neg A, \neg B$ |
|---|---|---|---|---|
| Success | 1 | 0 | 0 | 0 |
| Failure | 0 | 1 | 1 | 1 |

- Parallel (OR):

| X | $A, B$ | $A, \neg B$ | $\neg A, B$ | $\neg A, \neg B$ |
|---|---|---|---|---|
| Success | 1 | 1 | 1 | 0 |
| Failure | 0 | 0 | 0 | 1 |

# Extending the basic models

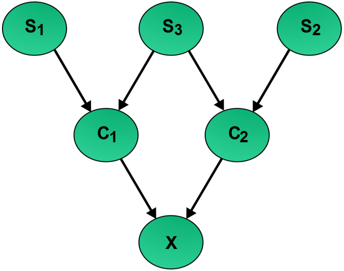
Multiconnected Networks
Variable Elimination
Conditioning
Junction Tree Algorithm
Approximate Inference
Loopy Propagation
Stochastic Simulation
Most probable explanation
Continuous variables
Applications
Information Validation
Reliability Analysis
References

- The BN representation of the basic serial/parallel cases can be directly generalized to represent any block diagram that can be reduced to a set of serial and parallel combinations of components
- There are some structures that can not be decomposed to a serial/parallel combination, such as a *bridge*. However, it is also possible to model these cases using BNs

Multiconnected Networks
Variable Elimination
Conditioning
Junction Tree Algorithm
Approximate Inference
Loopy Propagation
Stochastic Simulation
Most probable explanation
Continuous variables
Applications
Information Validation
Reliability Analysis
References

# Example

- Suppose that a system has two components that are affected by three possible failure sources. Source $S_1$ affects component $C_1$, source $S_2$ affects component $C_2$, and source $S_3$ affects both components (common cause)
- In the BN, the CPT for all three non root nodes ($C_1$, $C_2$, $X$) is equivalent to that of a serial component combination



Multiconnected Networks
Variable Elimination
Conditioning
Junction Tree Algorithm
Approximate Inference
Loopy Propagation
Stochastic Simulation
Most probable explanation
Continuous variables
Applications
Information Validation
Reliability Analysis
References

# Book

Sucar, L. E, *Probabilistic Graphical Models*, Springer 2015 – Chapter 7

Multiconnected Networks
Variable Elimination
Conditioning
Junction Tree Algorithm
Approximate Inference
Loopy Propagation
Stochastic Simulation
Most probable explanation
Continuous variables
Applications
Information Validation
Reliability Analysis
References

# Additional Reading (2)


- Ibargüengoytia, P.H., Sucar L.E., Vadera, S.: A Probabilistic Model for Sensor Validation. In: Proceedings of the Twelfth Conference on Uncertainty in Artificial Intelligence UAI-96, pp. 332–339. Morgan Kaufmann Publishers Inc. (1996)
- Ibargüengoytia, P.H., Vadera, S., Sucar L.E.: A Probabilistic Model for Information Validation. British Computer Journal. 49(1), 113–126 (2006)
- Jensen, F.V., Andersen, S.K.: Approximations in Bayesian Belief Universes for Knowledge Based Systems. In: Proceedings of the Sixth Conference on Uncertainty in Artificial Intelligence UAI-90, pp. 162–169. Elsevier, New York (1990)
- Jensen, F.V.: Bayesian Networks and Decision Graphs. Springer-Verlag, New York (2001)

Multiconnected Networks
Variable Elimination
Conditioning
Junction Tree Algorithm
Approximate Inference
Loopy Propagation
Stochastic Simulation
Most probable explanation
Continuous variables
Applications
Information Validation
Reliability Analysis
References

# Additional Reading (3)


- Korb, K.B., Nicholson, A.E.: Bayesian Artificial Intelligence. Second edition, CRC Press, Boca Raton, Florida (2010)
- Lauritzen, S., Spiegelhalter D. J.: Local Computations with Probabilities on Graphical Structures and their Application to Expert Systems. Journal of the Royal Statistical Society. Series B. 50(2), 157–224 (1988)
- Moral, S., Rumi, R., Salmerón, A.: Mixtures of Truncated Exponentials in Hybrid Bayesian Networks. Symbolic and Quantitative Approaches to Reasoning with Uncertainty. 2143, 156–167 (2001)
- Murphy K.P., Weiss, Y., Jordan, M.: Loopy Belief Propagation for Approximate Inference: An Empirical Study. In: Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence, pp. 467–475.

Multiconnected Networks
Variable Elimination
Conditioning
Junction Tree Algorithm
Approximate Inference
Loopy Propagation
Stochastic Simulation
Most probable explanation
Continuous variables
Applications
Information Validation
Reliability Analysis
References

# Additional Reading (4)


- Pourret, O. , Naim, P., Marcot, B. (Eds.): Bayesian belief networks: a practical guide to applications. Wiley, New Jersey (2008)
- Shenoy P., Shafer, G.: Axioms for Probability and Belief-Function Propagation. In: Uncertainty in Artificial Intelligence 4, pp. 169–198. Elsevier, New York (1990)
- Torres-Toledano, J.G., Sucar, L.E.: Bayesian Networks for Reliability Analysis of Complex Systems. In H. Coelho (Ed.), IBERAMIA'98, Lecture Notes in Computer Science, Vol.1484, Springer-Verlag, Berlin, pp. 195–206 (1998)