Introduction to Bayesian Networks

A Tutorial for the 66th MORS Symposium

23 - 25 June 1998
Naval Postgraduate School
Monterey, California

Dennis M. Buede
Joseph A. Tatman
Terry A. Bresnick

Overview

- Day 1
  - Motivating Examples
  - Basic Constructs and Operations
- Day 2
  - Propagation Algorithms
  - Example Application
- Day 3
  - Learning
  - Continuous Variables
  - Software

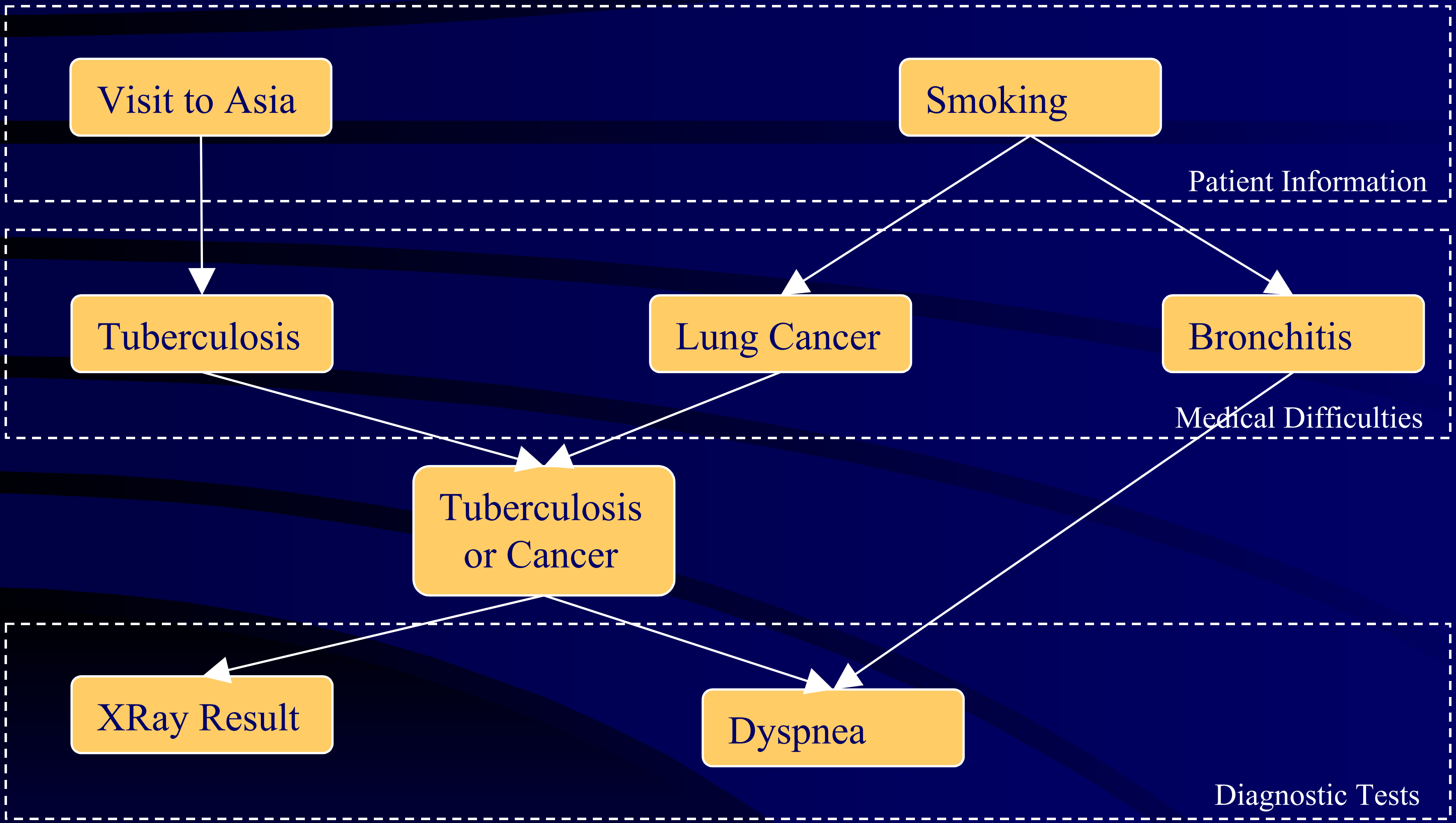
Day One Outline

- Introduction
- Example from Medical Diagnostics
- Key Events in Development
- Definition
- Bayes Theorem and Influence Diagrams
- Applications

Why the Excitement?

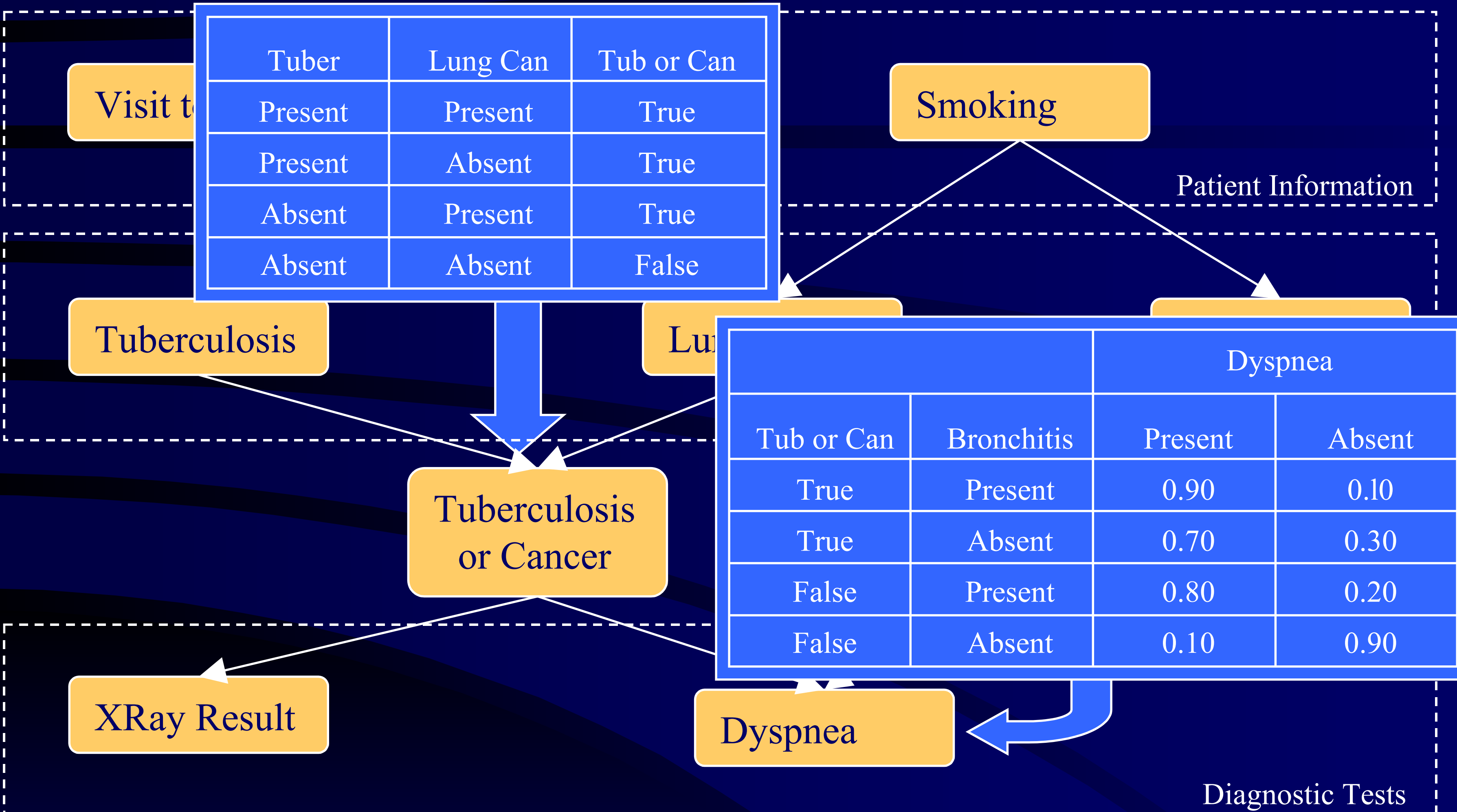
- What are they?
  - Bayesian nets are a network-based framework for representing and analyzing models involving uncertainty
- What are they used for?
  - Intelligent decision aids, data fusion, 3-E feature recognition, intelligent diagnostic aids, automated free text understanding, data mining
- Where did they come from?
  - Cross fertilization of ideas between the artificial intelligence, decision analysis, and statistic communities
- Why the sudden interest?
  - Development of propagation algorithms followed by availability of easy to use commercial software
  - Growing number of creative applications
- How are they different from other knowledge representation and probabilistic analysis tools?
  - Different from other knowledge-based systems tools because uncertainty is handled in mathematically rigorous yet efficient and simple way
  - Different from other probabilistic analysis tools because of network representation of problems, use of Bayesian statistics, and the synergy between these

Example from Medical Diagnostics



- **Network represents a knowledge structure that models the relationship between medical difficulties, their causes and effects, patient information and diagnostic tests**

Example from Medical Diagnostics

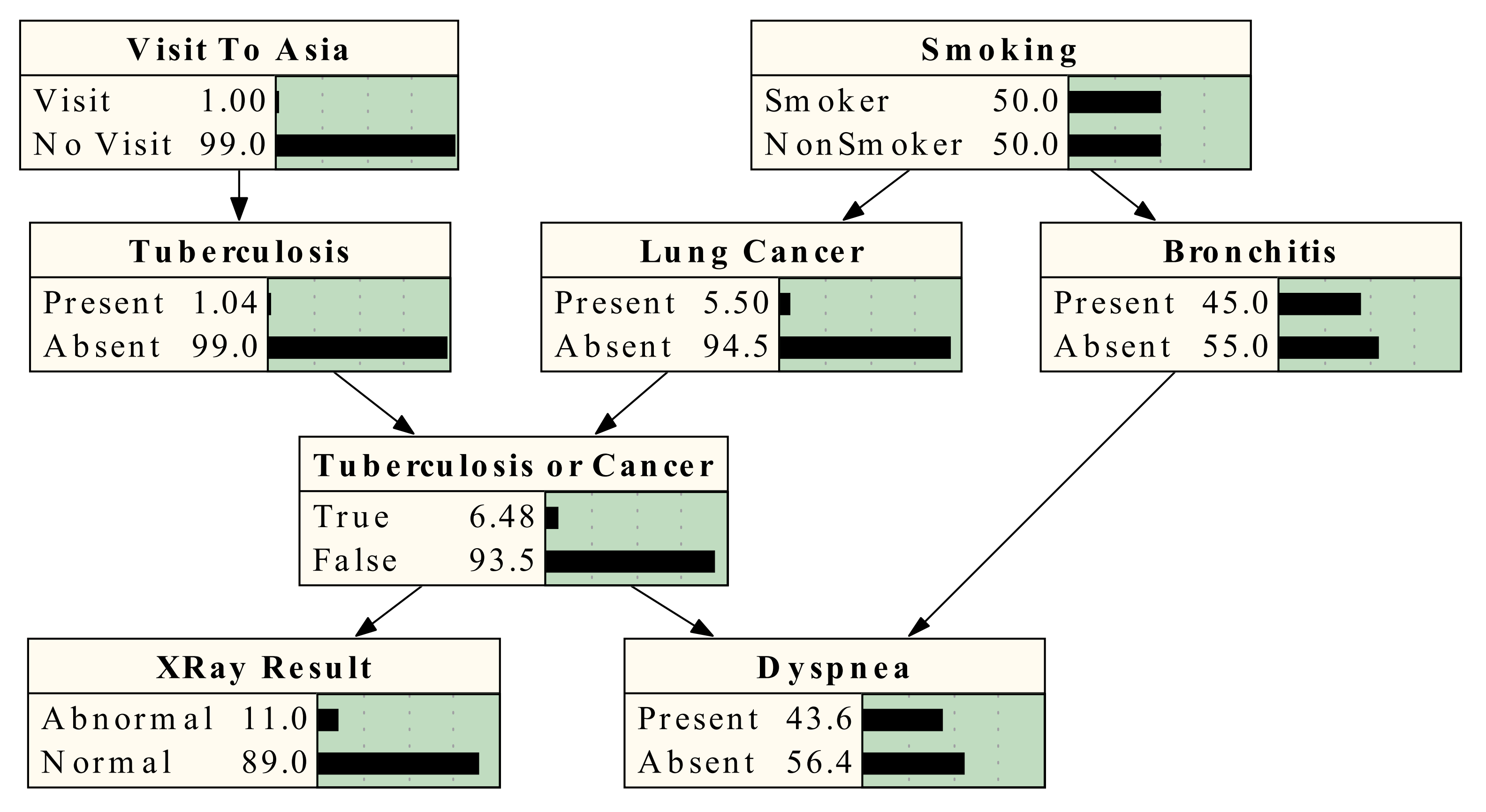


Tuber	Lung Can	Tub or Can
Present	Present	True
Present	Absent	True
Absent	Present	True
Absent	Absent	False

		Dyspnea	
Tub or Can	Bronchitis	Present	Absent
True	Present	0.90	0.10
True	Absent	0.70	0.30
False	Present	0.80	0.20
False	Absent	0.10	0.90

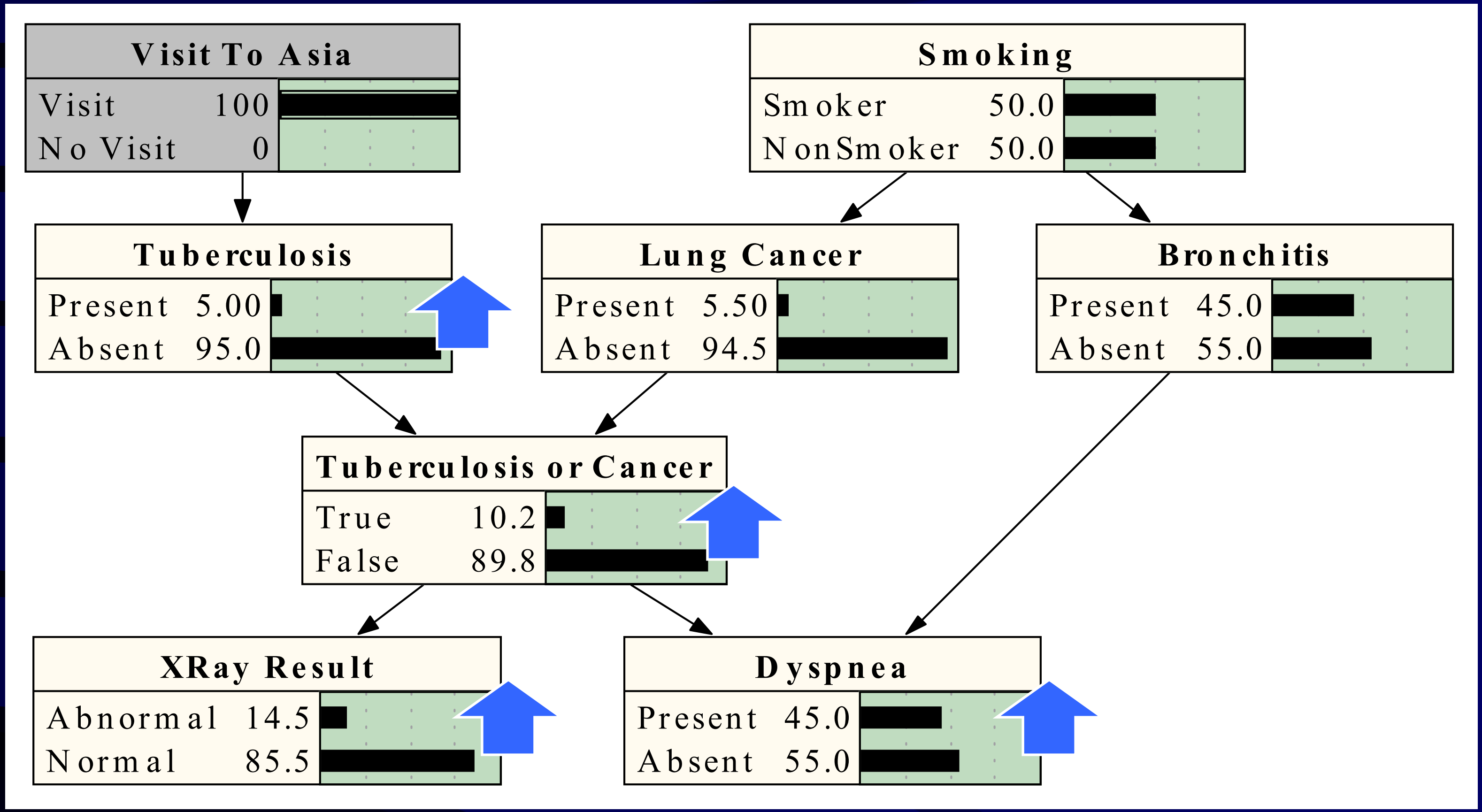
- **Relationship knowledge is modeled by deterministic functions, logic and conditional probability distributions**

Example from Medical Diagnostics



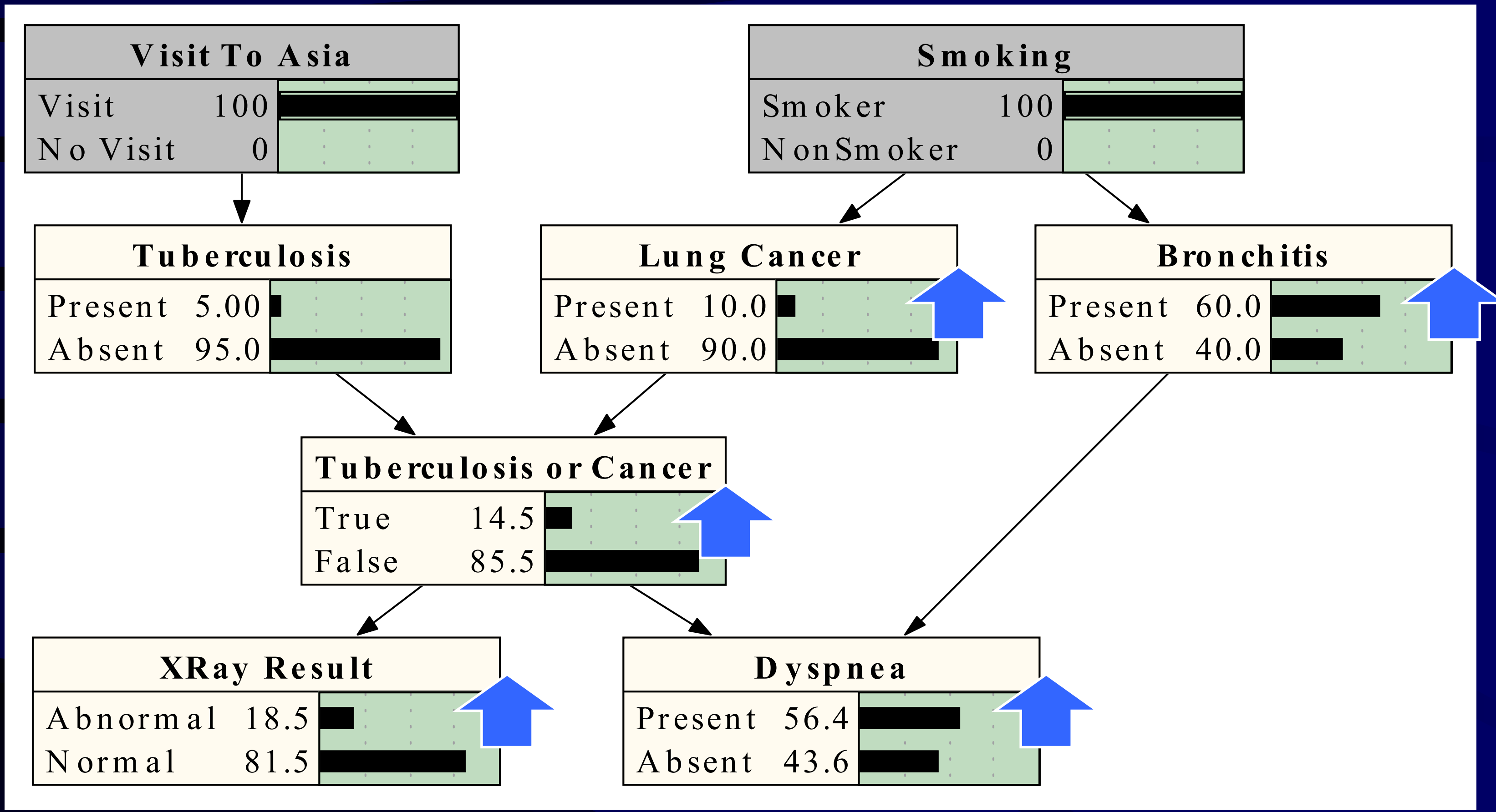
- **Propagation algorithm processes relationship information to provide an unconditional or marginal probability distribution for each node**
- **The unconditional or marginal probability distribution is frequently called the belief function of that node**

Example from Medical Diagnostics



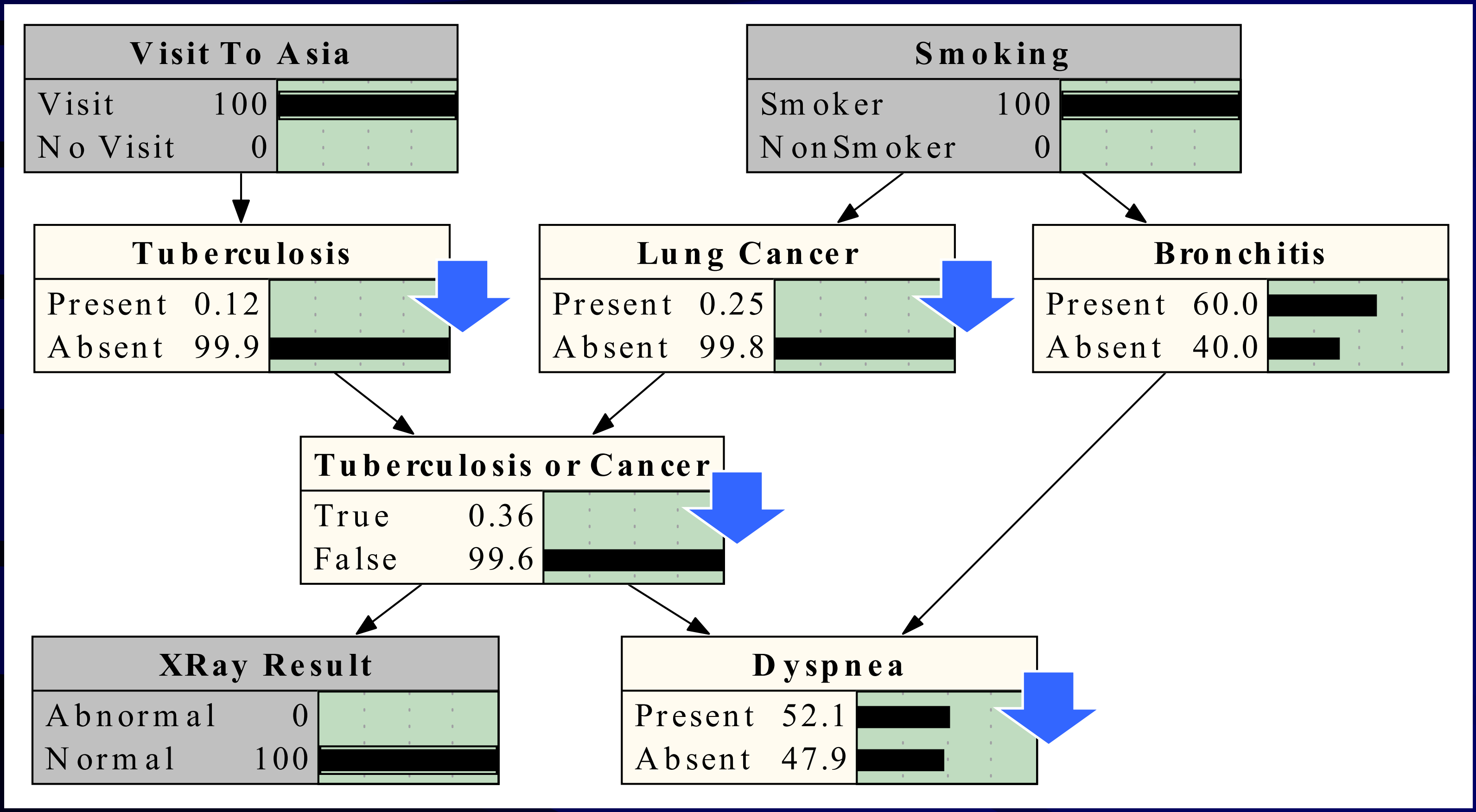
- **As a finding is entered, the propagation algorithm updates the beliefs attached to each relevant node in the network**
- **Interviewing the patient produces the information that “Visit to Asia” is “Visit”**
- **This finding propagates through the network and the belief functions of several nodes are updated**

Example from Medical Diagnostics



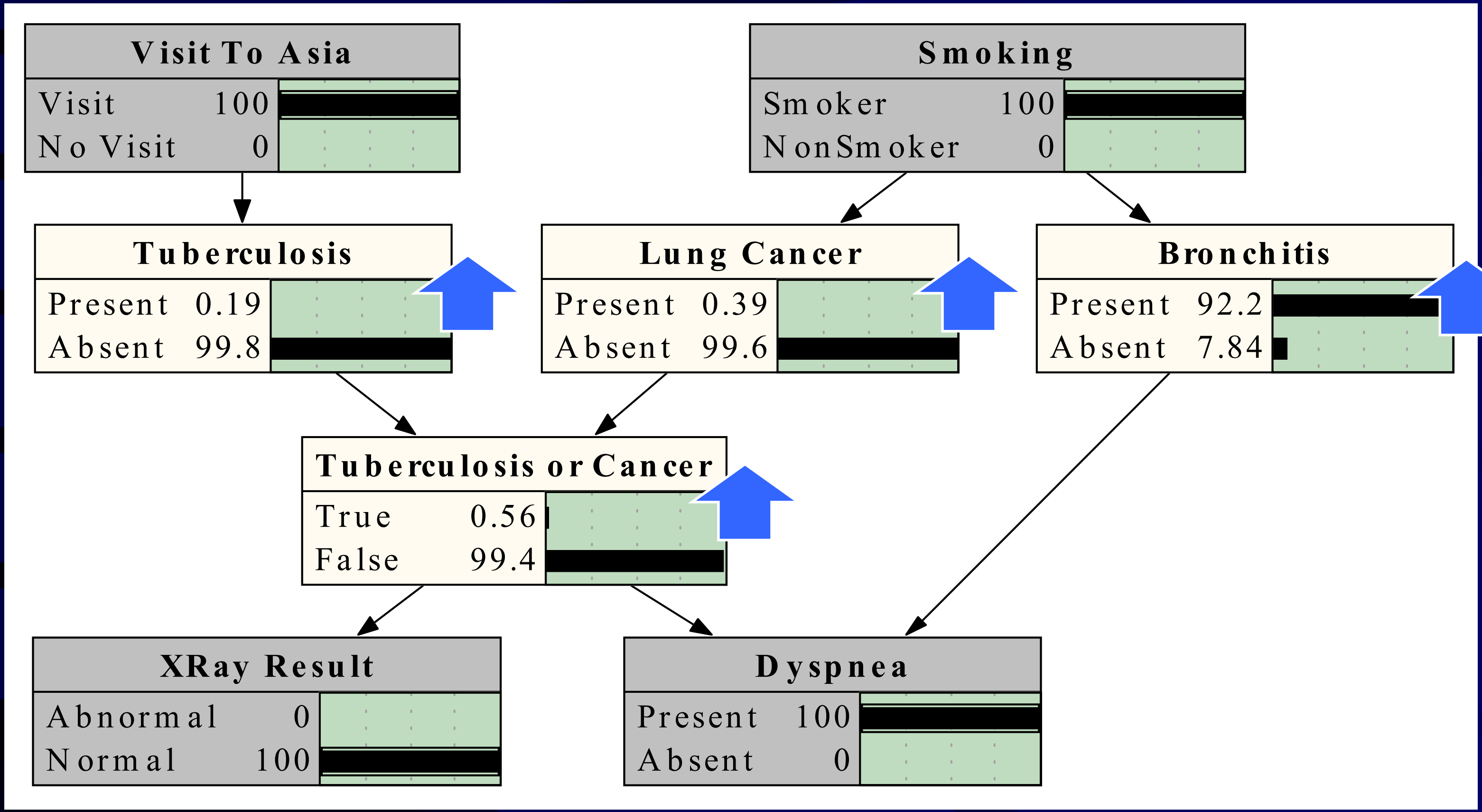
- **Further interviewing of the patient produces the finding "Smoking" is "Smoker"**
- **This information propagates through the network**

Example from Medical Diagnostics



- Finished with interviewing the patient, the physician begins the examination
- The physician now moves to specific diagnostic tests such as an X-Ray, which results in a “Normal” finding which propagates through the network
- Note that the information from this finding propagates backward and forward through the arcs

Example from Medical Diagnostics



- The physician also determines that the patient is having difficulty breathing, the finding “Present” is entered for “Dyspnea” and is propagated through the network
- The doctor might now conclude that the patient has bronchitis and does not have tuberculosis or lung cancer

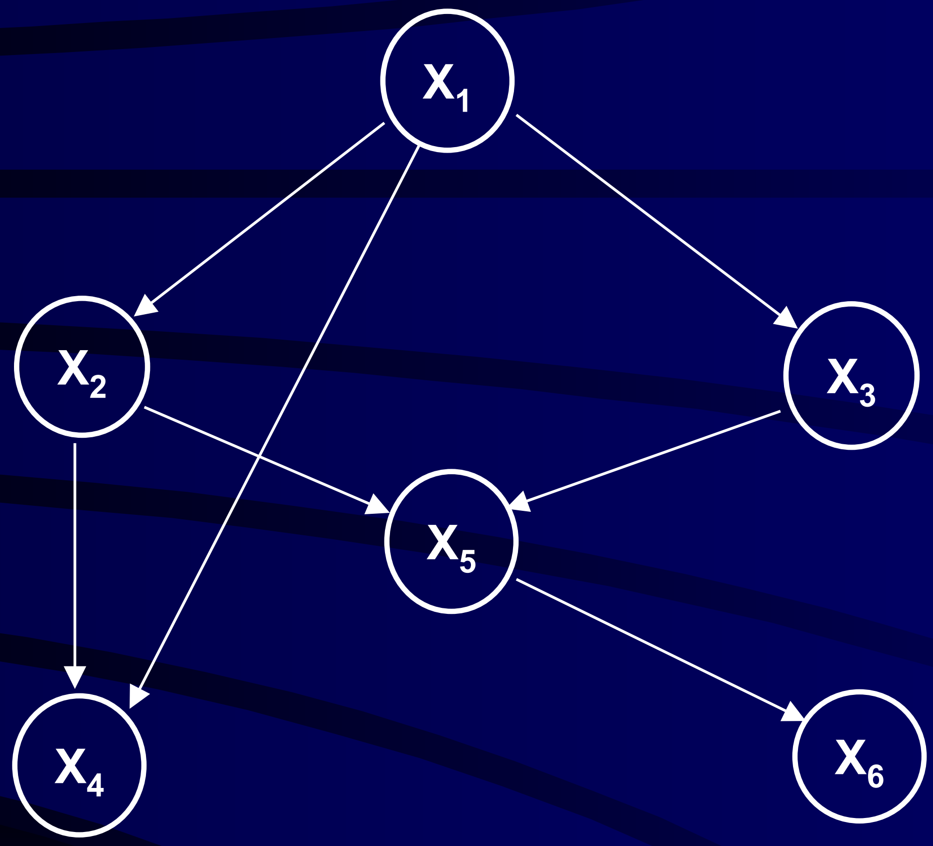
Applications

- Industrial
  - Processor Fault Diagnosis - by Intel
  - Auxiliary Turbine Diagnosis - GEMS by GE
  - Diagnosis of space shuttle propulsion systems - VISTA by NASA/Rockwell
  - Situation assessment for nuclear power plant - NRC
- Military
  - Automatic Target Recognition - MITRE
  - Autonomous control of unmanned underwater vehicle - Lockheed Martin
  - Assessment of Intent
- Medical Diagnosis
  - Internal Medicine
  - Pathology diagnosis - Intellipath by Chapman & Hall
  - Breast Cancer Manager with Intellipath
- Commercial
  - Financial Market Analysis
  - Information Retrieval
  - Software troubleshooting and advice - Windows 95 & Office 97
  - Pregnancy and Child Care - Microsoft
  - Software debugging - American Airlines' SABRE online reservation system

Definition of a Bayesian Network

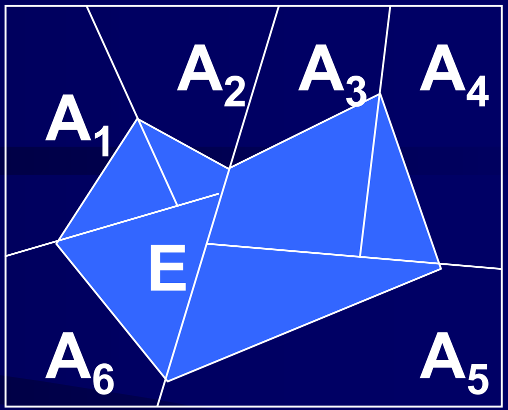
- Factored joint probability distribution as a directed graph:
  - structure for representing knowledge about uncertain variables
  - computational architecture for computing the impact of evidence on beliefs
- Knowledge structure:
  - variables are depicted as nodes
  - arcs represent probabilistic dependence between variables
  - conditional probabilities encode the strength of the dependencies
- Computational architecture:
  - computes posterior probabilities given evidence about selected nodes
  - exploits probabilistic independence for efficient computation

Sample Factored Joint Distribution



$$p(x_1, x_2, x_3, x_4, x_5, x_6) = p(x_6 \mid x_5)\, p(x_5 \mid x_3, x_2)\, p(x_4 \mid x_2, x_1)\, p(x_3 \mid x_1)\, p(x_2 \mid x_1)\, p(x_1)$$

Bayes Rule

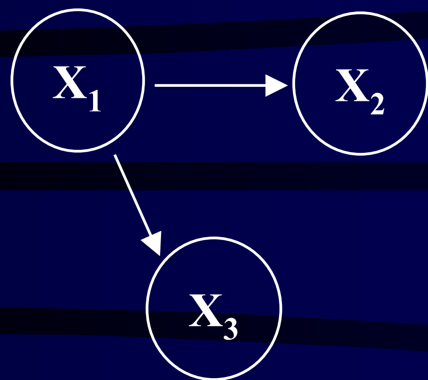


$$p(A \mid B) = \frac{p(A,B)}{p(B)} = \frac{p(B \mid A)p(A)}{p(B)}$$

$$p(A_i \mid E) = \frac{p(E \mid A_i)p(A_i)}{p(E)} = \frac{p(E \mid A_i)p(A_i)}{\sum_i p(E \mid A_i)p(A_i)}$$

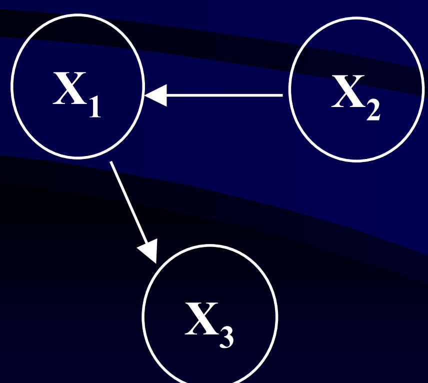
- **Based on definition of conditional probability**
- **$p(A_i|E)$ is posterior probability given evidence E**
- **$p(A_i)$ is the prior probability**
- **$P(E|A_i)$ is the likelihood of the evidence given A_i**
- **p(E) is the preposterior probability of the evidence**

Arc Reversal - Bayes Rule



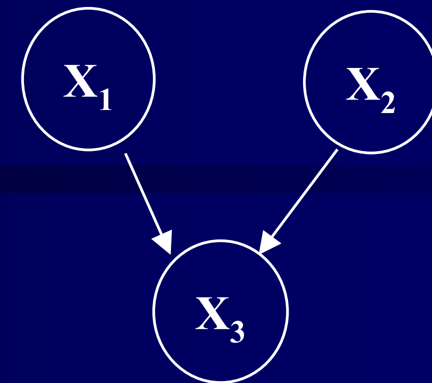
$p(x_1, x_2, x_3) = p(x_3 \mid x_1)\, p(x_2 \mid x_1)\, p(x_1)$

is equivalent to



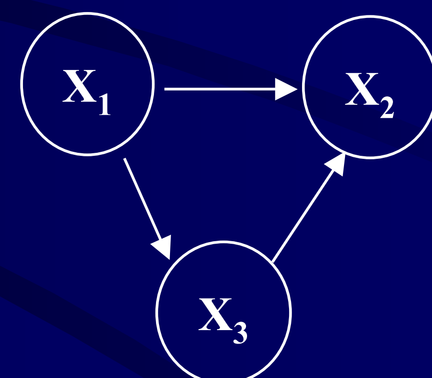
$p(x_1, x_2, x_3) = p(x_3 \mid x_1)\, p(x_2 , x_1)$

$= p(x_3 \mid x_1)\, p(x_1 \mid x_2)\, p(x_2)$



$p(x_1, x_2, x_3) = p(x_3 \mid x_2, x_1)\, p(x_2)\, p(x_1)$

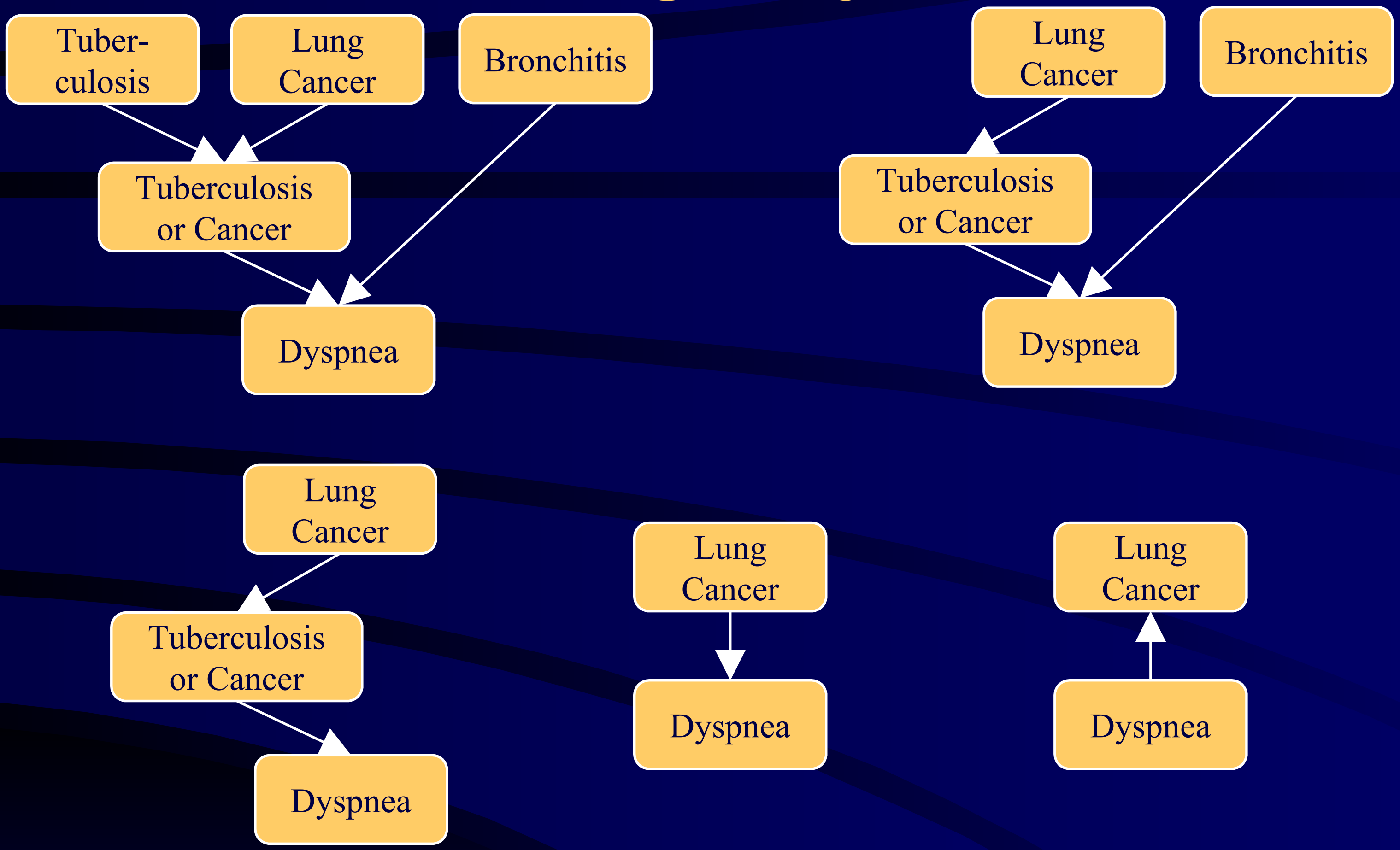
is equivalent to



$p(x_1, x_2, x_3) = p(x_3, x_2 \mid x_1)\, p(x_1)$

$= p(x_2 \mid x_3, x_1)\, p(x_3 \mid x_1)\, p(x_1)$

Inference Using Bayes Theorem



The general probabilistic inference problem is to find the probability of an event given a set of evidence

This can be done in Bayesian nets with sequential applications of Bayes Theorem

Why Not this Straightforward Approach?

- **Entire network must be considered to determine next node to remove**
- **Impact of evidence available only for single node, impact on eliminated nodes is unavailable**
- **Spurious dependencies between variables normally perceived to be independent are created and calculated**
- **Algorithm is inherently sequential, unsupervised parallelism appears to hold most promise for building viable models of human reasoning**

- **In 1986 Judea Pearl published an innovative algorithm for performing inference in Bayesian nets that overcomes these difficulties - TOMMORROW!!!!**

Overview

- Day 1
  – Motivating Examples
  – Basic Constructs and Operations
- Day 2
  – Propagation Algorithms
  – Example Application
- Day 3
  – Learning
  – Continuous Variables
  – Software

Introduction to Bayesian Networks

A Tutorial for the 66th MORS Symposium

23 - 25 June 1998
Naval Postgraduate School
Monterey, California

Dennis M. Buede
Joseph A. Tatman
Terry A. Bresnick

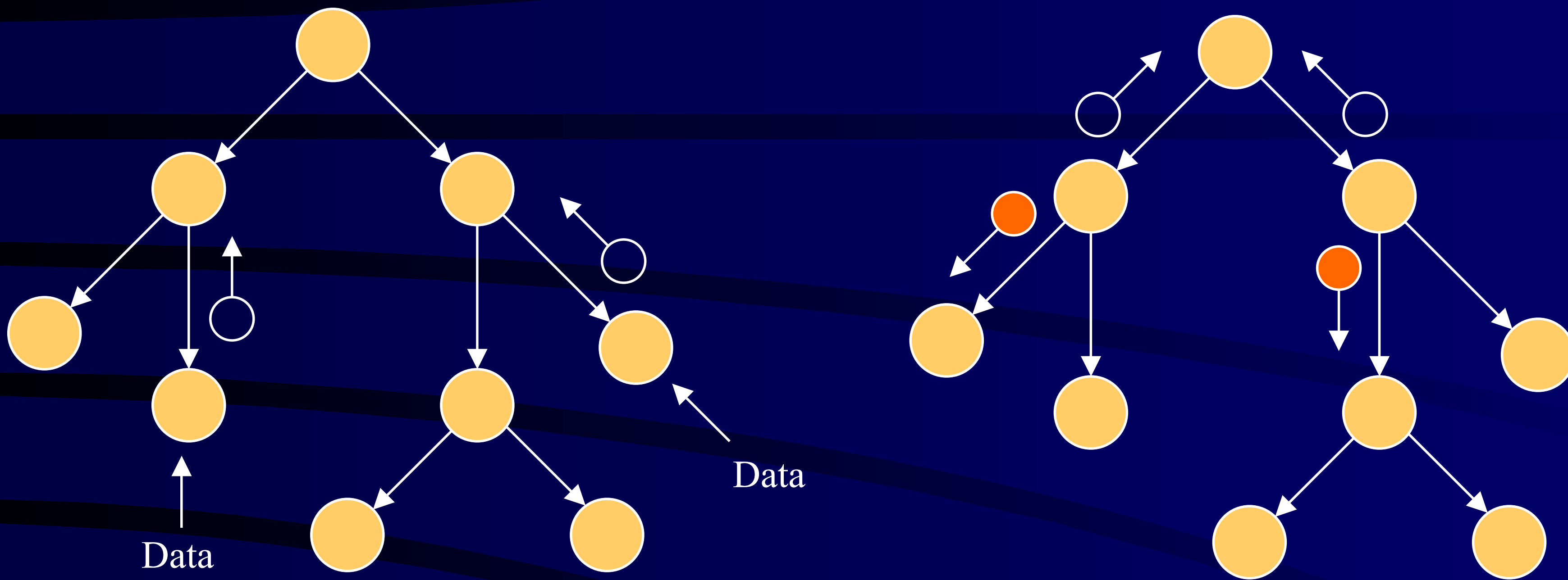
Overview

- Day 1
  - Motivating Examples
  - Basic Constructs and Operations
- Day 2
  - Propagation Algorithms
  - Example Application
- Day 3
  - Learning
  - Continuous Variables
  - Software

Overview of Bayesian Network Algorithms

- Singly vs. multiply connected graphs
- Pearl's algorithm
- Categorization of other algorithms
  - Exact
  - Simulation

Propagation Algorithm Objective

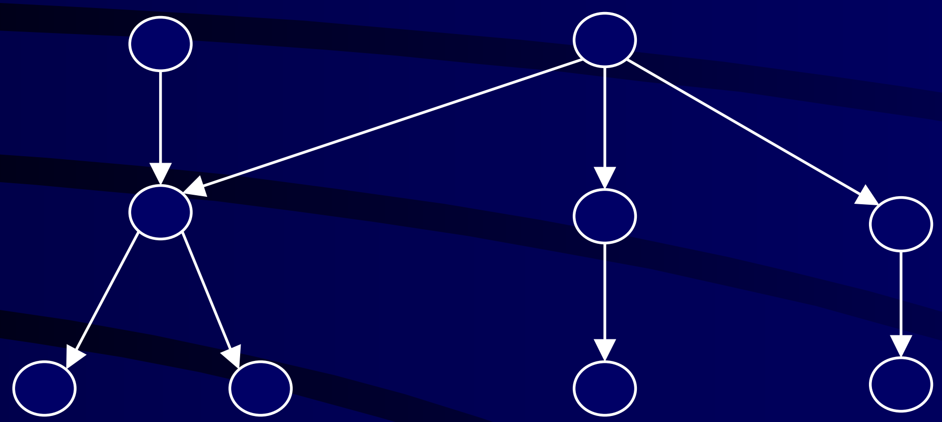


- **The algorithm's purpose is "… fusing and propagating the impact of new evidence and beliefs through Bayesian networks so that each proposition eventually will be assigned a certainty measure consistent with the axioms of probability theory." (Pearl, 1988, p 143)**

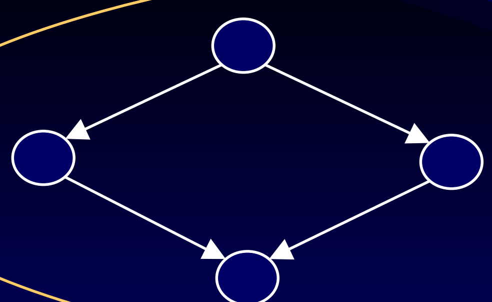
Singly Connected Networks (or Polytrees)

Definition : **A directed acyclic graph (DAG) in which only one semipath (sequence of connected nodes ignoring direction of the arcs) exists between any two nodes.**

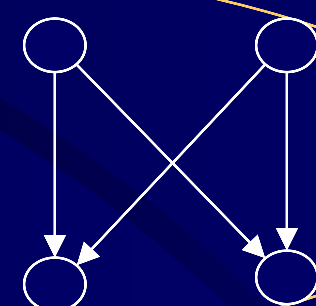
Polytree structure satisfies definition



Multiple parents and/or multiple children



Do not satisfy definition



Notation

X = a random variable (a vector of dimension m); x = a possible value of X
e = evidence (or data), a vector of dimension m
$M_{y|x}$ = p(y|x), the likelihood matrix or conditional probability distribution

$$
\begin{array}{cc}
 & \longrightarrow y \\
= \downarrow x & \begin{bmatrix} p(y_1|x_1) & p(y_2|x_1) & \cdots & p(y_n|x_1) \\ p(y_1|x_2) & p(y_2|x_2) & \cdots & p(y_n|x_2) \\ \cdots & \cdots & \cdots & \\ p(y_1|x_m) & p(y_2|x_m) & \cdots & p(y_n|x_m) \end{bmatrix}
\end{array}
$$

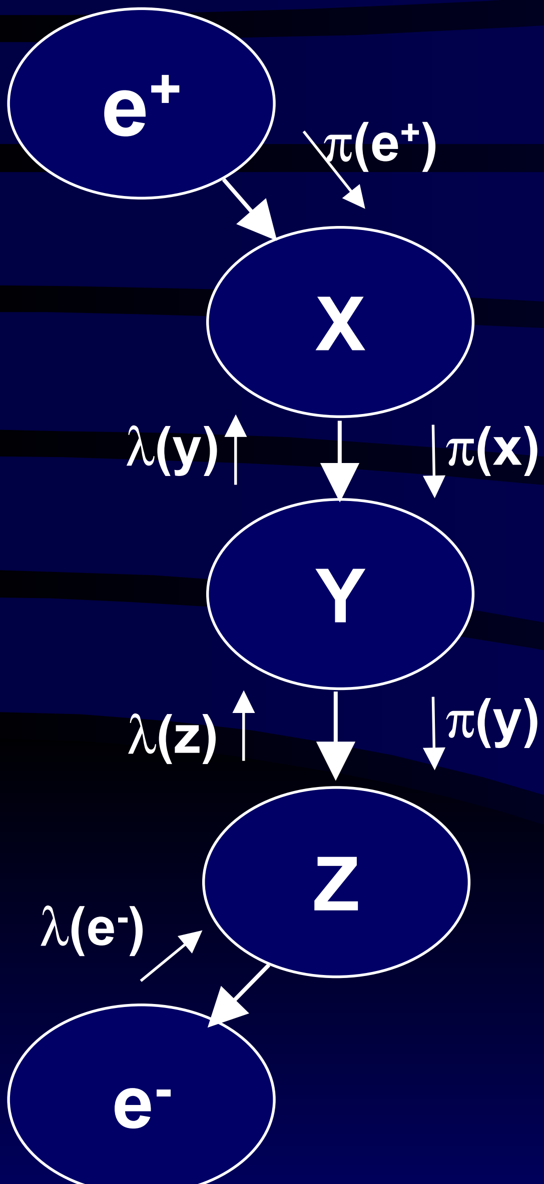
Bel (x) = p(x|e), the posterior (a vector of dimension m)

f(x) $\blacksquare$ g(x) = the term by term product (congruent multiplication) of two vectors, each of dimension m

f(x) $\bullet$ g(x) = the inner (or dot) product of two vectors, or the matrix multiplication of a vector and a matrix

α = a normalizing constant, used to normalize a vector so that its elements sum to 1.0

Bi-Directional Propagation in a Chain



Each node transmits a pi message to its children and a lambda message to its parents.

$Bel(Y) = p(y|e^+, e^-) = \alpha\, \pi(y)^T \blacksquare\, \lambda(y)$

where

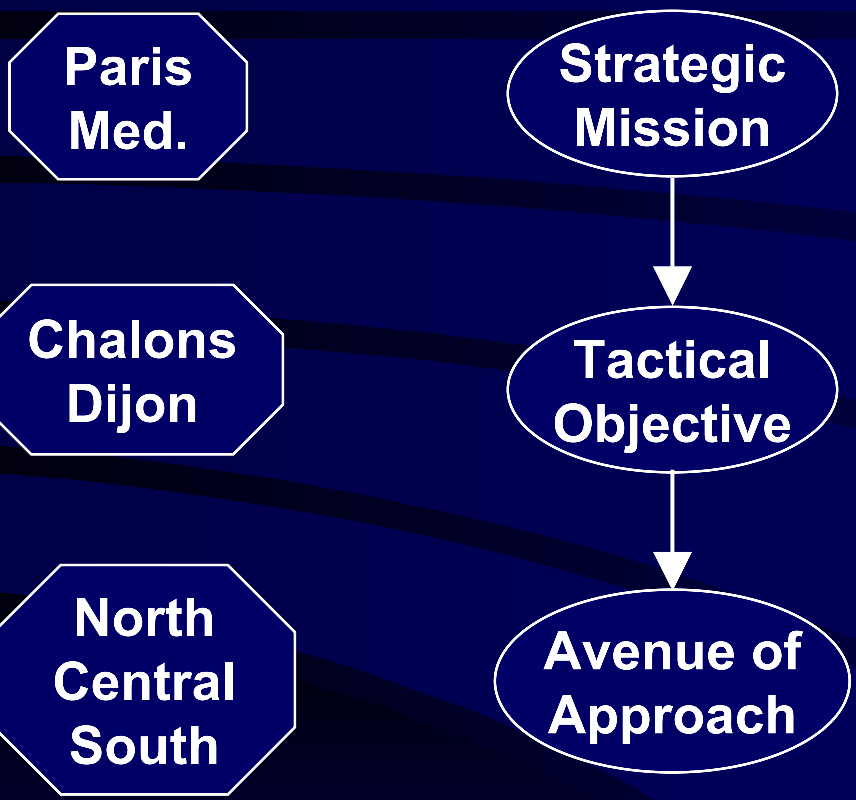
$\pi(y) = p(y|e^+)$, prior evidence; a row vector

$\lambda(y) = p(e^-|y)$, diagnostic or likelihood evidence; a column vector

$$\pi(y) = \Sigma_x\, p(y|x, e^+) \blacksquare\, p(x|\, e^+) = \Sigma_x\, p(y|x) \blacksquare\, \pi(x) = \pi(x) \bullet M_{y|x}$$

$$\lambda(y) = \Sigma_z\, p(e^-|y, z) \blacksquare\, p(z|\, y) = \Sigma_z\, p(e^-|z) \blacksquare\, p(z|\, y) = \Sigma_z\, \lambda(z) \blacksquare\, p(z|\, y) = M_{z|y} \bullet \lambda(z)$$

An Example: Simple Chain

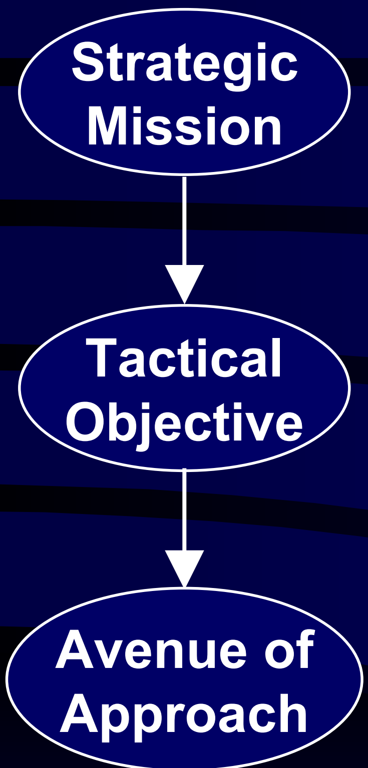


p(Paris) = 0.9
p(Med.) = 0.1

$$M_{TO|SM} = \begin{array}{c} \begin{array}{cc} \text{Ch} & \text{Di} \end{array} \\ \begin{bmatrix} .8 & .2 \\ .1 & .9 \end{bmatrix} \end{array} \begin{array}{l} \\ \text{Pa} \\ \text{Me} \end{array}$$

$$M_{AA|TO} = \begin{array}{c} \begin{array}{ccc} \text{No} & \text{Ce} & \text{So} \end{array} \\ \begin{bmatrix} .5 & .4 & .1 \\ .1 & .3 & .6 \end{bmatrix} \end{array} \begin{array}{l} \\ \text{Ch} \\ \text{Di} \end{array}$$

Sample Chain - Setup



(1) Set all lambdas to be a vector of 1's; $Bel(SM) = \alpha\, \lambda(SM) \blacksquare\, \pi(SM)$

	$\pi(SM)$	$Bel(SM)$	$\lambda(SM)$
Paris	0.9	0.9	1.0
Med.	0.1	0.1	1.0

(2) $\pi(TO) = \pi(SM)\, M_{TO|SM}$; $Bel(TO) = \alpha\, \lambda(TO) \blacksquare\, \pi(TO)$

	$\pi(TO)$	$Bel(TO)$	$\lambda(TO)$
Chalons	0.73	0.73	1.0
Dijon	0.27	0.27	1.0

(3) $\pi(AA) = \pi(TO)\, M_{AA|TO}$; $Bel(AA) = \alpha\, \lambda(AA) \blacksquare\, \pi(AA)$

	$\pi(AA)$	$Bel(AA)$	$\lambda(AA)$
North	0.39	0.40	1.0
Central	0.35	0.36	1.0
South	0.24	0.24	1.0

$$\mathbf{M}_{TO|SM} = \begin{bmatrix} .8 & .2 \\ .1 & .9 \end{bmatrix} \qquad \mathbf{M}_{AA|TO} = \begin{bmatrix} .5 & .4 & .1 \\ .1 & .3 & .6 \end{bmatrix}$$

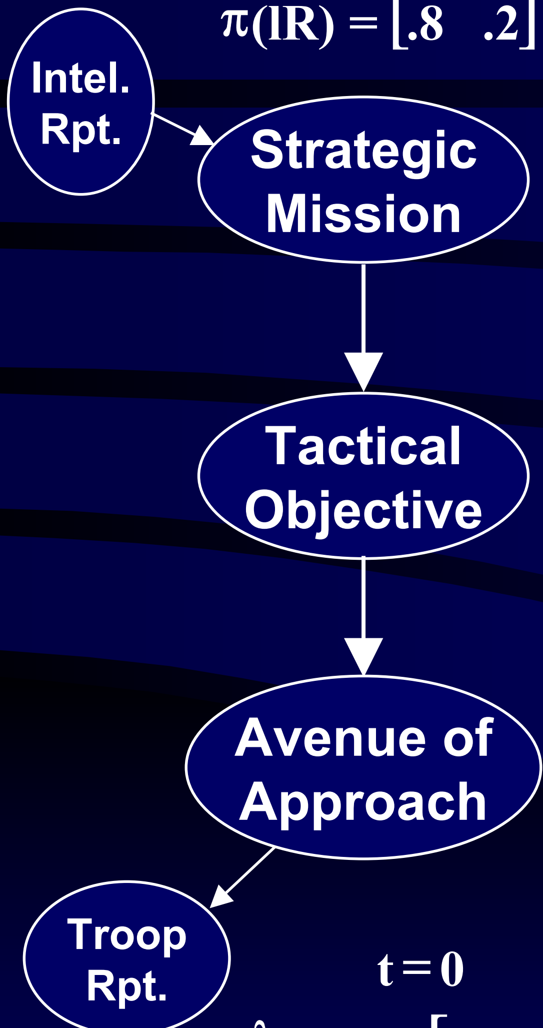
Sample Chain - 1st Propagation

$t = 0$

$\pi(IR) = [.8 \quad .2]$

$t = 1$

$\pi(SM) = \pi(IR)$



	π(SM)	Bel(SM)	λ(SM)
Paris	0.8	0.8	1.0
Med.	0.2	0.2	1.0

	π(TO)	Bel(TO)	λ(TO)
Chalons	0.73	0.73	1.0
Dijon	0.27	0.27	1.0

	π(AA)	Bel(AA)	λ(AA)
North	0.39	0.3	0.5
Central	0.35	0.5	1.0
South	0.24	0.2	0.6

$t = 0$

$\lambda(TR) = [.5 \quad 1 \quad .6]^T$

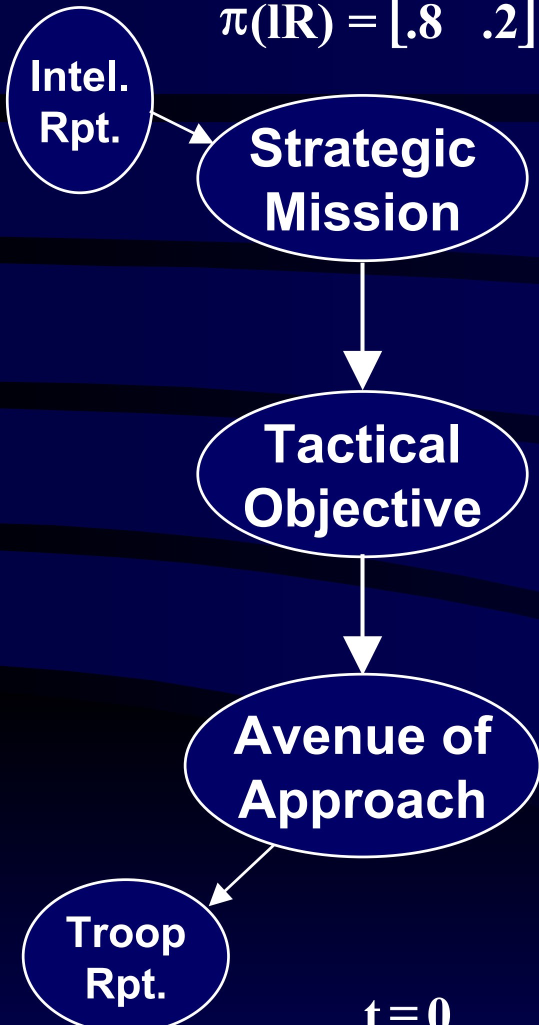
$t = 1$

$\lambda(AA) = \lambda(TR)$

Sample Chain - 2nd Propagation

$t = 0$

$\pi(\mathrm{IR}) = [.8 \quad .2]$



$t = 0$

$\lambda(\mathrm{TR}) = [.5 \quad 1 \quad .6]^T$

	$\pi(\mathrm{SM})$	$\mathrm{Bel}(\mathrm{SM})$	$\lambda(\mathrm{SM})$
Paris	0.8	0.8	1.0
Med.	0.2	0.2	1.0

$t = 2$

$\pi(\mathrm{TO}) = \pi(\mathrm{SM})\, M_{\mathrm{TO|SM}}$

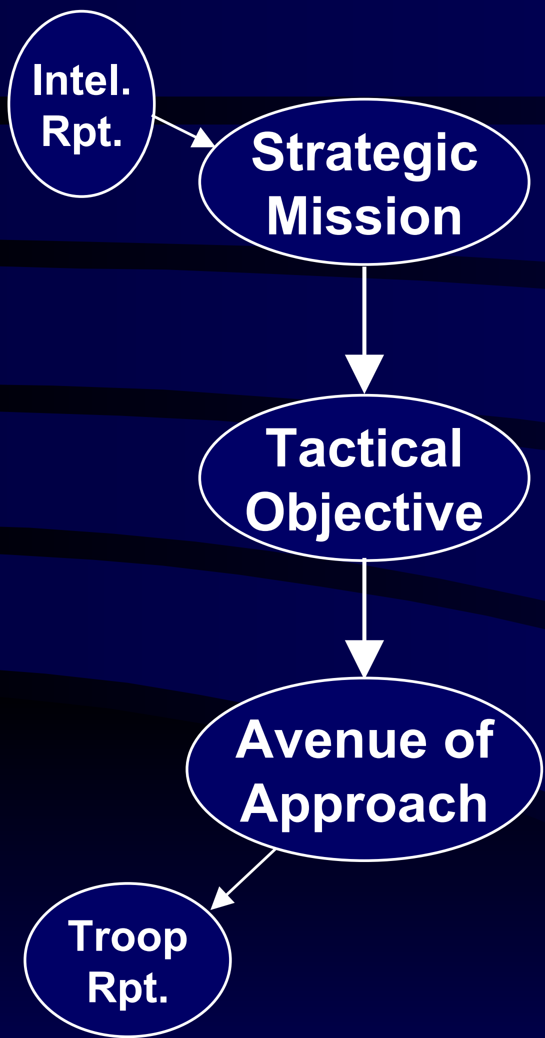
	$\pi(\mathrm{TO})$	$\mathrm{Bel}(\mathrm{TO})$	$\lambda(\mathrm{TO})$
Chalons	0.66	0.66	0.71
Dijon	0.34	0.34	0.71

$t = 2$

$\lambda(\mathrm{TO}) = M_{\mathrm{AA|TO}}\, \lambda(\mathrm{SM})$

	$\pi(\mathrm{AA})$	$\mathrm{Bel}(\mathrm{AA})$	$\lambda(\mathrm{AA})$
North	0.39	0.3	0.5
Central	0.35	0.5	1.0
South	0.24	0.2	0.6

Sample Chain - 3rd Propagation



	π(SM)	Bel(SM)	λ(SM)
Paris	0.8	0.8	0.71
Med.	0.2	0.2	0.71

t = 3

$\lambda(SM) = M_{TO|SM}\lambda(TO)$

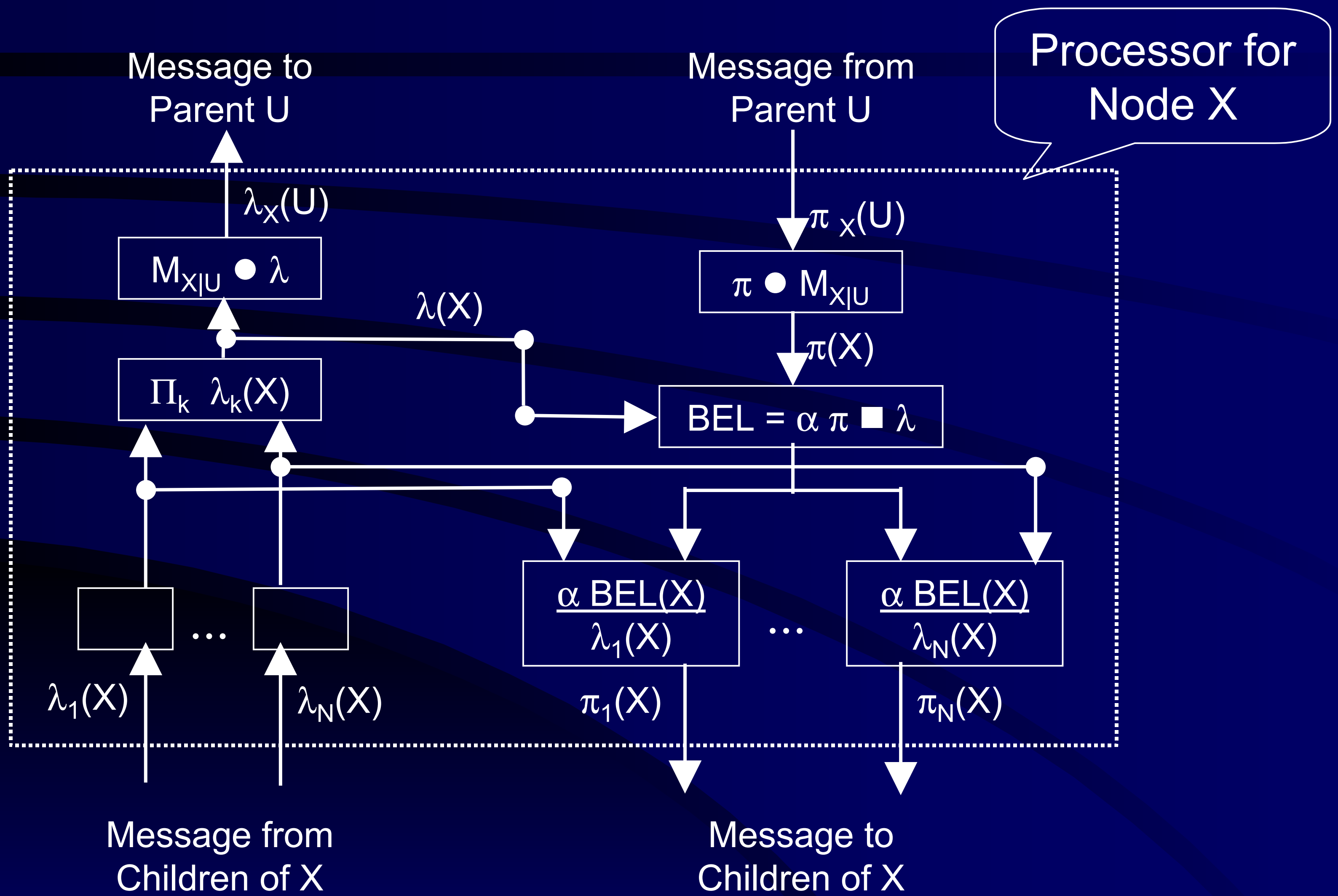
	π(TO)	Bel(TO)	λ(TO)
Chalons	0.66	0.66	0.71
Dijon	0.34	0.34	0.71

t = 3

$\pi(AA) = \pi(TO)\, M_{AA|TO}$

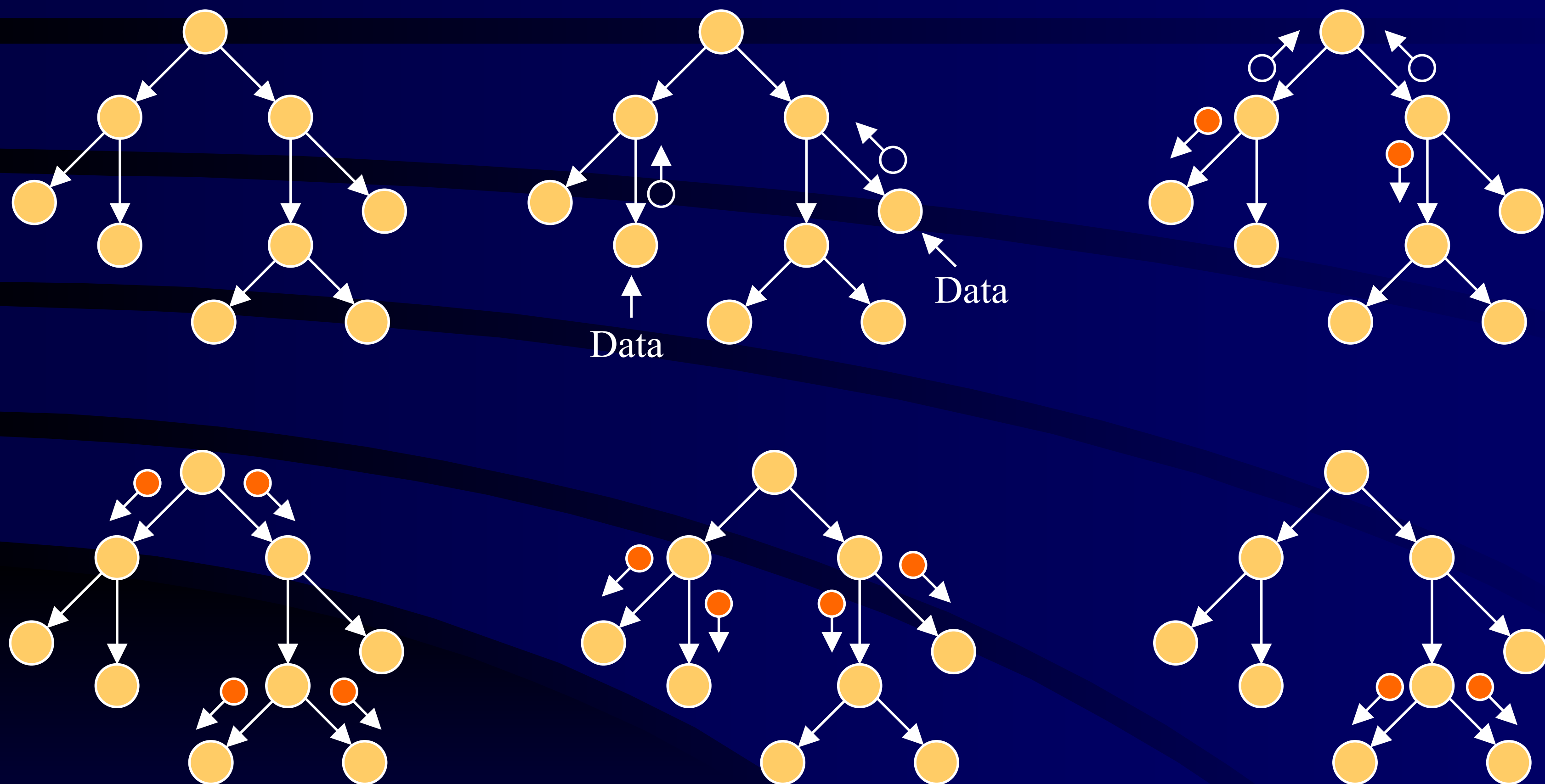
	π(AA)	Bel(AA)	λ(AA)
North	0.36	0.25	0.5
Central	0.37	0.52	1.0
South	0.27	0.23	0.6

Internal Structure of a Single Node Processor



Propagation Example

"The impact of each new piece of evidence is viewed as a perturbation that propagates through the network via message-passing between neighboring variables . . ." (Pearl, 1988, p 143`



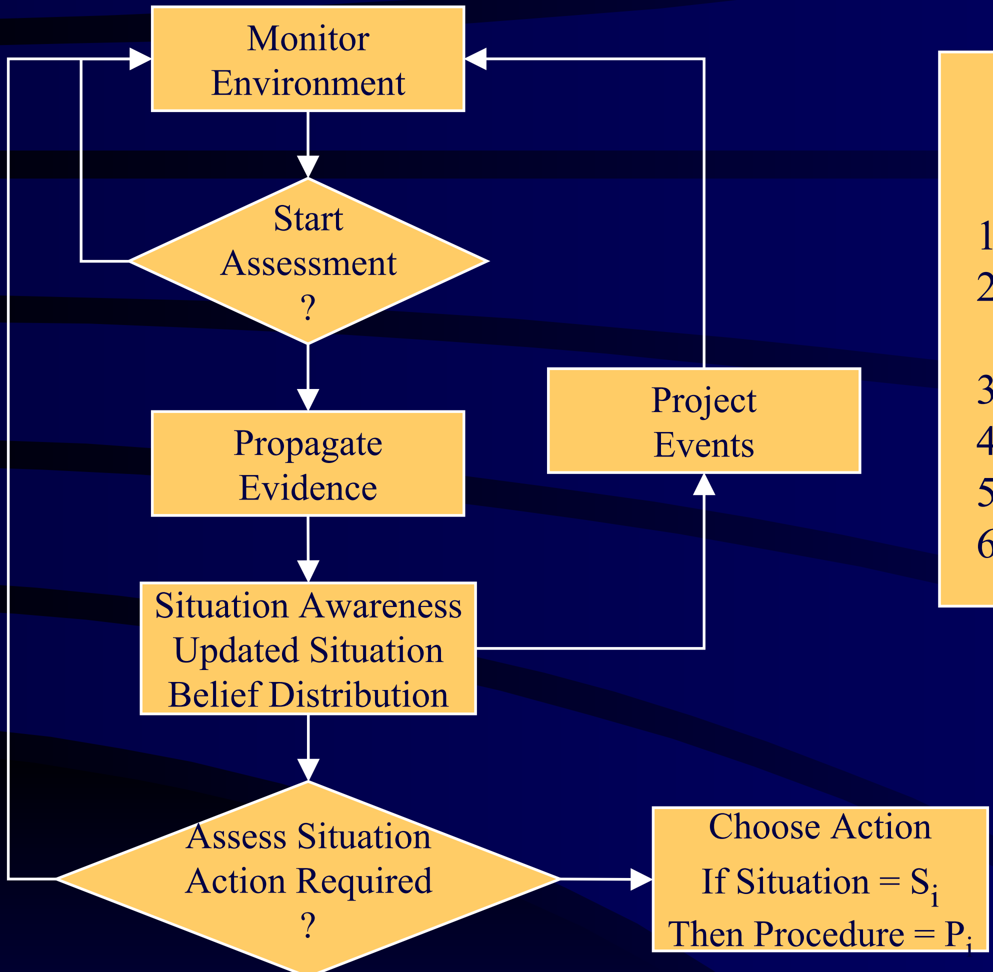
- **The example above requires five time periods to reach equilibrium after the introduction of data (Pearl, 1988, p 174)**

Categorization of Other Algorithms

- Exact algorithms
  - on original directed graph (only singly connected, e.g., Pearl)
  - on related undirected graph
    - Lauritzen & Spiegelhalter
    - Jensen
    - Symbolic Probabilistic Inference
  - on a different but related directed graph
  - using conditioning
  - using node reductions

- Simulation algorithms
  - Backward sampling
    - Stochastic simulation
  - Forward sampling
    - Logic sampling
    - Likelihood weighting
    - (With and without importance sampling)
    - (With and without Markov blanket scoring)

Decision Making in Nuclear Power Plant Operations

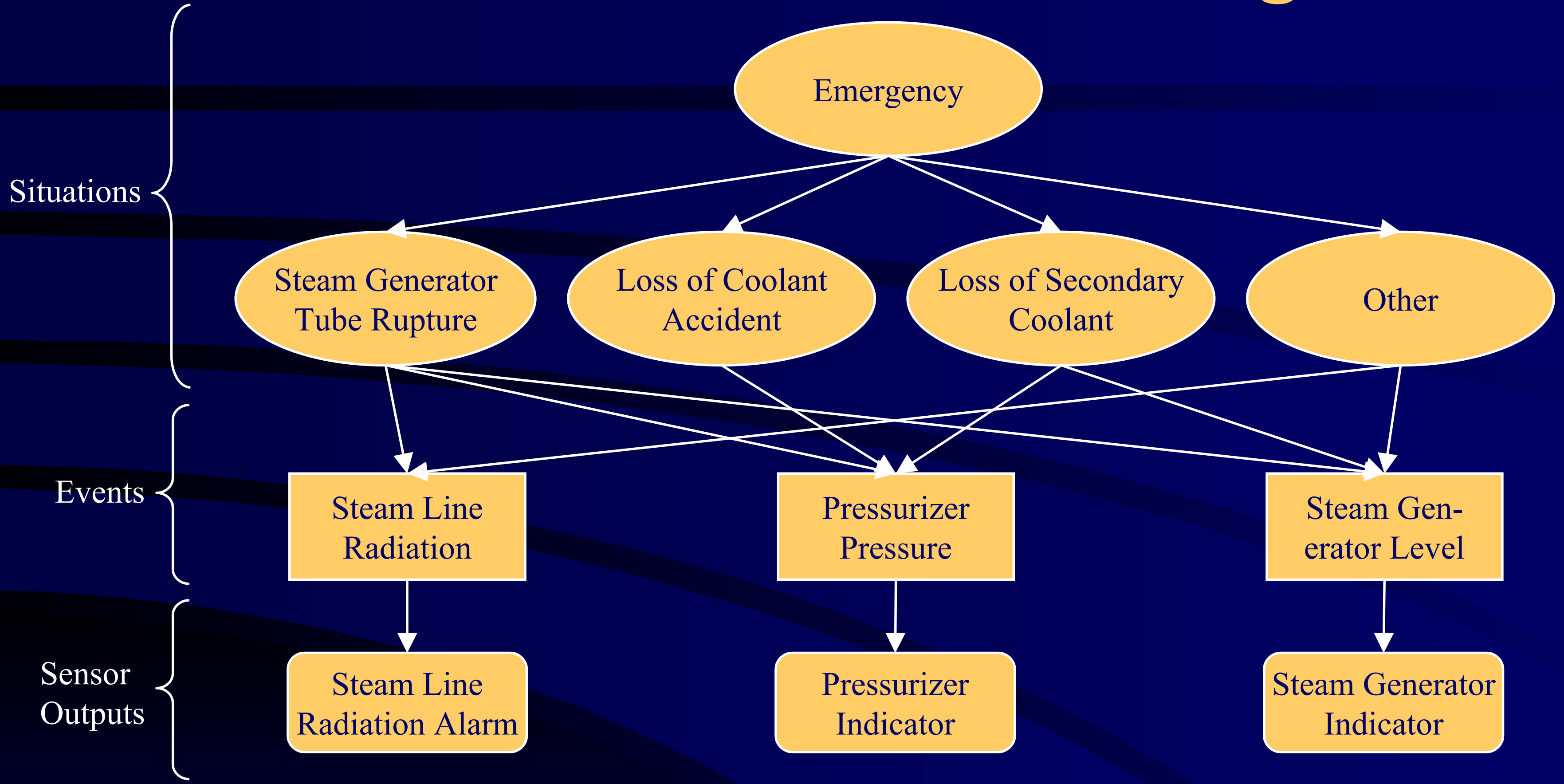


Situation Assessment (SA) Decision Making

1) Monitor the environment
2) Determine the need for situation assessment
3) Propagate event cues
4) Project Events
5) Assess Situation
6) Make Decision

- **"Decision making in nuclear power plant operations is characterized by:**
  - **Time pressure**
  - **Dynamically evolving scenarios**
  - **High expertise levels on the part of the operators**

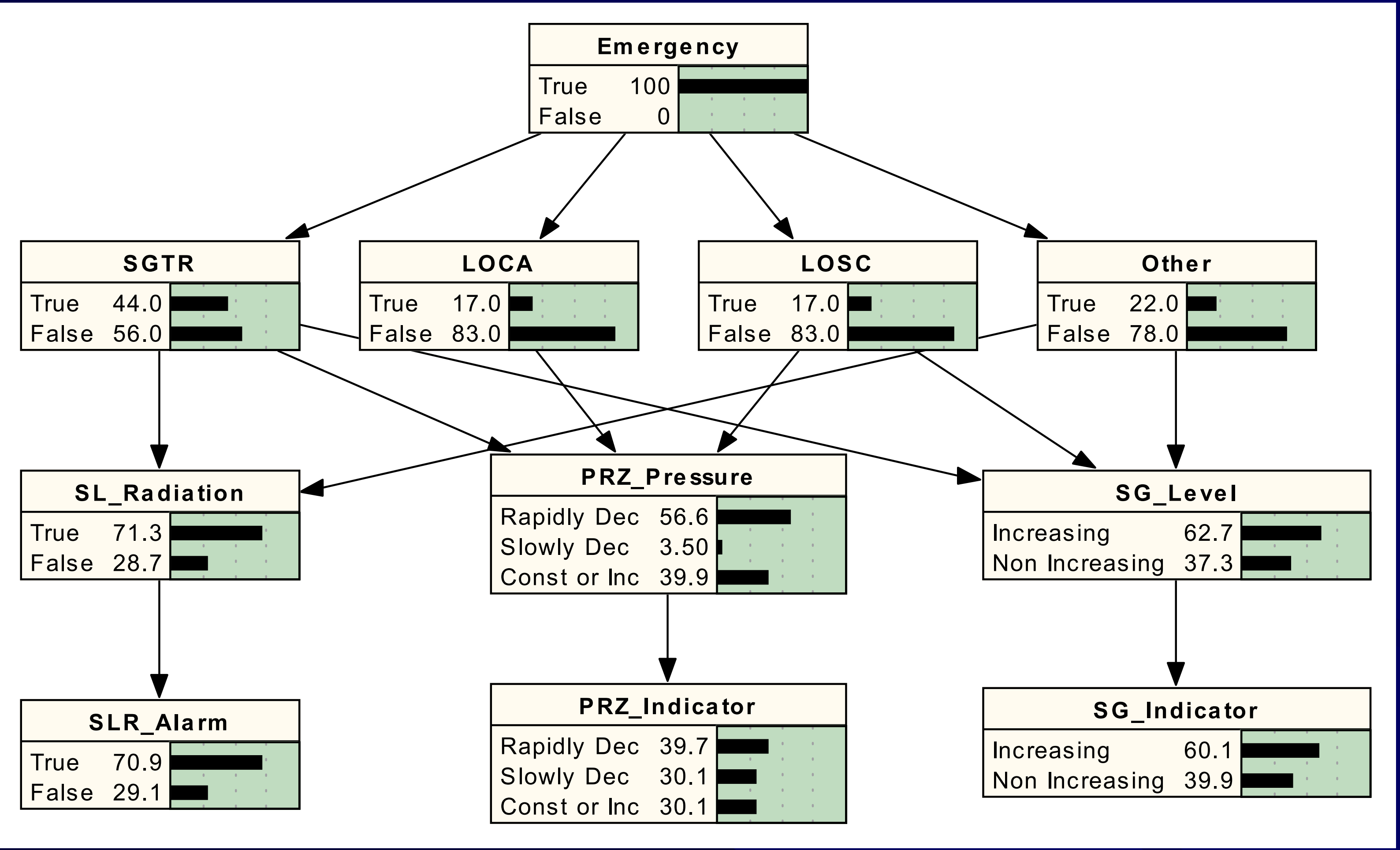
Model of Situation Assessment and Human Decision Making



- **The Bayesian net situation assessment model provides:**
  - **Knowledge of the structural relationship among situations, events, and event cues**
  - **Means of integrating the situations and events to form a holistic view of their meaning**
  - **Mechanism for projecting future events**

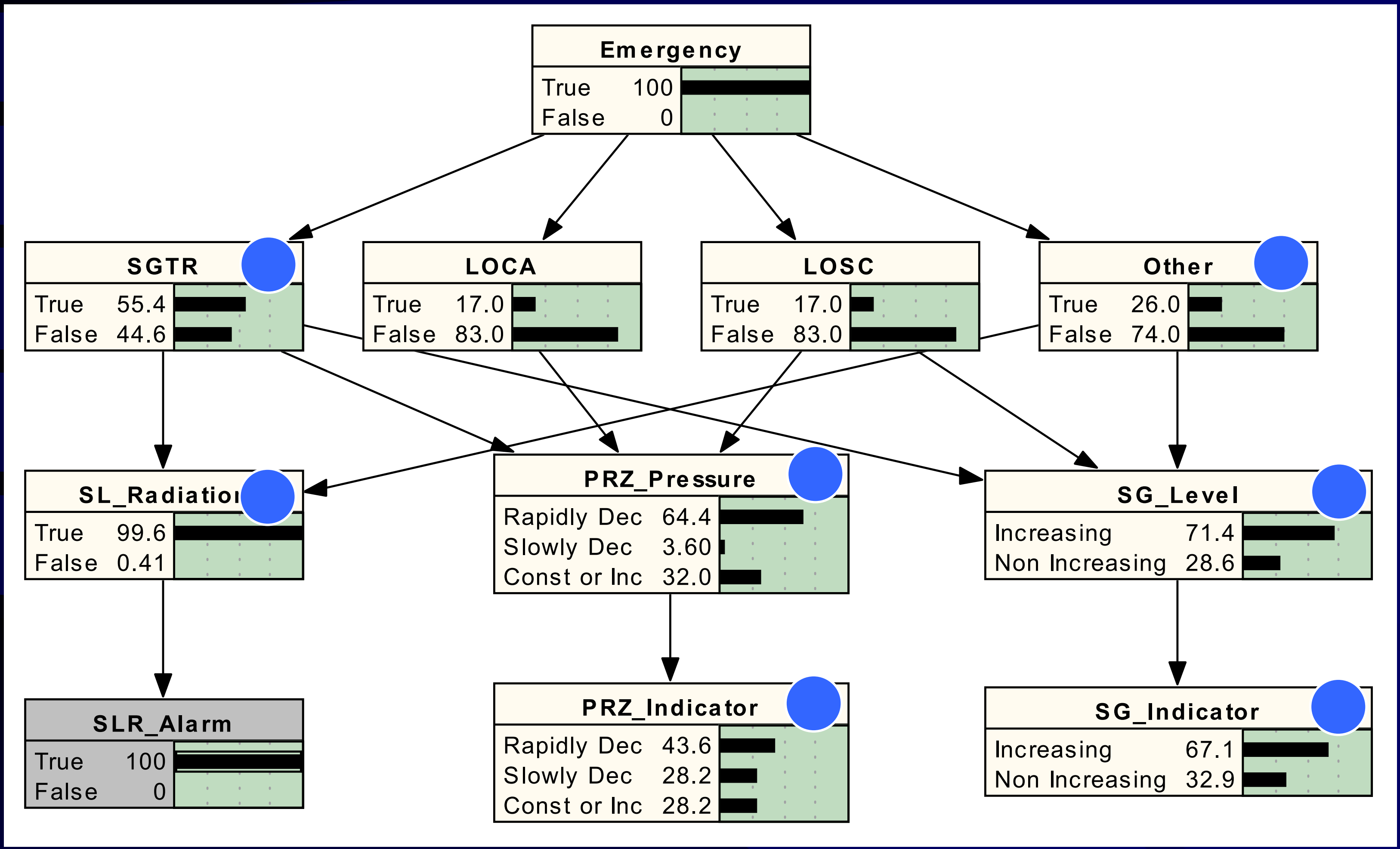
Situation Assessment Bayesian Net

Initial Conditions Given Emergency

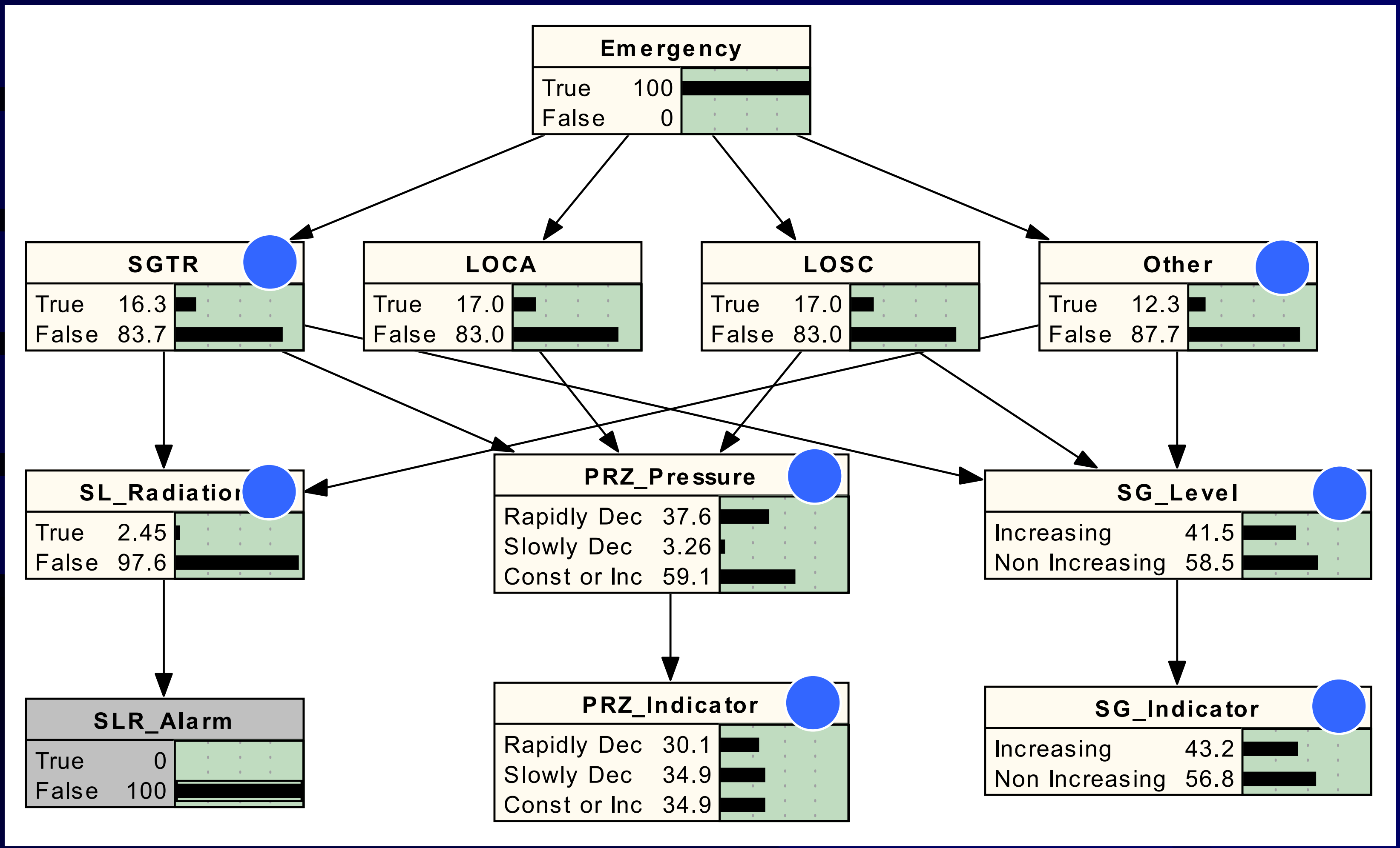


Situation Assessment Bayesian Net

Steam Line Radiation Alarm Goes High



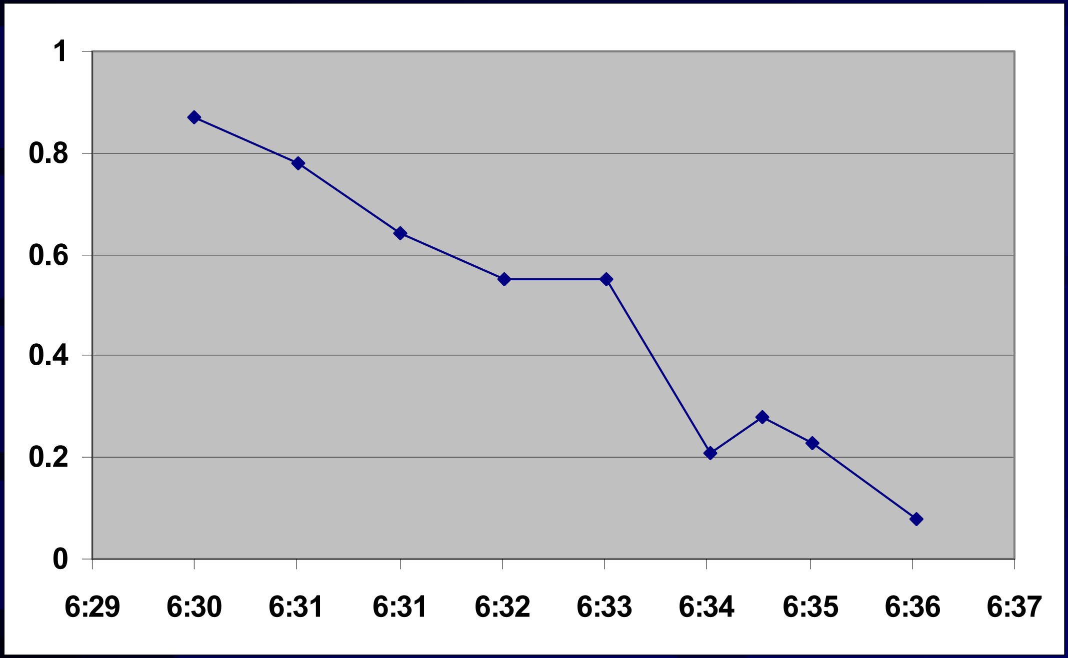
Situation Assessment Bayesian Net
Steam Line Radiation Alarm Goes Low



Simulation of SGTR Scenario
Event Timeline

Time	Event Cues	Actions
6:30:00		Steam generator tube rupture occurs
6:30:14	Radiation alarm	Operator observes that the radioactivity alarm for “A” steam line is on
6:30:21	Low pressure alarm	
6:30:34	Pressurizer level and pressure are decreasing rapidly	Charging FCV full open
6:30:44	Pressurizer pressure and level are still decreasing	Letdown isolation
6:30:54	Decrease in over-temperature-delta temperature limit	10% decrease in turbine load
6:32:34	Decreasing pressurizer pressure and level cannot be stopped from decreasing . . . Emergency	Manual trip
6:32:41		Automatic SI actuated
6:32:44	Reactor is tripped	EP-0 Procedure starts
6:33:44	Very low pressure of FW is present	FW is isolated
6:37:04	Pressurizer pressure less than 2350 psig	PORVs are closed
6:37:24	Radiation alarm, pressure decrease and SG level increase in loop “A”	SGTR is identified and isolated

Simulation of SGTR Scenario
Convergence of Situation Disparity



- Situation Disparity is defined as follows:
  - SD(t) = | Bel (S(t)) - Bel(S'(t)) |
  - S represents the actual situation
  - S' represents the perceived situation

Overview

- Day 1
  - Motivating Examples
  - Basic Constructs and Operations
- Day 2
  - Propagation Algorithms
  - Example Application
- Day 3
  - Learning
  - Continuous Variables
  - Software

Introduction to Bayesian Networks

A Tutorial for the 66th MORS Symposium

23 - 25 June 1998
Naval Postgraduate School
Monterey, California

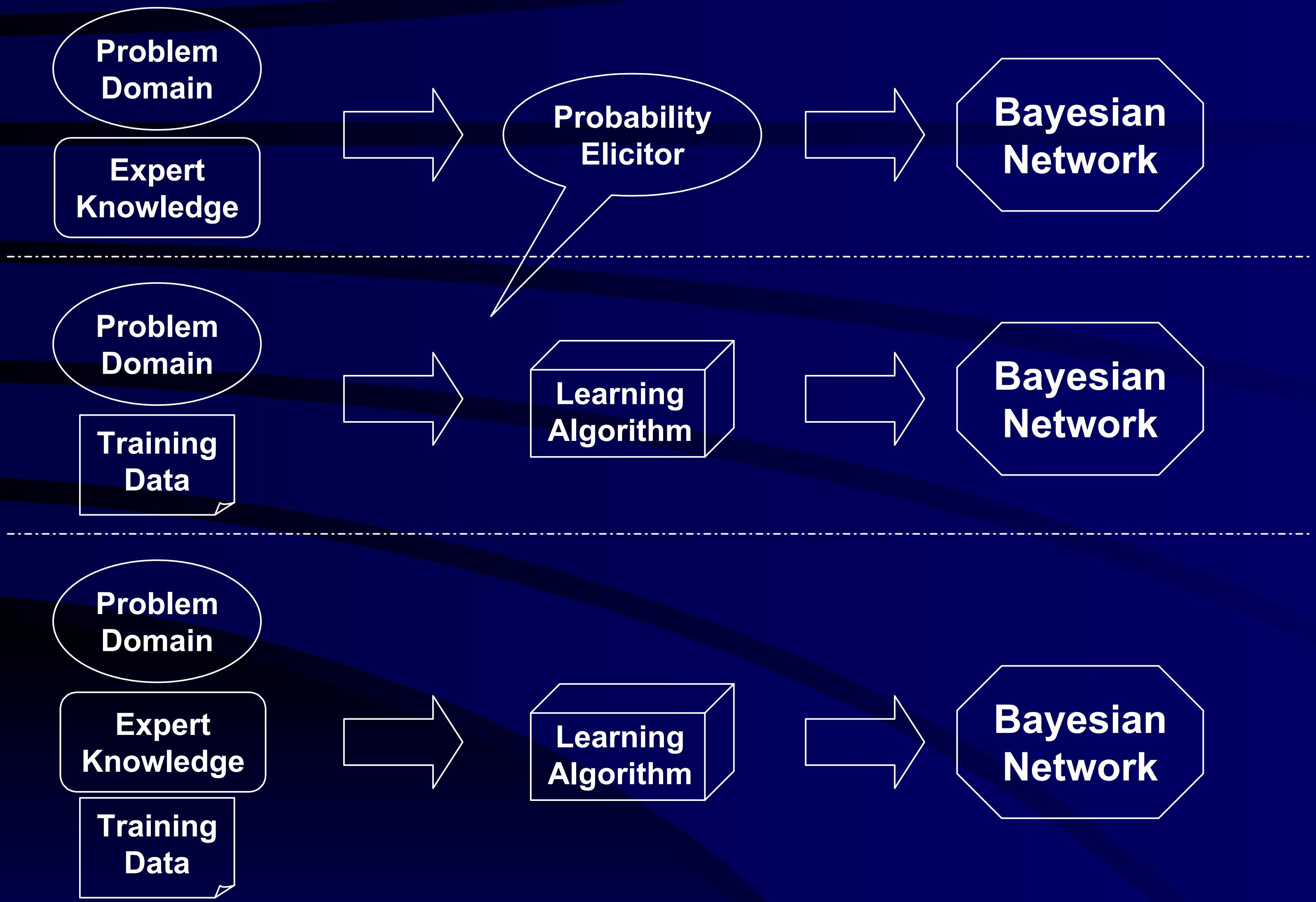
Dennis M. Buede, dbuede@gmu.edu
Joseph A. Tatman, jatatman@aol.com
Terry A. Bresnick, bresnick@ix.netcom.com

http://www.gmu.edu -
Depts (Info Tech & Eng) - Sys. Eng. - Buede

Overview

- **Day 1**
  - **Motivating Examples**
  - **Basic Constructs and Operations**
- **Day 2**
  - **Propagation Algorithms**
  - **Example Application**
- **Day 3**
  - **Learning**
  - **Continuous Variables**
  - **Software**

Building BN Structures



Learning Probabilities from Data

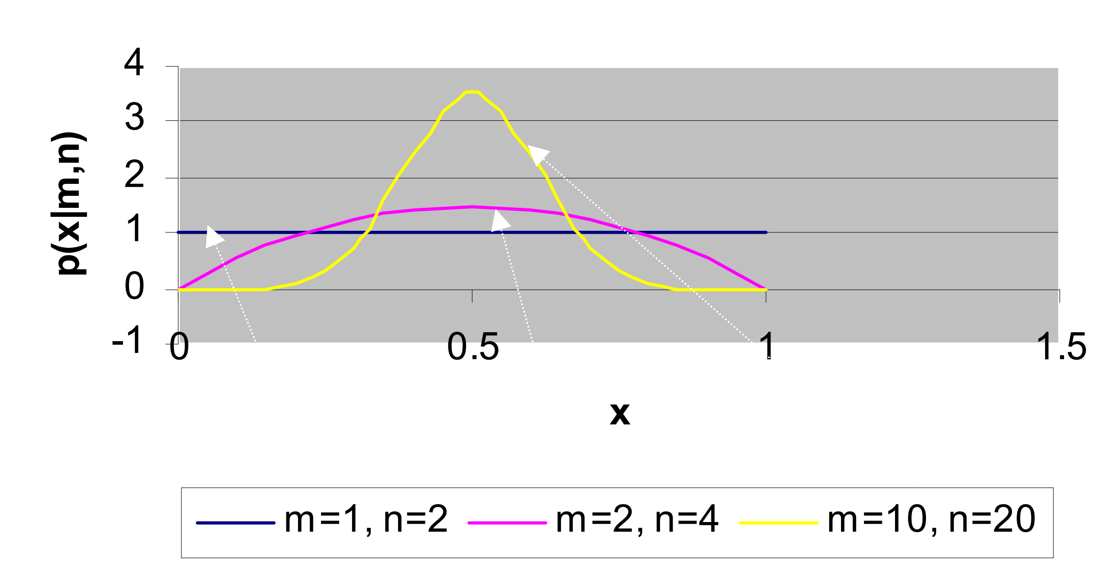
- Exploit conjugate distributions
  - Prior and posterior distributions in same family
  - Given a pre-defined functional form of the likelihood
- For probability distributions of a variable defined between 0 and 1, and associated with a discrete sample space for the likelihood
  - Beta distribution for 2 likelihood states (e.g., head on a coin toss)
  - Multivariate Dirichlet distribution for 3+ states in likelihood space

Beta Distribution

$$p_{Beta}(x \mid m,n) = \frac{\Gamma(n)}{\Gamma(m)\Gamma(n-m)} x^{m-1}(1-x)^{n-m-1}$$

$$mean = \frac{m}{n}$$

$$variance = \frac{m(1-m/n)}{n(n+1)}$$



Multivariate Dirichlet Distribution

$$p_{\text{Dirichlet}}(x \mid m_1, m_2, \ldots, m_N) = \frac{\Gamma(\sum_{i=1}^{N} m_i)}{\Gamma(m_1)\Gamma(m_2)\ldots\Gamma(m_N)} x^{m_1 - 1} x^{m_2 \text{-} 1} \ldots x^{m_N - 1}$$

$$\text{mean of the } i^{\text{th}} \text{ state} = \frac{m_i}{\sum_{i=1}^{N} m_i}$$

$$\text{variance of the } i^{\text{th}} \text{ state} = \frac{m_i (1 - m_i / \sum_{i=1}^{N} m_i)}{\sum_{i=1}^{N} m_i (\sum_{i=1}^{N} m_i + 1)}$$

Updating with Dirichlet

- Choose prior with $m_1 = m_2 = \ldots = m_N = 1$
  - Assumes no knowledge
  - Assumes all states equally likely: .33, .33, .33
  - Data changes posterior most quickly
  - Setting $m_i = 101$ for all i would slow effect of data down
- Compute number of records in database in each state
- For 3 state case:
  - 99 records in first state, 49 in second, 99 in third
  - Posterior values of m's: 100, 50, 100
  - Posterior probabilities equal means: .4, .2, .4
  - For m_i equal 101, posterior probabilities would be: .36, .27, .36

Learning BN Structure from Data

- Entropy Methods
  - Earliest method
  - Formulated for trees and polytrees
- Conditional Independence (CI)
  - Define conditional independencies for each node (Markov boundaries)
  - Infer dependencies within Markov boundary

- Score Metrics
  - Most implemented method
  - Define a quality metric to maximize
  - Use greedy search to determine the next best arc to add
  - Stop when metric does not increase by adding an arc
- Simulated Annealing & Genetic Algorithms
  - Advancements over greedy search for score metrics

Sample Score Metrics

- Bayesian score: p(network structure | database)
- Information criterion: log p(database | network structure and parameter set)
  - Favors complete networks
  - Commonly add a penalty term on the number of arcs
- Minimum description length: equivalent to the information criterion with a penalty function
  - Derived using coding theory

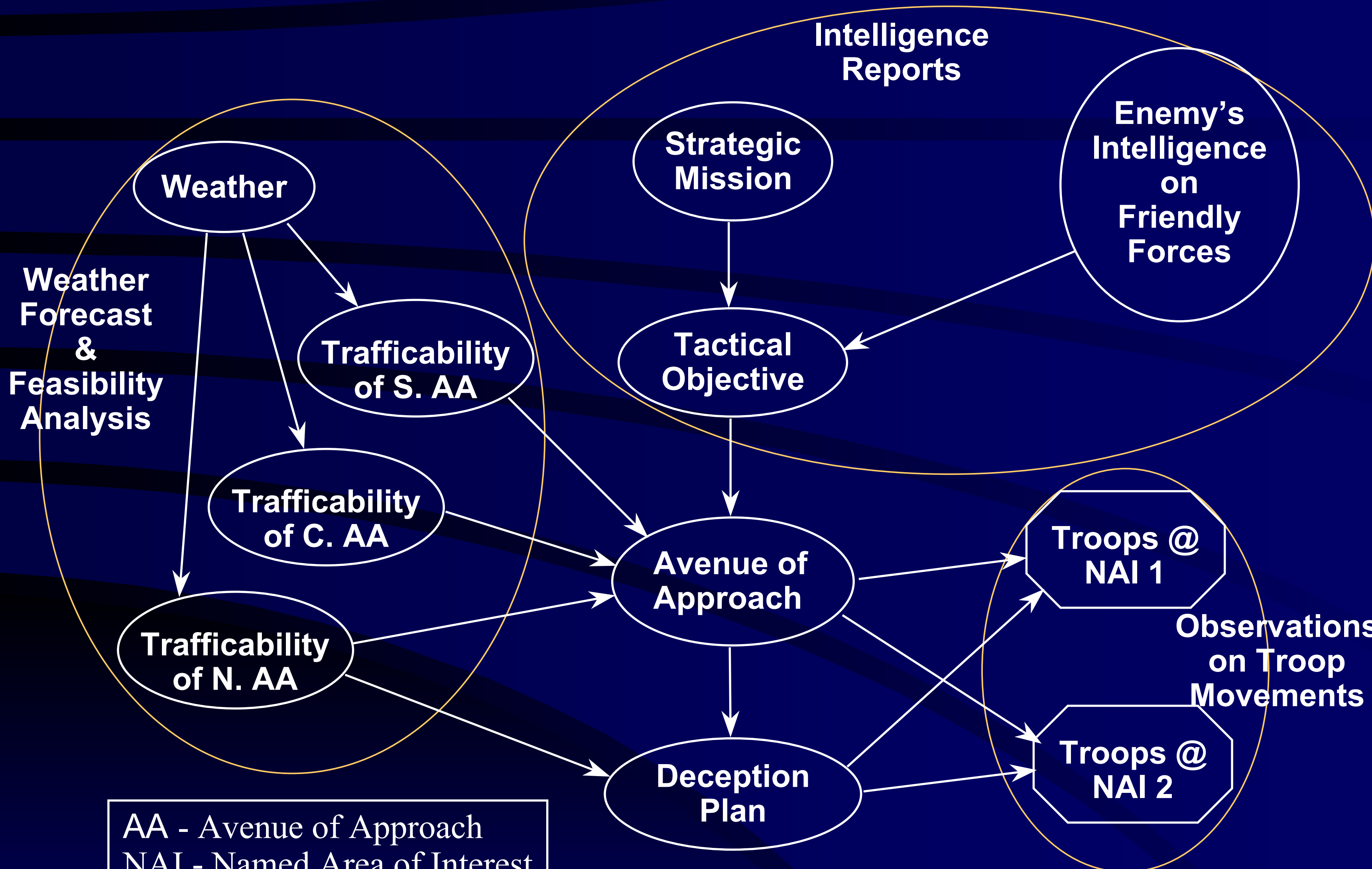
Features for Adding Knowledge to Learning Structure

- Define Total Order of Nodes
- Define Partial Order of Nodes by Pairs
- Define “Cause & Effect” Relations

Demonstration of Bayesian Network Power Constructor

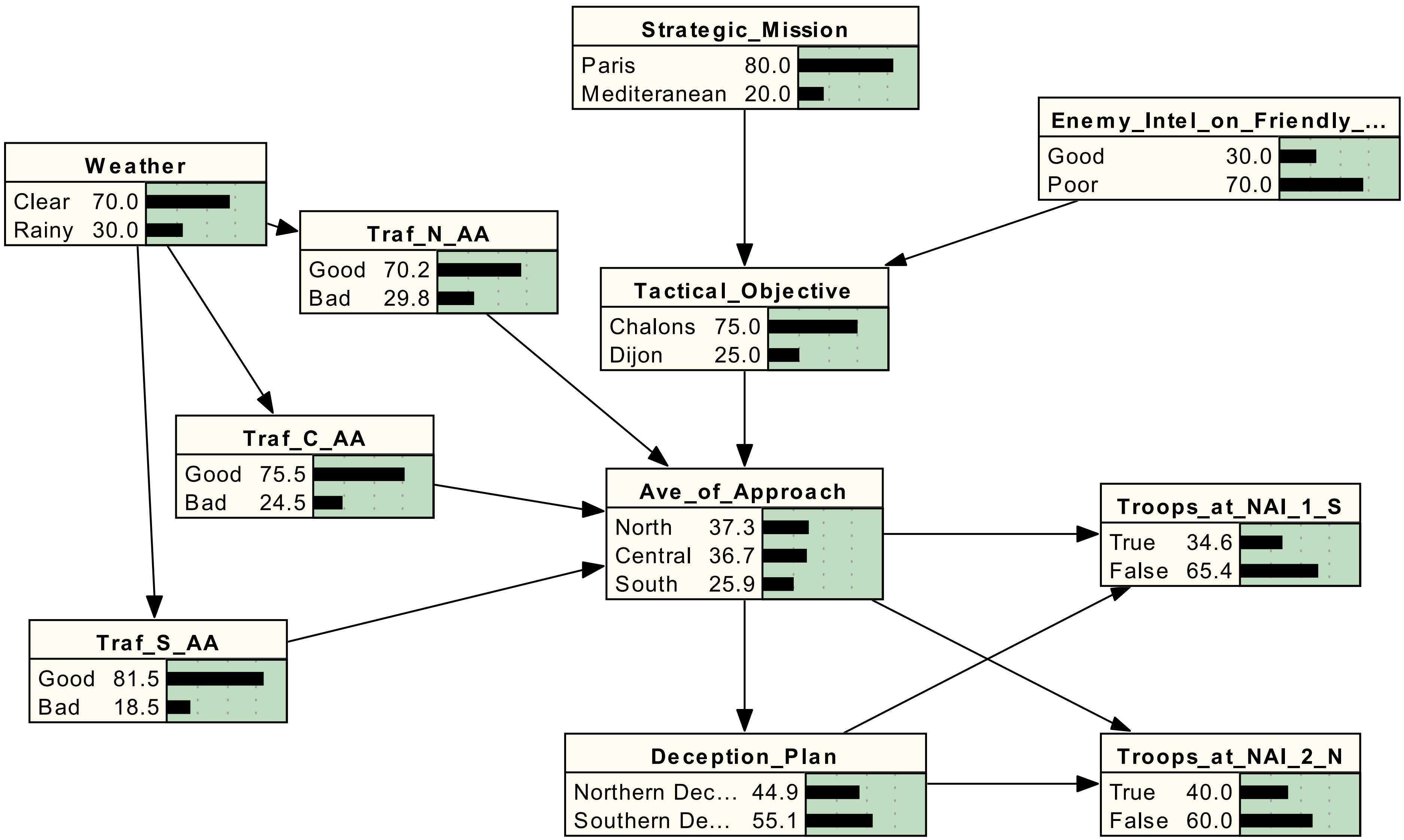
- Generate a random sample of cases using the original “true” network
  - 1000 cases
  - 10,000 cases
- Use sample cases to learn structure (arc locations and directions) with a CI algorithm in Bayesian Power Constructor
- Use same sample cases to learn probabilities for learned structure with priors set to uniform distributions
- Compare “learned” network to “true” network

Enemy Intent

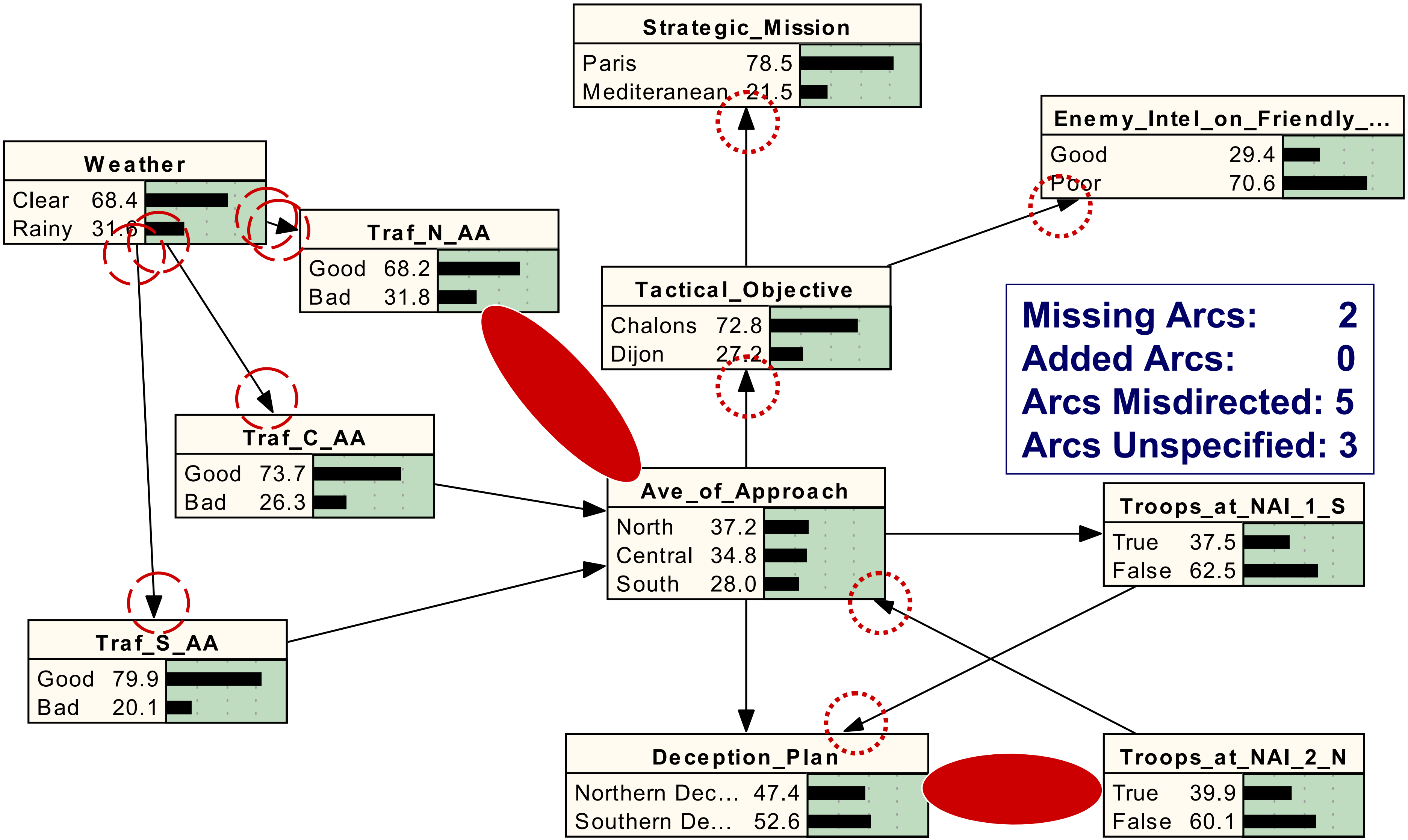


AA - Avenue of Approach
NAI - Named Area of Interest

Original Network

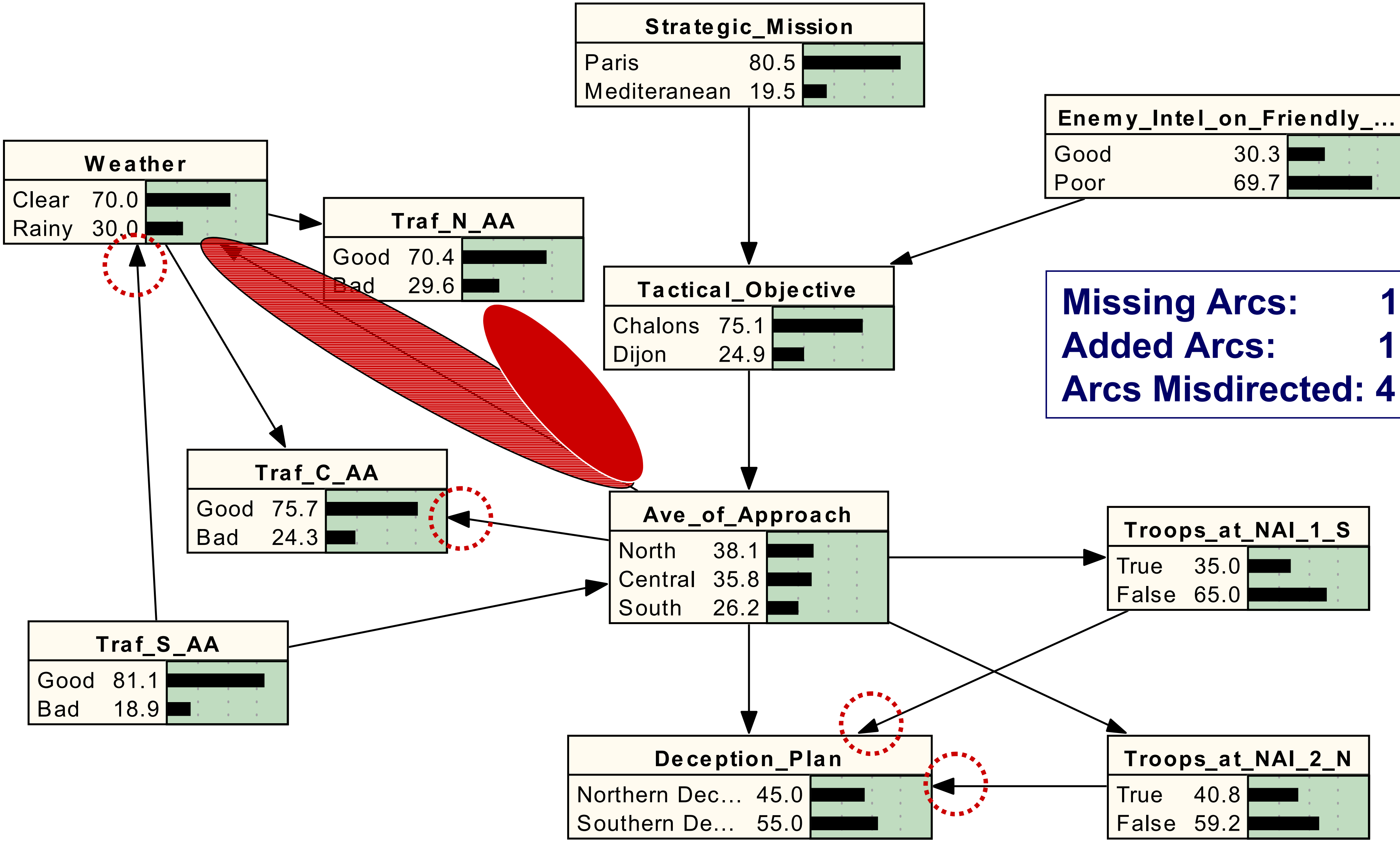


Learned Network with 1000 Cases



Missing Arcs: 2
Added Arcs: 0
Arcs Misdirected: 5
Arcs Unspecified: 3

Learned Network with 10,000 Cases



Missing Arcs: 1
Added Arcs: 1
Arcs Misdirected: 4

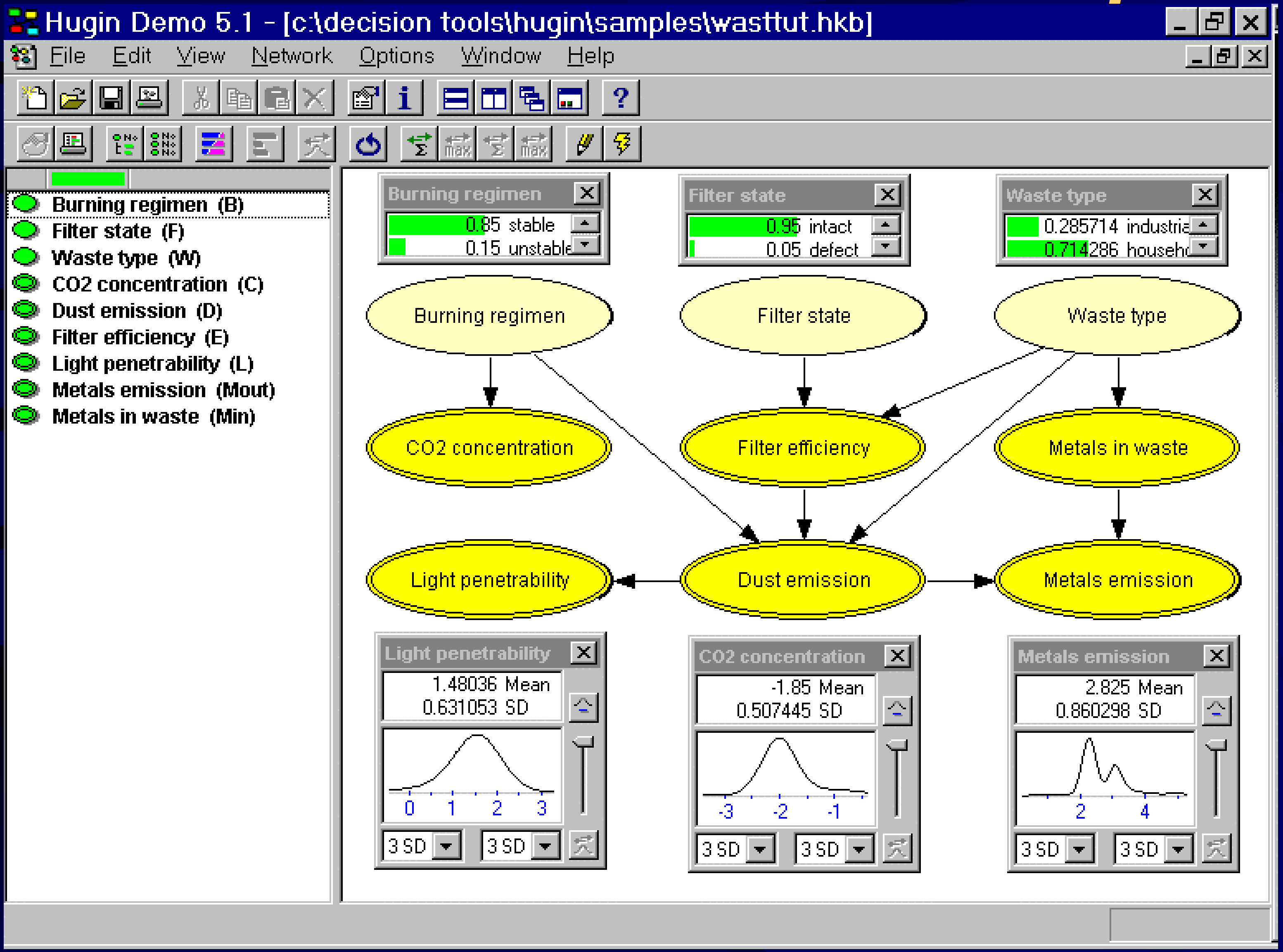
Comparison of Learned Networks with Truth

p(AoA)	Truth	1 K	10 K
Prior	.37, .37, .26	.37, .35, .28	.38, .36, .26
“Clear”	.41, .37, .22	.38, .36, .26	.41, .36, .23
“Rainy”	.30, .36, .34	.35, .32, .33	.30, .36, .34
“NAI 1 True”	.15, .13, .71	.17, .12, .71	.16, .12, .71
“Rain, NAI 1 True”	.10, .11, .79	.15, .10, .75	.11, .11, .78
“Rain, NAI 1 & 2 True”	.56, .02, .43	.59, .05, .36	.56, .03, .40

Summary of Comparison

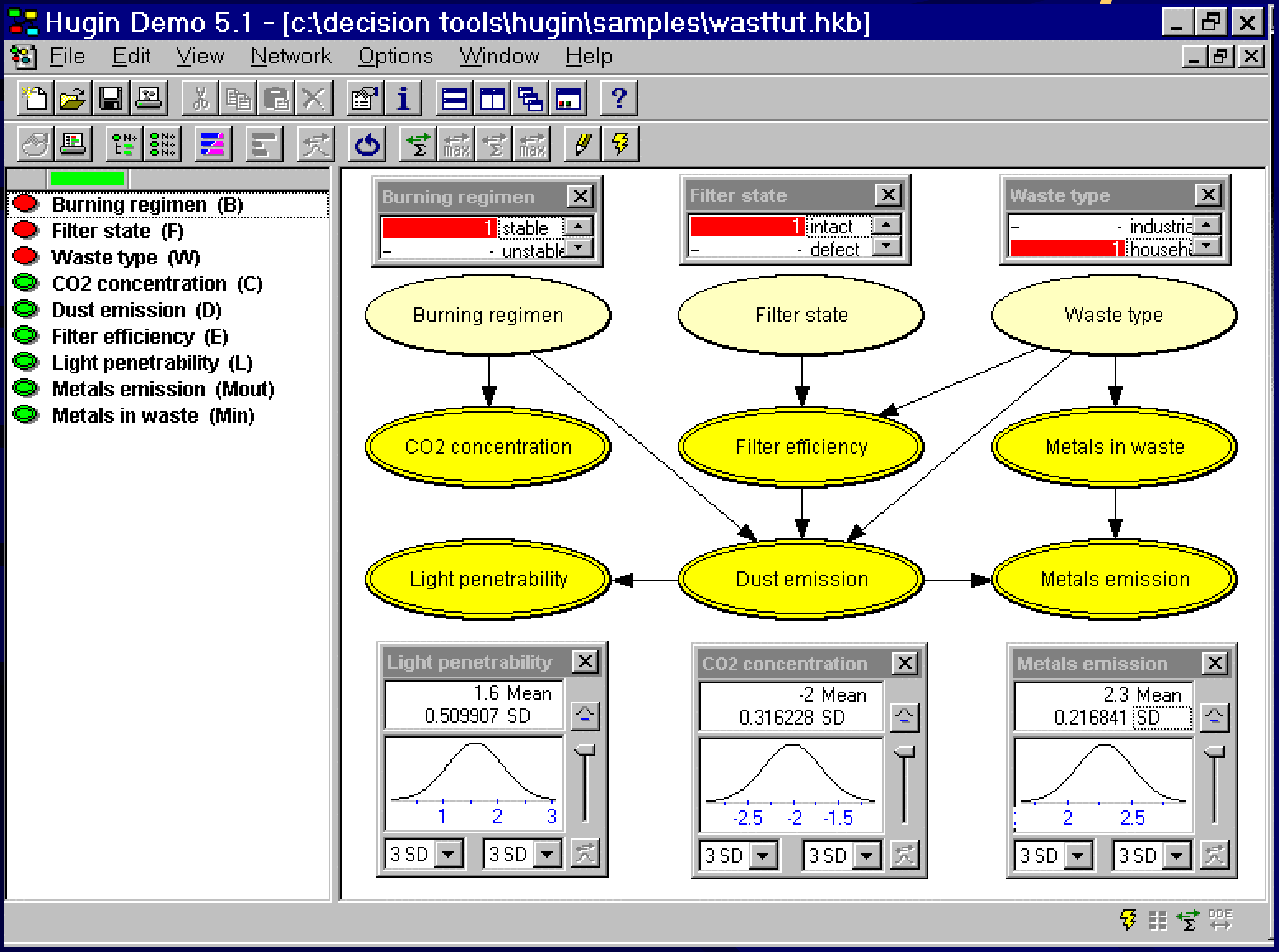
- Reasonable accuracy can be obtained with a relatively small sample
  - Prior probabilities (before data) look better than posterior probabilities (after data) for small samples
- More data improves results, but may not guarantee learning the same network
- Using partial order expertise can improve the structure of the learned network
- Comparison did not have any nodes with low probability outcomes
  - Learning algorithms requires 200-400 samples per outcome
  - In some cases, even 10,000 data points will not be enough

Continuous Variables Example



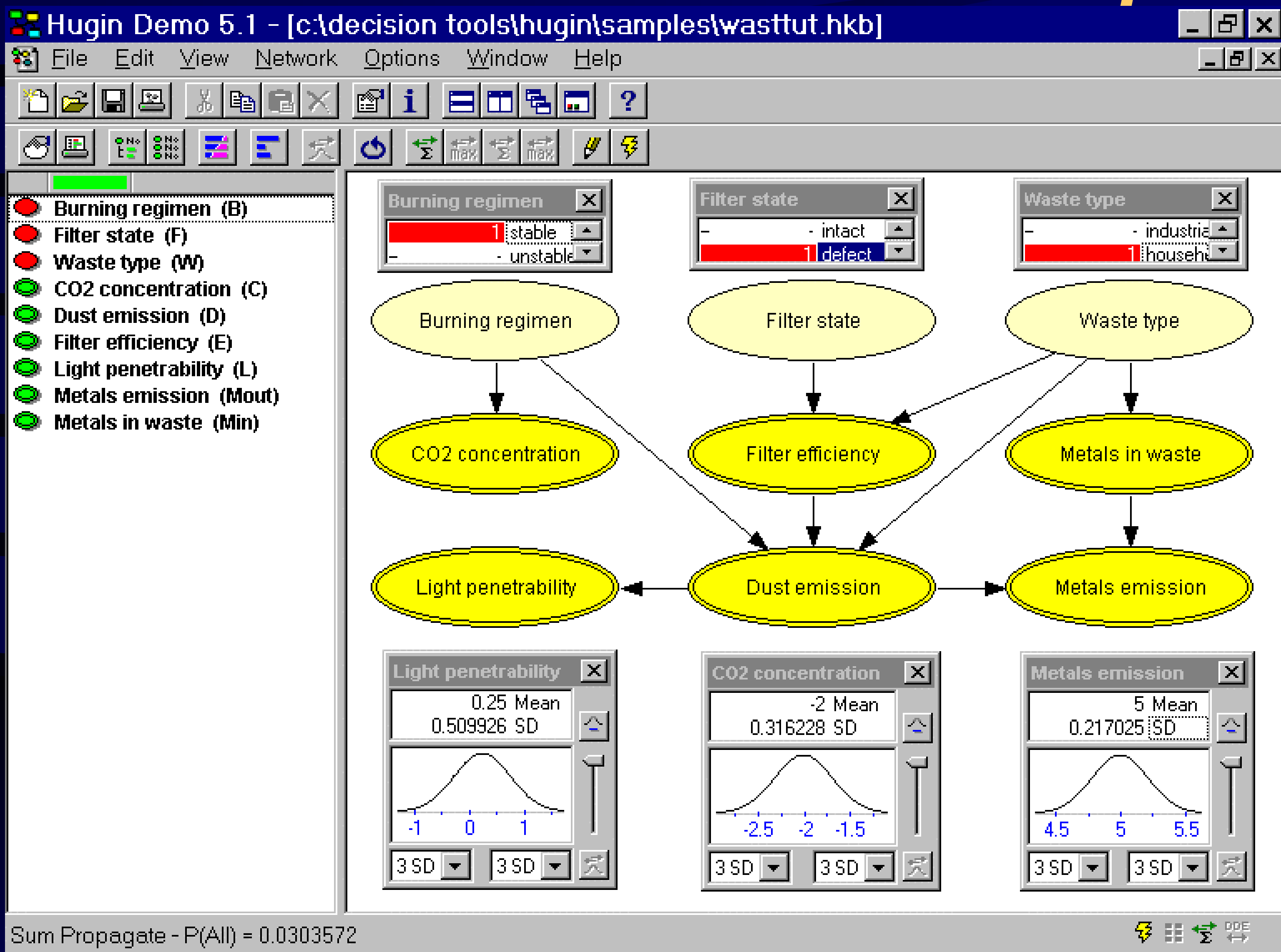
- Data from three sensors can be fused to gain information on relevant variables

Continuous Variables Example



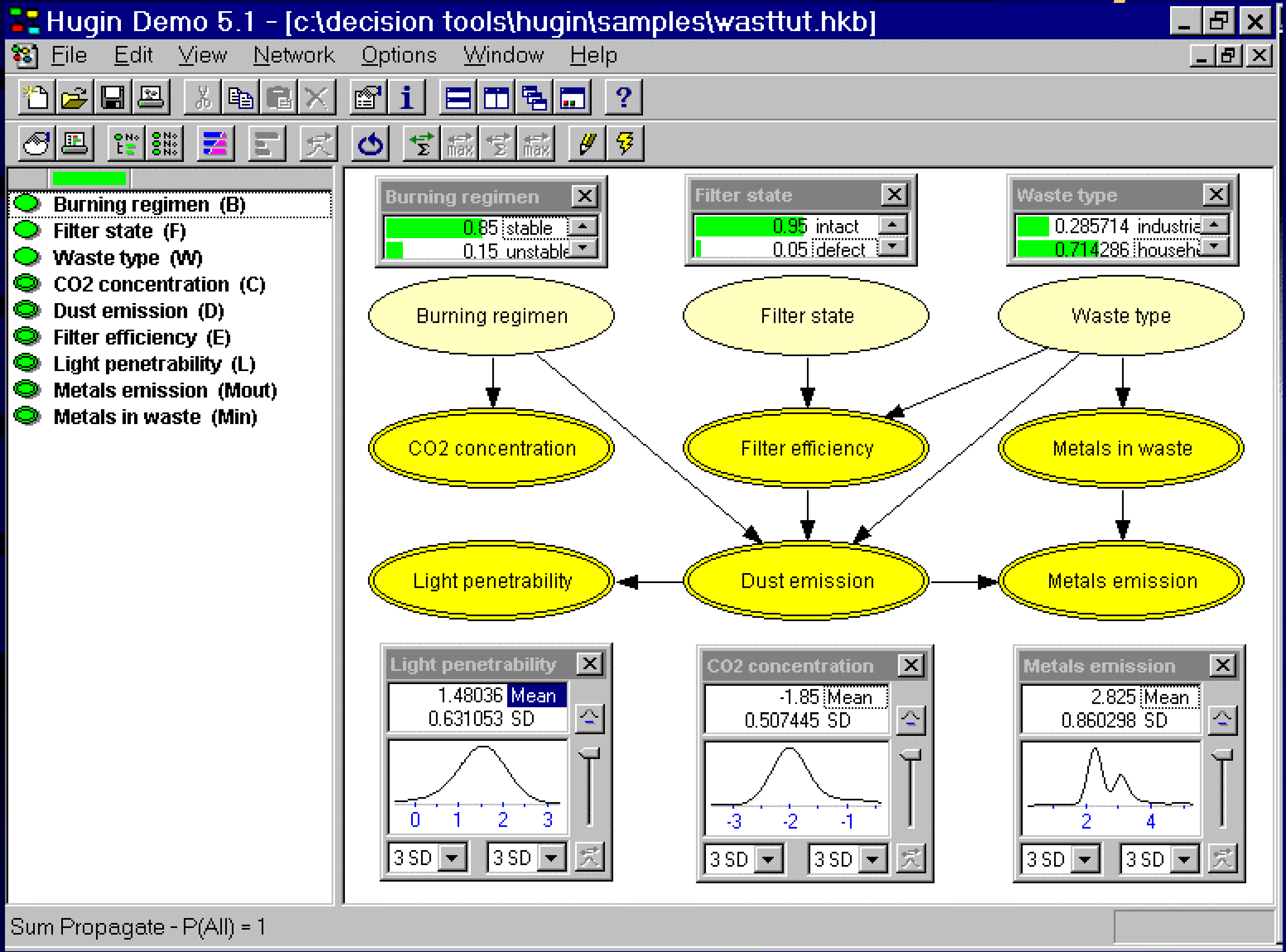
Entering values for the three discrete random variables shifts the sensor mean values

Continuous Variables Example



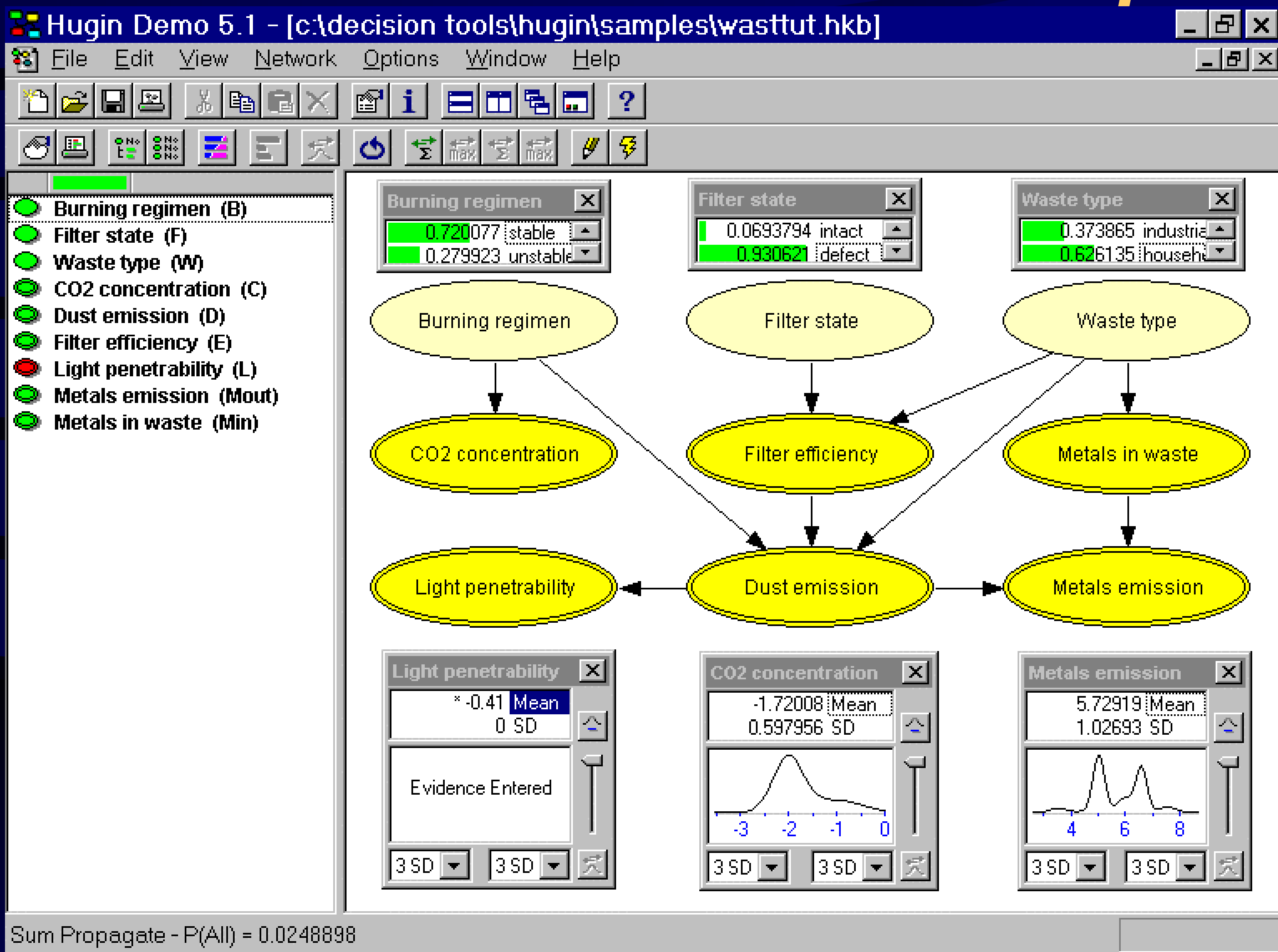
- A defective filter has a strong impact on the light penetrability and metal emissions sensors

Continuous Variables Example



- What can we learn about the three state variables given sensor outputs?

Continuous Variables Example



- A light penetrability reading that is 3 sigma low is a strong indicator of a defective filter

Software

- Many software packages available
  - See Russell Almond's Home Page
- Netica
  - www.norsys.com
  - Very easy to use
  - Implements learning of probabilities
  - Will soon implement learning of network structure
- Hugin
  - www.hugin.dk
  - Good user interface
  - Implements continuous variables

Basic References

- Pearl, J. (1988). Probabilistic Reasoning in Intelligent Systems. San Mateo, CA: Morgan Kauffman.
- Oliver, R.M. and Smith, J.Q. (eds.) (1990). Influence Diagrams, Belief Nets, and Decision Analysis, Chichester, Wiley.
- Neapolitan, R.E. (1990). Probabilistic Reasoning in Expert Systems, New York: Wiley.
- Schum, D.A. (1994). The Evidential Foundations of Probabilistic Reasoning, New York: Wiley.
- Jensen, F.V. (1996). An Introduction to Bayesian Networks, New York: Springer.

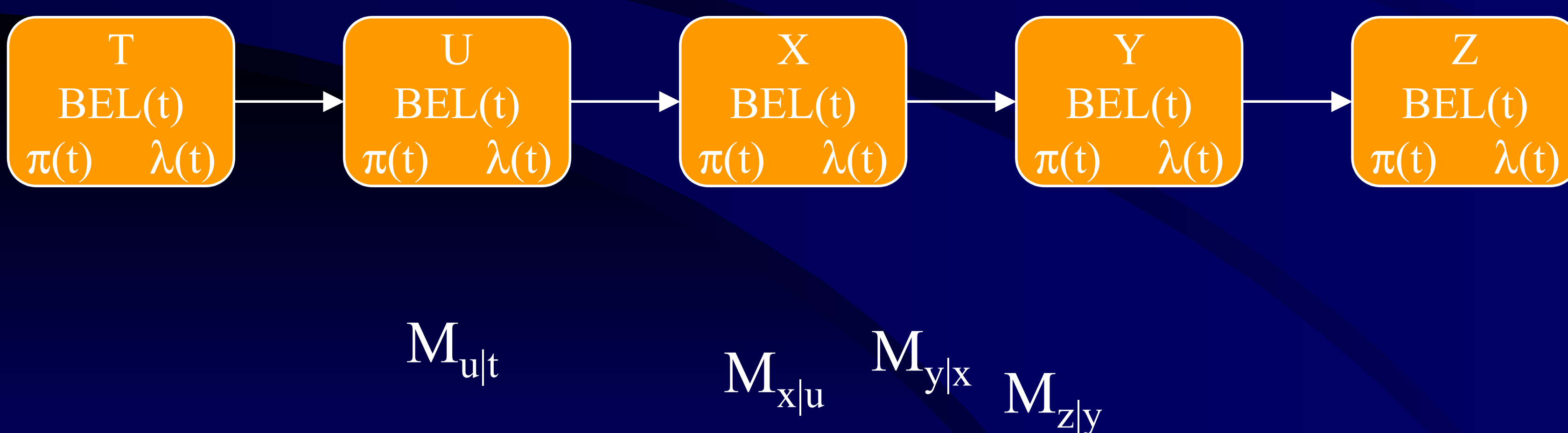
Algorithm References

- Chang, K.C. and Fung, R. (1995). Symbolic Probabilistic Inference with Both Discrete and Continuous Variables, IEEE SMC, 25(6), 910-916.
- Cooper, G.F. (1990) The computational complexity of probabilistic inference using Bayesian belief networks. Artificial Intelligence, 42, 393-405,
- Jensen, F.V, Lauritzen, S.L., and Olesen, K.G. (1990). Bayesian Updating in Causal Probabilistic Networks by Local Computations. Computational Statistics Quarterly, 269-282.
- Lauritzen, S.L. and Spiegelhalter, D.J. (1988). Local computations with probabilities on graphical structures and their application to expert systems. J. Royal Statistical Society B, 50(2), 157-224.
- Pearl, J. (1988). Probabilistic Reasoning in Intelligent Systems. San Mateo, CA: Morgan Kauffman.
- Shachter, R. (1988). Probabilistic Inference and Influence Diagrams. Operations Research, 36(July-August), 589-605.
- Suermondt, H.J. and Cooper, G.F. (1990). Probabilistic inference in multiply connected belief networks using loop cutsets. International Journal of Approximate Reasoning, 4, 283-306.

Backup

The Propagation Algorithm

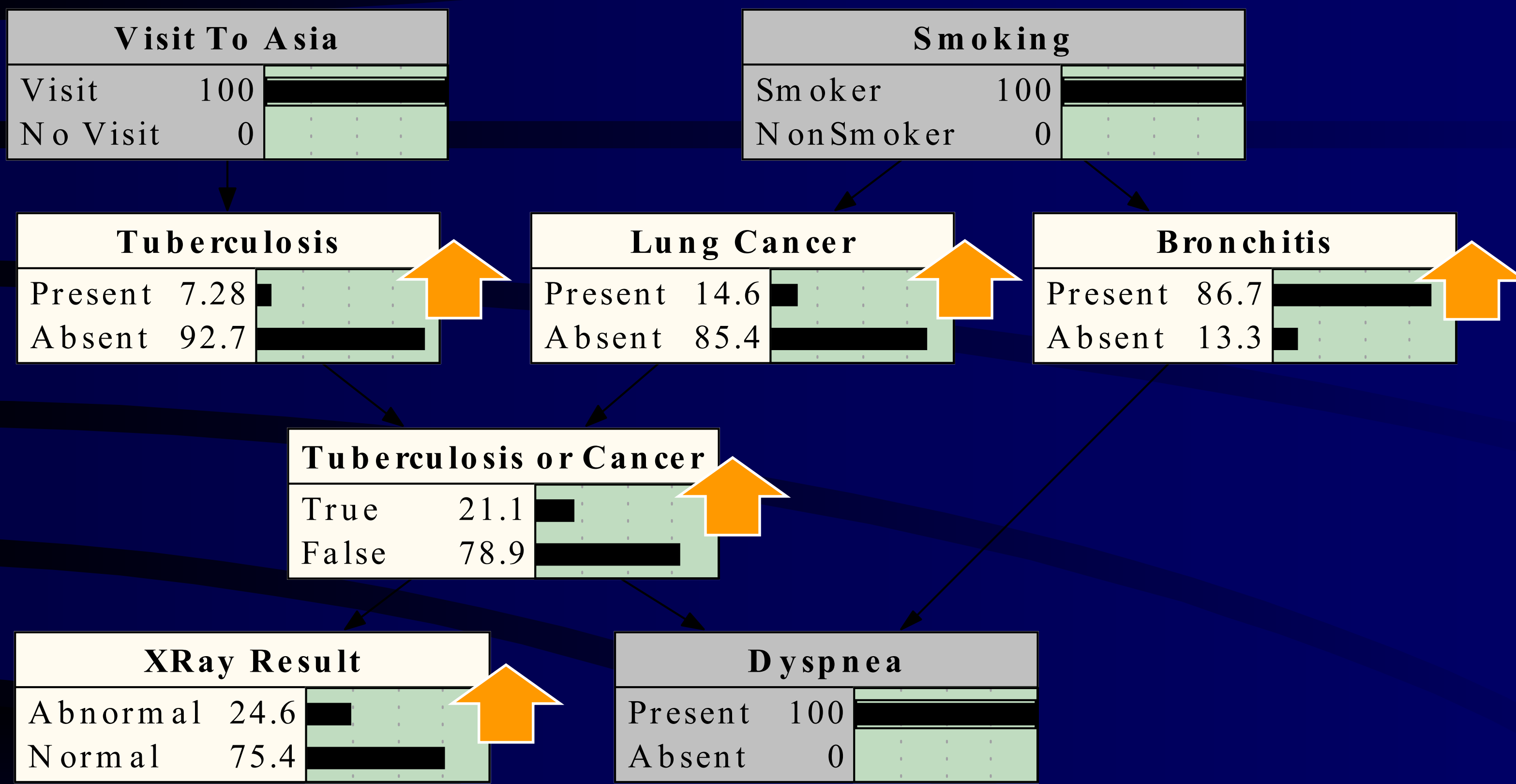
- As each piece of evidence is introduced, it generates:
  - A set of "π" messages that propagate through the network in the direction of the arcs
  - A set of "λ" messages that propagate through the network against the direction of the arcs
- As each node receives a "π" or "λ" message:
  - The node updates its own "π" or "λ" and sends it out onto the network
  - The node uses its updated "π" or "λ" to update its BEL function



Key Events in Development of Bayesian Nets

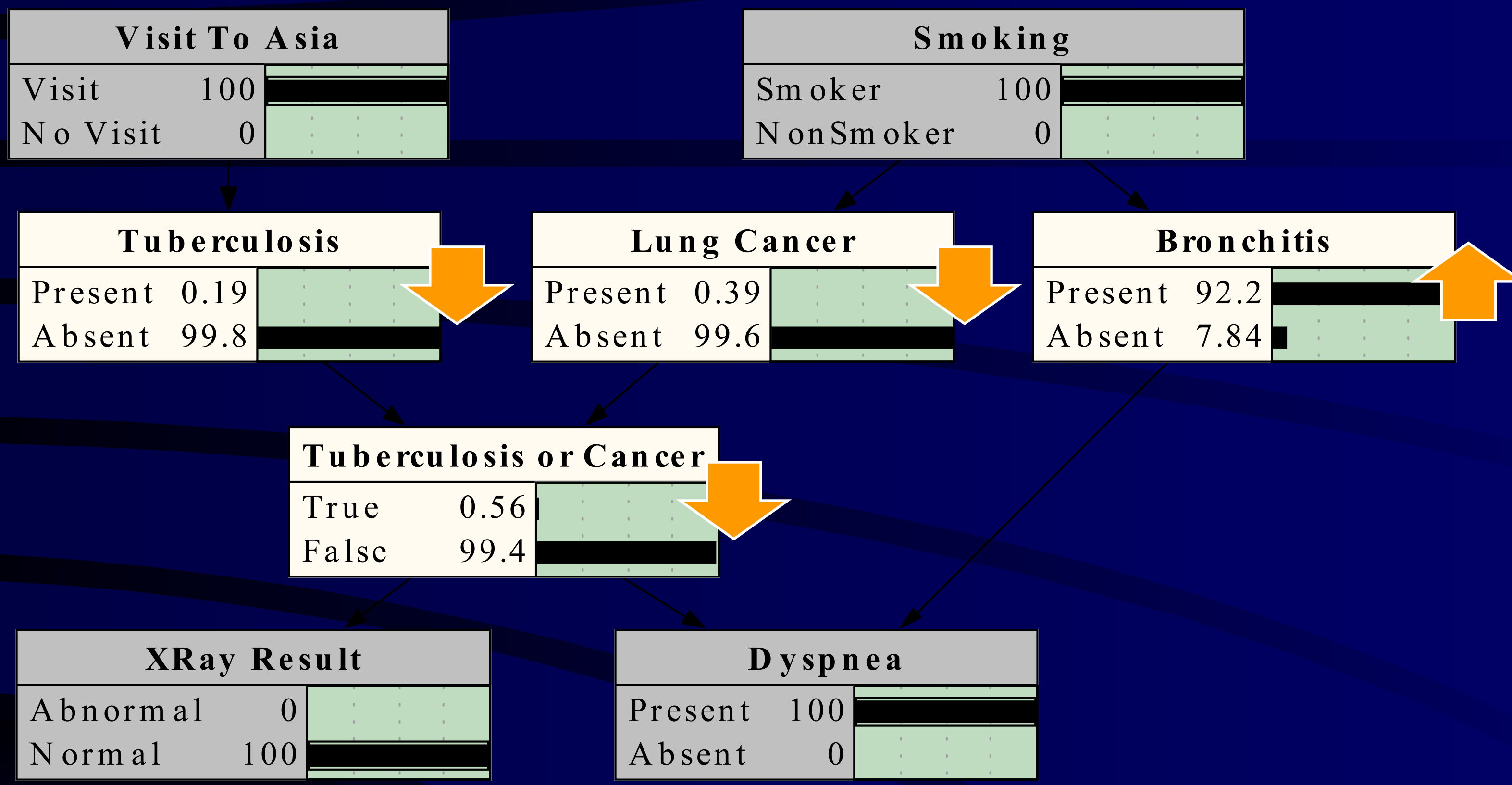
- 1763 Bayes Theorem presented by Rev Thomas Bayes (posthumously) in the Philosophical Transactions of the Royal Society of London
- 19xx Decision trees used to represent decision theory problems
- 19xx Decision analysis originates and uses decision trees to model real world decision problems for computer solution
- 1976 Influence diagrams presented in SRI technical report for DARPA as technique for improving efficiency of analyzing large decision trees
- 1980s Several software packages are developed in the academic environment for the direct solution of influence diagrams
- 1986? Holding of first Uncertainty in Artificial Intelligence Conference motivated by problems in handling uncertainty effectively in rule-based expert systems
- 1986 "Fusion, Propagation, and Structuring in Belief Networks" by Judea Pearl appears in the journal *Artificial Intelligence*
- 1986,1988 Seminal papers on solving decision problems and performing probabilistic inference with influence diagrams by Ross Shachter
- 1988 Seminal text on belief networks by Judea Pearl, Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference
- 199x Efficient algorithm
- 199x Bayesian nets used in several industrial applications
- 199x First commercially available Bayesian net analysis software available

Example from Medical Diagnostics

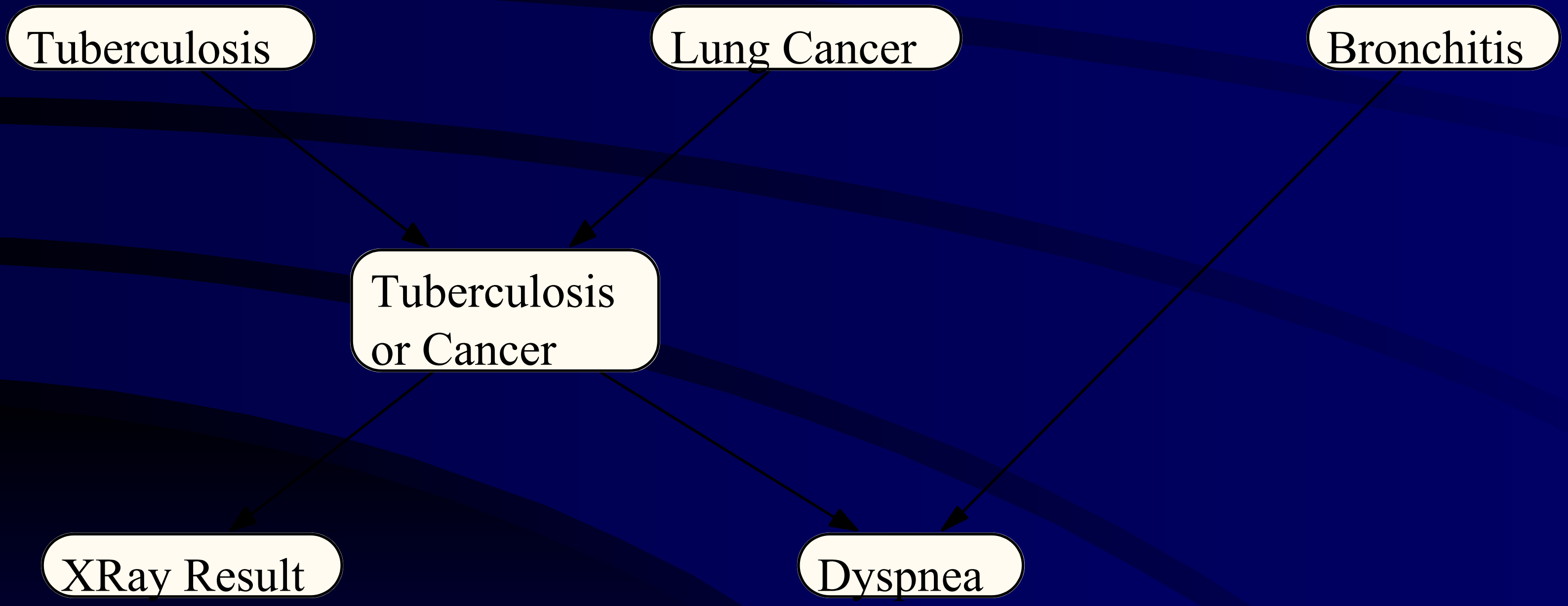


- **Finished with interviewing the patient, the physician begins the examination**
- **The physician determines that the patient is having difficulty breathing, the finding "Dyspnea" is "Present" is entered and propagated through the network**
- **Note that the information from this finding propagates backward through the arcs**

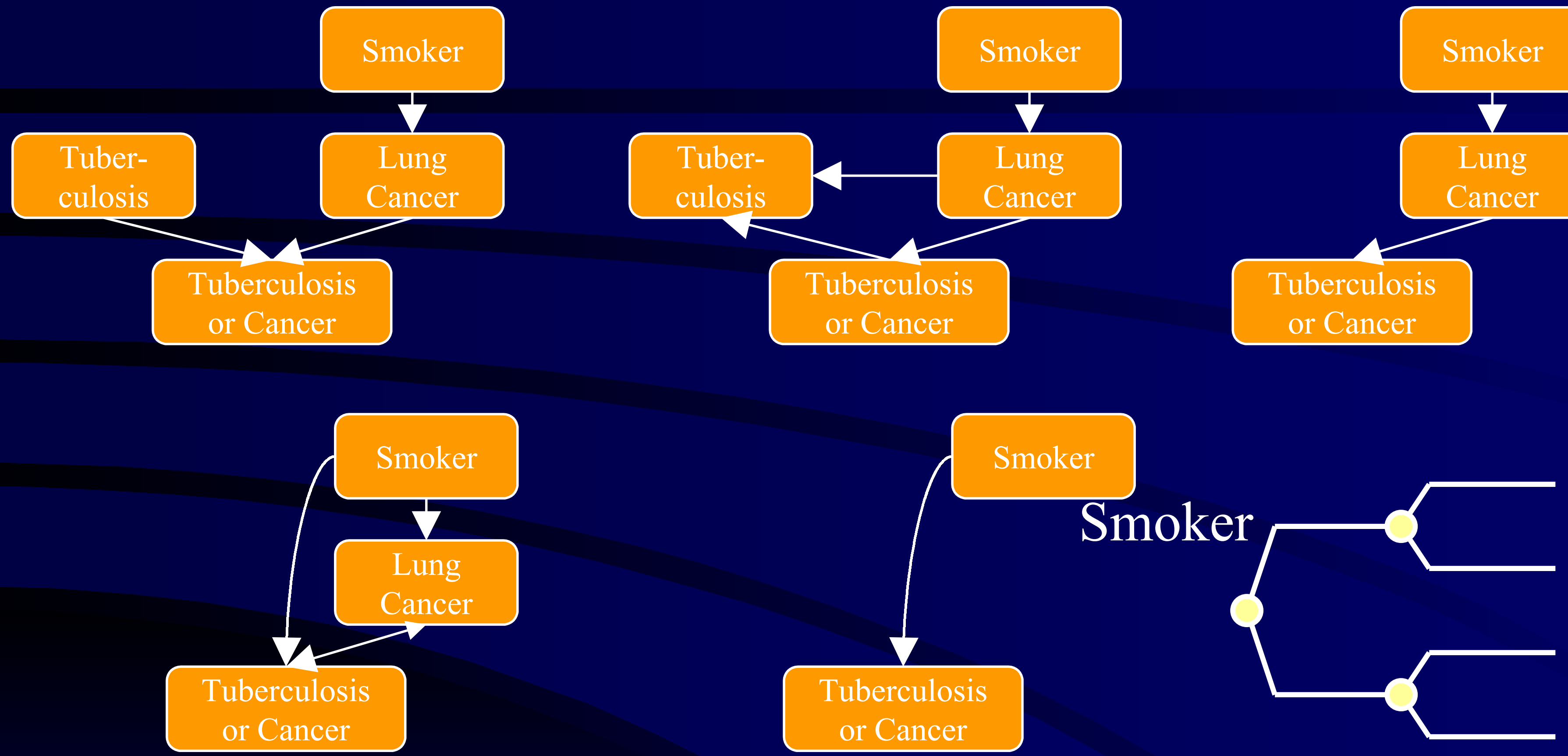
Example from Medical Diagnostics



- The physician now moves to specific diagnostic tests such as an X-Ray, which results in a "Normal" finding which propagates through the network
- The doctor might now conclude that the evidence strongly indicates the patient has bronchitis and does not have tuberculosis or lung cancer


Tuberculosis
Lung Cancer
Bronchitis
Tuberculosis or Cancer
XRay Result
Dyspnea

Inference Using Bayes Theorem



- **The general probabilistic inference problem is to find the probability of an event given a set of evidence**
- **This can be done in Bayesian nets with sequential applications of Bayes Theorem**

Sample Chain - Setup

(1) Set all lambdas to be a vector of 1's; $Bel(SM) = \alpha\, \lambda(SM) \blacksquare \pi(SM)$

	$\pi(SM)$	Bel(SM)	$\lambda(SM)$
Paris	0.9	0.9	1.0
Med.	0.1	0.1	1.0

(2) $\pi(TO) = \pi(SM)\, M_{TO|SM}$; $Bel(TO) = \alpha\, \lambda(TO) \blacksquare \pi(TO)$

	$\pi(TO)$	Bel(TO)	$\lambda(TO)$
Chalons	0.73	0.73	1.0
Dijon	0.27	0.27	1.0

(3) $\pi(AA) = \pi(TO)\, M_{AA|TO}$; $Bel(AA) = \alpha\, \lambda(AA) \blacksquare \pi(AA)$

	$\pi(AA)$	Bel(AA)	$\lambda(AA)$
North	0.39	0.73	1.0
Central	0.35	0.27	1.0
South	0.24	0.24	1.0

$$M_{TO|SM} = \begin{bmatrix} .8 & .2 \\ .1 & .9 \end{bmatrix} \qquad M_{AA|TO} = \begin{bmatrix} .5 & .4 & .1 \\ .1 & .3 & .6 \end{bmatrix}$$