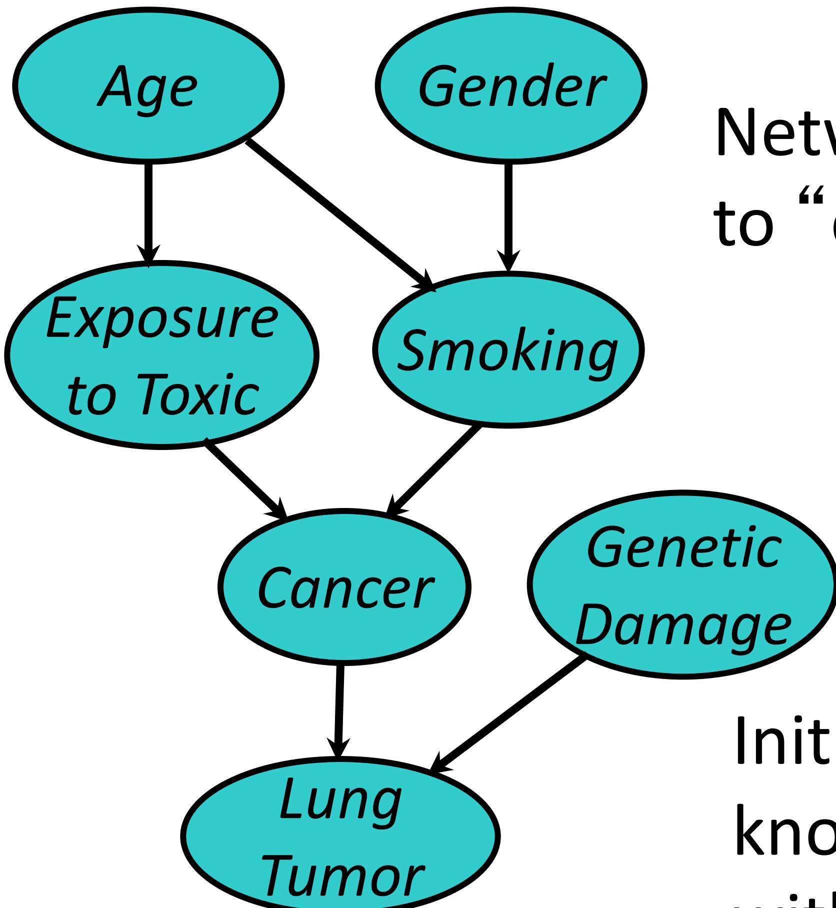
CMSC 471:
Reasoning with Bayesian Belief Network

Chapters 12 & 13

KMA Solaiman – ksolaima@umbc.edu

Some slides courtesy Tim Finin and Frank Ferraro

KA2: Structuring Bayesian Belief Network



Network structure corresponding to "causality" is usually good.

Initially this uses the designer's knowledge but can be checked with data

KA3: The Numbers

- For each variable we have a table of probability of its value for values of its **parents**
- For variables w/o parents, we have **prior probabilities**

$$S \in \{no, light, heavy\}$$
$$C \in \{none, benign, malignant\}$$

smoking priors	
no	0.80
light	0.15
heavy	0.05

	smoking		
cancer	**no**	**light**	**heavy**
none	0.96	0.88	0.60
benign	0.03	0.08	0.25
malignant	0.01	0.04	0.15

Three (Four) kinds of reasoning

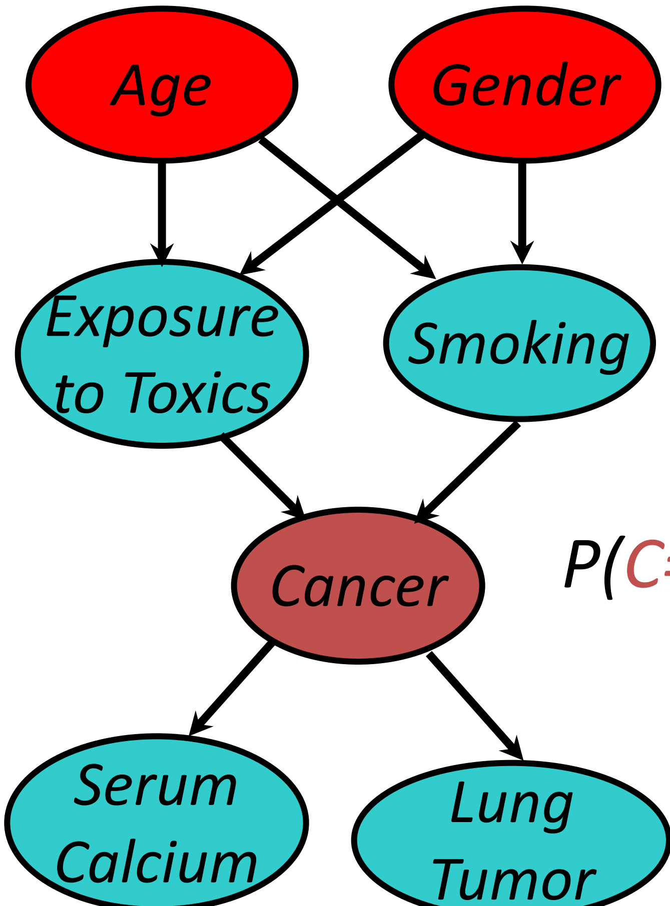
BBNs support three main kinds of reasoning:

- **Predicting** conditions given predispositions
- **Diagnosing** conditions given symptoms (and predisposing)
- **Explaining** a condition by one or more predispositions

To which we can add a fourth:

- **Deciding** on an action based on probabilities of the conditions

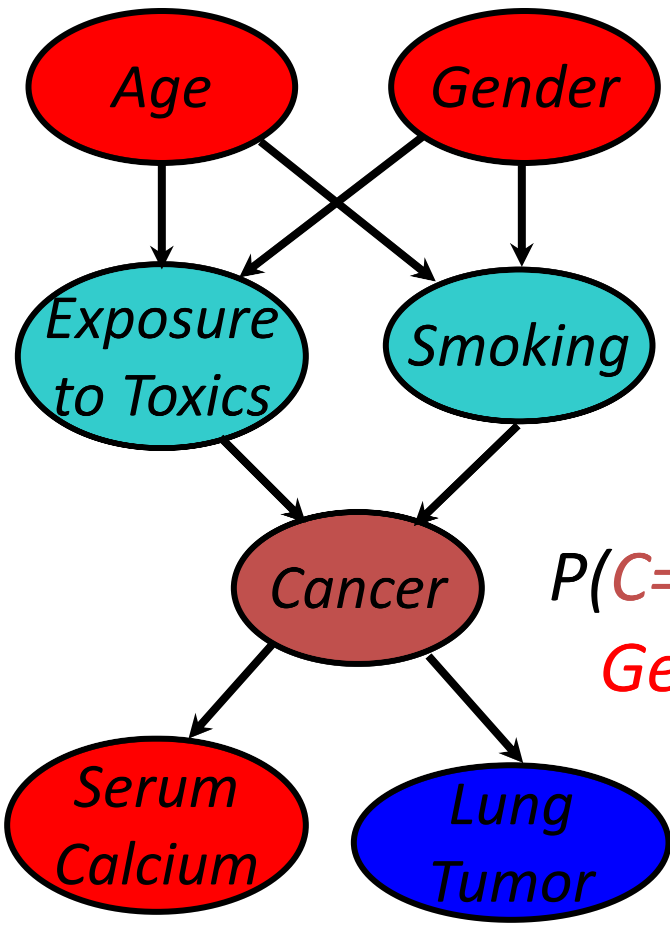
Predictive Inference



How likely are elderly males to get malignant cancer?

P(C=malignant | Age>60, Gender=male)

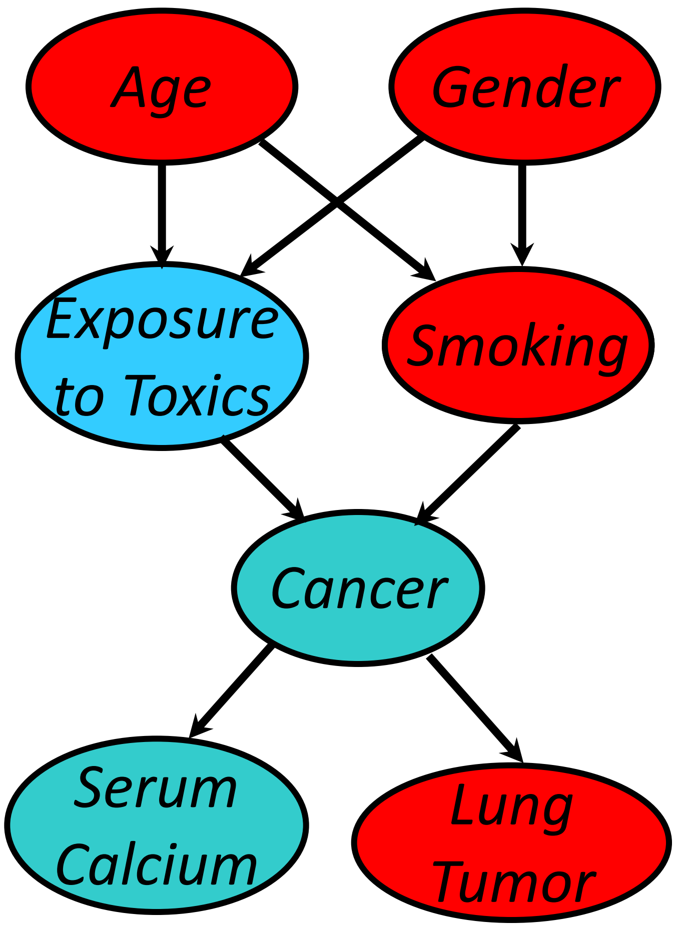
Predictive and diagnostic combined



How likely is an elderly male patient with high Serum Calcium to have malignant cancer?

P(C=malignant | Age>60, Gender= male, Serum Calcium = high)

Explaining away



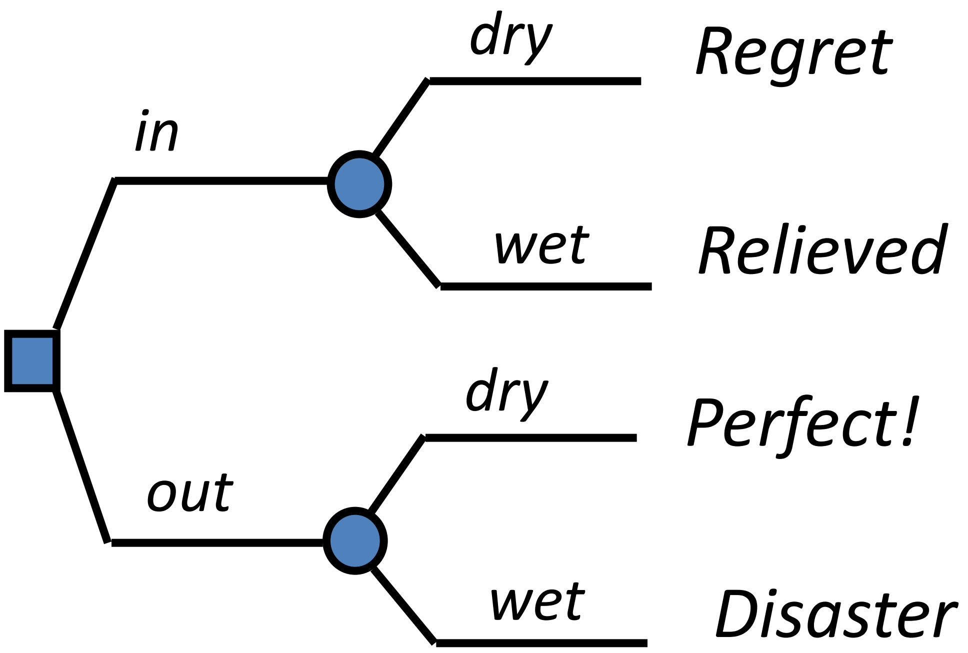
- If we see a lung tumor, the probability of heavy smoking and of exposure to toxics both go up
- If we then observe heavy smoking, the probability of exposure to toxics goes back down

Decision making

- A decision is a medical domain might be a choice of treatment (e.g., radiation or chemotherapy)
- Decisions should be made to **maximize expected utility**
- View decision making in terms of
  - Beliefs/Uncertainties
  - Alternatives/Decisions
  - Objectives/Utilities

Decision Problem

Should I have my party inside or outside?



Decision Making with BBNs

- Today's weather forecast might be either sunny, cloudy or rainy
- Should you take an umbrella when you leave?
- Your decision depends only on the forecast
  - The forecast "depends on" the actual weather
- Your satisfaction depends on your decision and the weather
  - Assign a utility to each of four situations: (rain|no rain) x (umbrella, no umbrella)

Decision Making with BBNs

- Extend BBN framework to include two new kinds of nodes: **decision** and **utility**
- **Decision** node computes the expected utility of a decision given its parent(s) (e.g., forecast) and a valuation
- **Utility** node computes utility value given its parents, e.g. a decision and weather
  - Assign utility to each situations: (rain|no rain) x (umbrella, no umbrella)
  - Utility value assigned to each is probably subjective

Fundamental Inference & Learning Question

- Compute posterior probability of a node given some other nodes

$$p(Q|x_1, \dots, x_j)$$

- Some techniques
  - MLE (maximum likelihood estimation)/MAP (maximum a posteriori) [covered 2nd]
  - Variable Elimination [covered 1st]
  - (Loopy) Belief Propagation ((Loopy) BP)
  - Monte Carlo
  - Variational methods
  - …

Advanced topics

Variable Elimination

- Inference: Compute posterior probability of a node given some other nodes

$$p(Q|x_1, \dots, x_j)$$

- Variable elimination: An algorithm for exact inference
  - Uses dynamic programming
  - Not necessarily polynomial time!

Variable Elimination (High-level)

Goal: $p(Q|x_1, \ldots, x_j)$

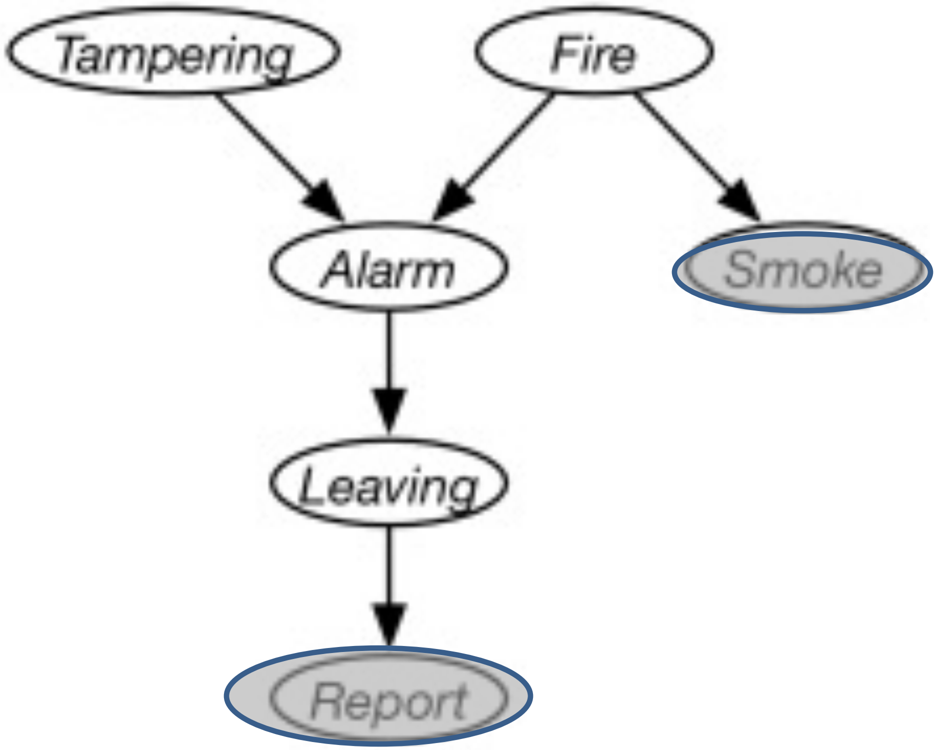
(The word "factor" is used for each CPT.)

1. Pick one of the non-conditioned, MB variables
2. Eliminate this variable by marginalizing (summing) it out from all factors (CPTs) that contain it
3. Go back to 1 until no (MB) variables remain
4. Multiply the remaining factors and normalize.

Variable Elimination: Example

(The word “factor” is used for each CPT.)

1. Pick one of the non-conditioned, MB variables
2. Eliminate this variable by marginalizing (summing) it out from all factors (CPTs) that contain it
3. Go back to 1 until no (MB) variables remain
4. Multiply the remaining factors and normalize.

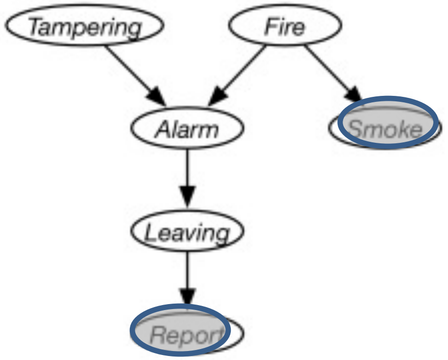


Goal: P(Tampering | Smoke=true ∧ Report=true)

Variable Elimination: Example

(The word "factor" is used for each CPT.)

1. Pick one of the non-conditioned, MB variables
2. Eliminate this variable by marginalizing (summing) it out from all factors (CPTs) that contain it
3. Go back to 1 until no (MB) variables remain
4. Multiply the remaining factors and normalize.



Goal: P(Tampering | Smoke=true ∧ Report=true)

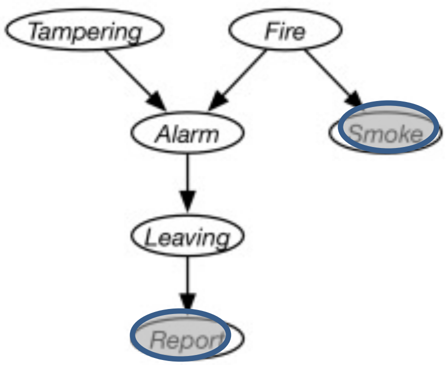
$ConditionalProbability$	$Factor$
$P(Tampering)$	$f_0(Tampering)$
$P(Fire)$	$f_1(Fire)$
$P(Alarm \mid Tampering, Fire)$	$f_2(Tampering, Fire, Alarm)$
$P(Smoke = yes \mid Fire)$	$f_3(Fire)$
$P(Leaving \mid Alarm)$	$f_4(Alarm, Leaving)$
$P(Report = yes \mid Leaving)$	$f_5(Leaving)$

Variable Elimination: Example

(The word "factor" is used for each CPT.)

1. Pick one of the non-conditioned, MB variables
2. Eliminate this variable by marginalizing (summing) it out from all factors (CPTs) that contain it
3. Go back to 1 until no (MB) variables remain
4. Multiply the remaining factors and normalize.

$Conditional Probability$	$Factor$
$P(Tampering)$	$f_0(Tampering)$
$P(Fire)$	$f_1(Fire)$
$P(Alarm \mid Tampering, Fire)$	$f_2(Tampering, Fire, Alarm)$
$P(Smoke = yes \mid Fire)$	$f_3(Fire)$
$P(Leaving \mid Alarm)$	$f_4(Alarm, Leaving)$
$P(Report = yes \mid Leaving)$	$f_5(Leaving)$



Goal: P(Tampering | Smoke=true ∧ Report=true)

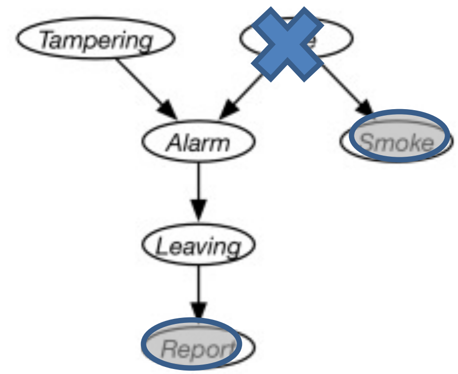
Task: Eliminate Fire

Variable Elimination: Example

(The word "factor" is used for each CPT.)

1. Pick one of the non-conditioned, MB variables
2. Eliminate this variable by marginalizing (summing) it out from all factors (CPTs) that contain it
3. Go back to 1 until no (MB) variables remain
4. Multiply the remaining factors and normalize.

$Conditional Probability$	$Factor$
$P(Tampering)$	$f_0(Tampering)$
$P(Fire)$	$f_1(Fire)$
$P(Alarm \mid Tampering, Fire)$	$f_2(Tampering, Fire, Alarm)$
$P(Smoke = yes \mid Fire)$	$f_3(Fire)$
$P(Leaving \mid Alarm)$	$f_4(Alarm, Leaving)$
$P(Report = yes \mid Leaving)$	$f_5(Leaving)$



Goal: P(Tampering | Smoke=true ∧ Report=true)

f1(Fire)

f2(Tampering, Fire, Alarm)

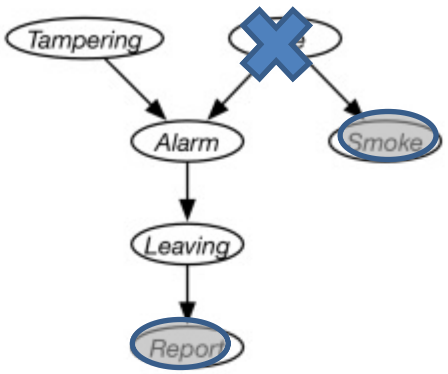
f3(Fire)

f6(Tampering, Alarm) =

$$= \sum_u f_1(\mathrm{Fire} = u) f_2(T, F = u, A) f_3(F = u)$$

$$= \sum_u p(\mathrm{Fire} = u) p(A \mid T, F = u) p(S = y \mid F = u)$$

Variable Elimination: Example



(The word "factor" is used for each CPT.)

1. Pick one of the non-conditioned, MB variables
2. Eliminate this variable by marginalizing (summing) it out from all factors (CPTs) that contain it
3. Go back to 1 until no (MB) variables remain
4. Multiply the remaining factors and normalize.

Goal: P(Tampering | Smoke=true ∧ Report=true)

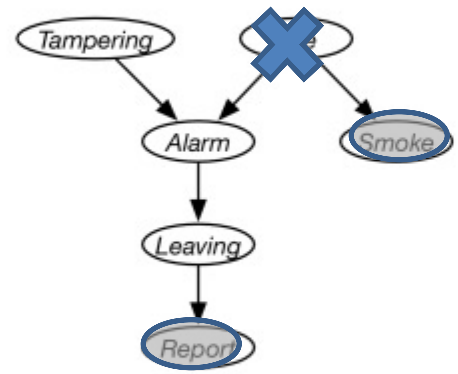
f6(Tampering, Alarm) =

$$= \sum_u p(\text{Fire} = u) p(A \mid T, F = u) p(S = y \mid F = u)$$

$$= p(\text{Fire} = y) p(A \mid T, F = y) p(S = y \mid F = y) + p(\text{Fire} = n) p(A \mid T, F = n) p(S = y \mid F = n)$$

$Conditional Probability$	$Factor$
$P(Tampering)$	$f_0(Tampering)$
$P(Fire)$	$f_1(Fire)$
$P(Alarm \mid Tampering, Fire)$	$f_2(Tampering, Fire, Alarm)$
$P(Smoke = yes \mid Fire)$	$f_3(Fire)$
$P(Leaving \mid Alarm)$	$f_4(Alarm, Leaving)$
$P(Report = yes \mid Leaving)$	$f_5(Leaving)$

Variable Elimination: Example



(The word "factor" is used for each CPT.)

1. Pick one of the non-conditioned, MB variables
2. Eliminate this variable by marginalizing (summing) it out from all factors (CPTs) that contain it
3. Go back to 1 until no (MB) variables remain
4. Multiply the remaining factors and normalize.

$Conditional Probability$	$Factor$
$P(Tampering)$	$f_0(Tampering)$
$P(Fire)$	$f_1(Fire)$
$P(Alarm \mid Tampering, Fire)$	$f_2(Tampering, Fire, Alarm)$
$P(Smoke = yes \mid Fire)$	$f_3(Fire)$
$P(Leaving \mid Alarm)$	$f_4(Alarm, Leaving)$
$P(Report = yes \mid Leaving)$	$f_5(Leaving)$

Goal: P(Tampering | Smoke=true ∧ Report=true)

f6(Tampering, Alarm) =

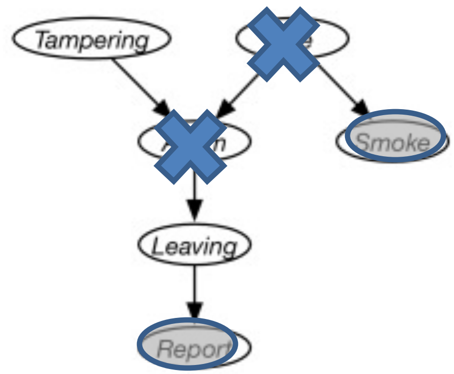
$$= \sum_u p(\text{Fire} = u) p(A \mid T, F = u) p(S = y \mid F = u)$$

Tamp.	Alarm	f6
Yes	Yes	$p(\text{Fire} = y) p(A = y \mid T = y, F = y) p(S = y \mid F = y) + p(\text{Fire} = n) p(A = y \mid T = y, F = n) p(S = y \mid F = n)$
Yes	No	…
No	No	…
No	Yes	…

Variable Elimination: Example

(The word "factor" is used for each CPT.)

1. Pick one of the non-conditioned, MB variables
2. Eliminate this variable by marginalizing (summing) it out from all factors (CPTs) that contain it
3. Go back to 1 until no (MB) variables remain
4. Multiply the remaining factors and normalize.



Goal: P(Tampering | Smoke=true ∧ Report=true)

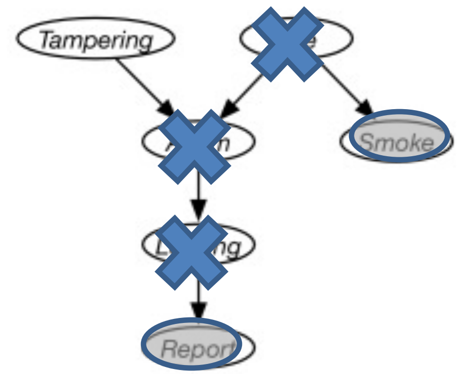
Task: Eliminate Alarm

$Conditional Probability$	$Factor$
$P(Tampering)$	$f_0(Tampering)$
$P(Fire)$	$f_1(Fire)$
$P(Alarm \mid Tampering, Fire)$	$f_2(Tampering, Fire, Alarm)$
$P(Smoke = yes \mid Fire)$	$f_3(Fire)$
$P(Leaving \mid Alarm)$	$f_4(Alarm, Leaving)$
$P(Report = yes \mid Leaving)$	$f_5(Leaving)$

Variable Elimination: Example

(The word "factor" is used for each CPT.)

1. Pick one of the non-conditioned, MB variables
2. Eliminate this variable by marginalizing (summing) it out from all factors (CPTs) that contain it
3. Go back to 1 until no (MB) variables remain
4. Multiply the remaining factors and normalize.



Goal: P(Tampering | Smoke=true ∧ Report=true)

ConditionalProbability	*Factor*
$P(Tampering)$	$f_0(Tampering)$
$P(Fire)$	$f_1(Fire)$
$P(Alarm \mid Tampering, Fire)$	$f_2(Tampering, Fire, Alarm)$
$P(Smoke = yes \mid Fire)$	$f_3(Fire)$
$P(Leaving \mid Alarm)$	$f_4(Alarm, Leaving)$
$P(Report = yes \mid Leaving)$	$f_5(Leaving)$

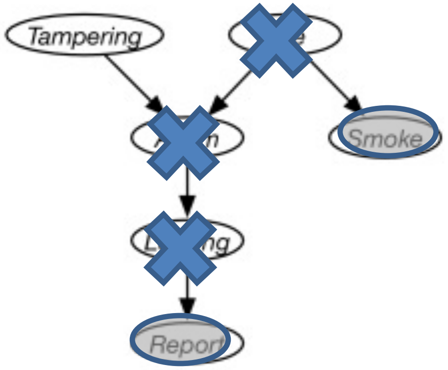
…other computations not shown---see the book or lecture…

PM example 9.27

Variable Elimination: Example

(The word "factor" is used for each CPT.)

1. Pick one of the non-conditioned, MB variables
2. Eliminate this variable by marginalizing (summing) it out from all factors (CPTs) that contain it
3. Go back to 1 until no (MB) variables remain
4. Multiply the remaining factors and normalize.



Goal: P(Tampering | Smoke=true ∧ Report=true)

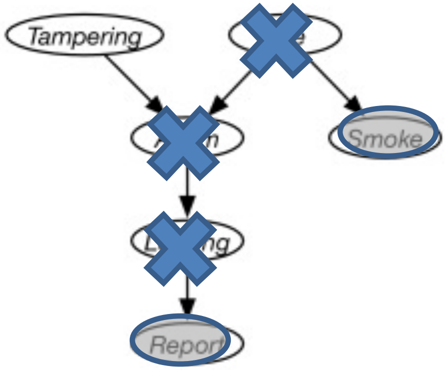
Task: Normalize in order to compute p(Tampering)

We'll have a single factor f9(Tampering):

$$p(T = u) = \frac{f_9(T = u)}{\sum_v f_9(T = v)}$$

$Conditional Probability$	$Factor$
$P(Tampering)$	$f_0(Tampering)$
$P(Fire)$	$f_1(Fire)$
$P(Alarm \mid Tampering, Fire)$	$f_2(Tampering, Fire, Alarm)$
$P(Smoke = yes \mid Fire)$	$f_3(Fire)$
$P(Leaving \mid Alarm)$	$f_4(Alarm, Leaving)$
$P(Report = yes \mid Leaving)$	$f_5(Leaving)$

Variable Elimination: Example



Goal: P(Tampering | Smoke=true ∧ Report=true)

(The word "factor" is used for each CPT.)

1. Pick one of the non-conditioned, MB variables
2. Eliminate this variable by marginalizing (summing) it out from all factors (CPTs) that contain it
3. Go back to 1 until no (MB) variables remain
4. Multiply the remaining factors and normalize.

Conditional Probability	Factor
$P(Tampering)$	$f_0(Tampering)$
$P(Fire)$	$f_1(Fire)$
$P(Alarm \mid Tampering, Fire)$	$f_2(Tampering, Fire, Alarm)$
$P(Smoke = yes \mid Fire)$	$f_3(Fire)$
$P(Leaving \mid Alarm)$	$f_4(Alarm, Leaving)$
$P(Report = yes \mid Leaving)$	$f_5(Leaving)$

Task: Normalize in order to compute ***p(Tampering)***

We'll have a single factor f9(Tampering):

$$p(T = yes) = \frac{f_9(T = yes)}{f_9(T = yes) + f_9(T = no)}$$

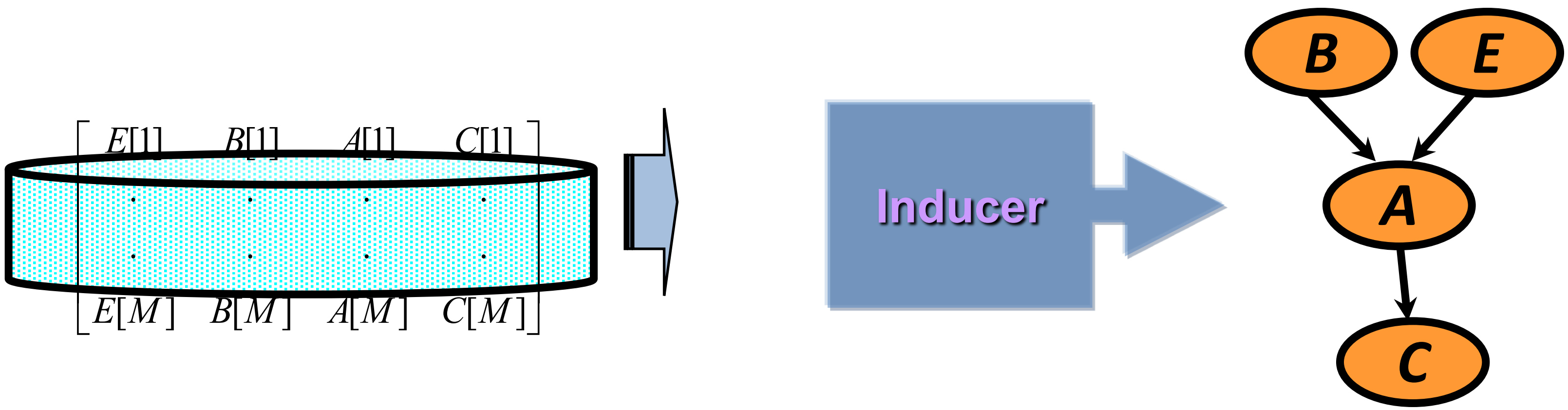
Variable Elimination: Example

- The posterior distribution over *Tampering* is given by

$$\frac{P(Tampering = u)\ f_9(Tampering = u)}{\sum_v P(Tampering = v)\ f_9(Tampering = v)}$$

Learning Bayesian networks

- Given training set $D = \{x[1], \ldots, x[M]\}$
- Find graph that best matches ***D***
  - model selection
  - parameter estimation



Data D

Learning Bayesian Networks

- Describe a BN by specifying its (1) structure and (2) conditional probability tables (CPTs)
- Both can be learned from data, but
  - learning structure much harder than learning parameters
  - learning when some nodes are hidden, or with missing data harder still
- Four cases:

Structure	*Observability*	*Method*
Known	Full	Maximum Likelihood Estimation
Known	Partial	EM (or gradient ascent)
Unknown	Full	Search through model space
Unknown	Partial	EM + search through model space

Variations on a theme

- **Known structure, fully observable**: only need to do parameter estimation
- **Unknown structure, fully observable:** do heuristic search through structure space, then parameter estimation
- **Known structure, missing values:** use expectation maximization (EM) to estimate parameters
- **Known structure, hidden variables:** apply adaptive probabilistic network (APN) techniques
- **Unknown structure, hidden variables:** too hard to solve!

Fundamental Inference Question

- Compute posterior probability of a node given some other nodes

$$p(Q|x_1, \ldots, x_j)$$

- Some techniques
  - MLE (maximum likelihood estimation)/MAP (maximum a posteriori) [covered 2nd]
  - Variable Elimination [covered 1st]
  - (Loopy) Belief Propagation ((Loopy) BP)
  - Monte Carlo
  - Variational methods
  - …

Advanced topics

Parameter estimation

- Assume known structure
- Goal: estimate BN parameters θ
  - entries in local probability models, P(X | Parents(X))
- A parameterization θ is good if it is likely to generate the observed data:

$$L(\theta : D) = P(D \mid \theta) = \prod_{m} P(x[m] \mid \theta)$$

i.i.d. samples

- Maximum Likelihood Estimation (MLE) Principle: Choose θ* so as to maximize *L*

Parameter estimation II

- The likelihood **decomposes** according to the structure of the network

  → we get a separate estimation task for each parameter
- The MLE (maximum likelihood estimate) solution for **discrete** data & RV values:
  - for each value *x* of a node *X*
  - and each instantiation ***u*** of *Parents(X)*

$$\theta^*_{\boldsymbol{x}|\boldsymbol{u}} = \frac{N(\boldsymbol{x}, \boldsymbol{u})}{N(\boldsymbol{u})}$$

sufficient statistics

  - Just need to collect the counts for every combination of parents and children observed in the data
  - MLE is equivalent to an assumption of a uniform prior over parameter values

Estimating Probability of Heads

X=1 X=0

- I show you the above coin X, and hire you to estimate the probability that it will turn up heads ($X = 1$) or tails ($X = 0$)
- You flip it repeatedly, observing
  - it turns up heads α_1 times
  - it turns up tails α_0 times
- Your estimate for $P(X = 1)$ is....?

$$\hat{P}(X=1) = \frac{\alpha_1}{\alpha_1 + \alpha_0}$$

Estimating θ = P(X=1)

X=1 X=0

Test A:

α_1 α_0

100 flips: 51 Heads (X=1), 49 Tails (X=0)

$$\frac{\alpha_1}{\alpha_1+\alpha_0} = \frac{51}{100} \rightarrow \hat{P}(X=1) = 0.51$$

Test B:

α_1 α_0

3 flips: 2 Heads (X=1), 1 Tails (X=0)

$$= \frac{2}{2+1} = 0.666$$

Maximum Likelihood Estimation

X=1 X=0

P(X=1) = θ P(X=0) = (1-θ)

Data D: $= \{1 \quad 0 \quad 0 \quad 1\} \quad 1$

$P(D|\theta) = \theta \cdot (1-\theta) \cdot (1-\theta) \cdot \theta \cdot \theta = \theta^{\alpha_1} (1-\theta)^{\alpha_0}$

Flips produce data D with α_1 heads, α_0 tails

- flips are independent, identically distributed 1's and 0's (Bernoulli)
- α_1 and α_0 are counts that sum these outcomes (Binomial)

$$P(D|\theta) = P(\alpha_1, \alpha_0|\theta) = \theta^{\alpha_1}(1-\theta)^{\alpha_0}$$

Maximum Likelihood Estimate for Θ

$$\begin{aligned} \widehat{\theta} &= \arg\max_{\theta} \ \ln P(\mathcal{D} \mid \theta) \\ &= \arg\max_{\theta} \ \ln \theta^{\alpha_H} (1-\theta)^{\alpha_T} \end{aligned}$$

- Set derivative to zero: $\boxed{\frac{d}{d\theta} \ln P(\mathcal{D} \mid \theta) = 0}$

[C. Guestrin]

$$\hat{\theta} = \arg\max_{\theta} \ \ln P(D|\theta)$$

$$= \arg\max_{\theta} \ \ln \left[\theta^{\alpha_1}(1-\theta)^{\alpha_0}\right]$$

- Set derivative to zero: $\frac{d}{d\theta} \ln P(\mathcal{D} \mid \theta) = 0$

hint: $\frac{\partial \ln \theta}{\partial \theta} = \frac{1}{\theta}$

$$\frac{\partial}{\partial\theta} \ \alpha_1 \ln\theta + \alpha_0 \ln(1-\theta)$$

$$\alpha_1 \frac{1}{\theta} + \alpha_0 \underbrace{\frac{\partial \ln(1-\theta)}{\partial\theta}}$$

$$\underbrace{\frac{\partial \ln(1-\theta)}{\partial(1-\theta)}}_{\frac{1}{1-\theta}} \cdot \underbrace{\frac{\partial(1-\theta)}{\partial\theta}}_{-1}$$

$$\boxed{0 = \alpha_1 \frac{1}{\theta} - \frac{\alpha_0}{1-\theta}}$$

$$\theta = \frac{\alpha_1}{\alpha_1 + \alpha_0}$$

Summary:
Maximum Likelihood Estimate

X=1 X=0

P(X=1) = θ
P(X=0) = 1-θ
(Bernoulli)

- Each flip yields boolean value for X

$$X \sim \text{Bernoulli: } P(X) = \theta^X (1-\theta)^{(1-X)}$$

- Data set D of independent, identically distributed (iid) flips produces α_1 ones, α_0 zeros (Binomial)

$$P(D|\theta) = P(\alpha_1, \alpha_0|\theta) = \theta^{\alpha_1}(1-\theta)^{\alpha_0}$$

$$\hat{\theta}^{MLE} = \text{argmax}_\theta \, P(D|\theta) = \frac{\alpha_1}{\alpha_1+\alpha_0}$$

Learning: Maximum Likelihood Estimation (MLE)

Core concept in intro statistics:

- Observe some data $\mathcal{X}$
- Compute some distribution $g(\mathcal{X})$ to {predict, explain, generate} $\mathcal{X}$
- Assume g is controlled by parameters ϕ, i.e., $g_\phi(\mathcal{X})$
  - Sometimes written $g(\mathcal{X}; \phi)$
- Learning appropriate value(s) of ϕ allows you to **GENERALIZE** about $\mathcal{X}$

Learning: Maximum Likelihood Estimation (MLE)

Central to machine learning:

- Observe some data $(\mathcal{X}, \mathcal{Y})$
- Compute some function $f(\mathcal{X})$ to {predict, explain, generate} $\mathcal{Y}$
- Assume f is controlled by parameters θ, i.e., $f_\theta(\mathcal{X})$
  - Sometimes written $f(\mathcal{X}; \theta)$

Learning Parameters for the Die Model

$$p(w_1, w_2, \ldots, w_N) = p(w_1)p(w_2)\cdots p(w_N) = \prod_i p(w_i)$$

maximize (log-) likelihood to learn the probability parameters

Q: Why is maximizing log-likelihood a reasonable thing to do?

Learning Parameters for the Die Model

$$p(w_1, w_2, \dots, w_N) = p(w_1)p(w_2)\cdots p(w_N) = \prod_i p(w_i)$$

maximize (log-) likelihood to learn the probability parameters

Q: Why is maximizing log-likelihood a reasonable thing to do?

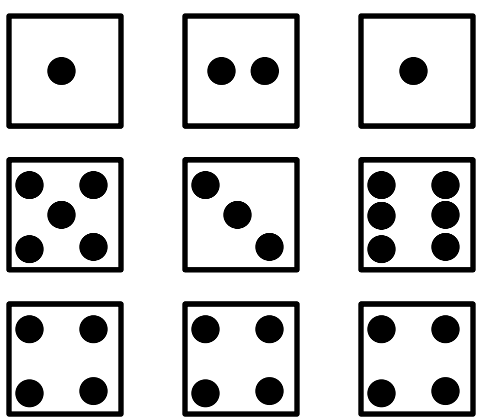
A: Develop a good model for what we observe

Learning Parameters for the Die Model: Maximum Likelihood (Intuition)

$$p(w_1, w_2, \ldots, w_N) = p(w_1)p(w_2)\cdots p(w_N) = \prod_i p(w_i)$$

maximize (log-) likelihood to learn the probability parameters

If you observe these 9 rolls…



…what are "reasonable" estimates for p(w)?

p(1) = ?

p(2) = ?

p(3) = ?

p(4) = ?

p(5) = ?

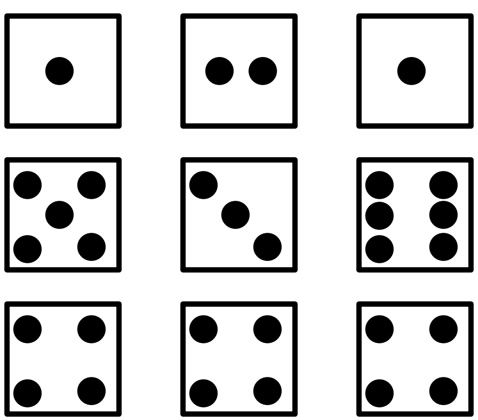
p(6) = ?

Learning Parameters for the Die Model: Maximum Likelihood (Intuition)

$$p(w_1, w_2, \ldots, w_N) = p(w_1)p(w_2) \cdots p(w_N) = \prod_i p(w_i)$$

maximize (log-) likelihood to learn the probability parameters

If you observe these 9 rolls…



…what are "reasonable" estimates for p(w)?

p(1) = 2/9 p(2) = 1/9

p(3) = 1/9 p(4) = 3/9

p(5) = 1/9 p(6) = 1/9

maximum likelihood estimates

Learning: Maximum Likelihood Estimation (MLE)

Core concept in intro statistics:

- Observe some data $\mathcal{X}$
- Compute some distribution $g(\mathcal{X})$ to {predict, explain, generate} $\mathcal{X}$
- Assume g is controlled by parameters ϕ, i.e., $g_\phi(\mathcal{X})$
  - Sometimes written $g(\mathcal{X}; \phi)$
- Learning appropriate value(s) of ϕ allows you to **GENERALIZE** about $\mathcal{X}$

How do we "learn appropriate value(s) of ϕ?"

Many different options: a common one is **maximum likelihood estimation (MLE)**

- Find values ϕ s.t. $g_\phi(\mathcal{X} = \{x_1, \ldots, x_N\})$ is maximized
- Independence assumptions are very useful here!
- Logarithms are also useful!

Learning: Maximum Likelihood Estimation (MLE)

Core concept in intro statistics:

- Observe some data $\mathcal{X}$
- Compute some distribution $g(\mathcal{X})$ to {predict, explain, generate} $\mathcal{X}$
- Assume g is controlled by parameters ϕ, i.e., $g_\phi(\mathcal{X})$
  - Sometimes written $g(\mathcal{X}; \phi)$
- MLE: Find values ϕ s.t. $g_\phi(\mathcal{X} = \{x_1, \dots, x_N\})$ is maximized

Example: How much does it snow?

- $\mathcal{X} = \{x_1, x_2, \dots, x_N\}$ are snowfall values from the previous N storms
- Goal: learn ϕ such that g correctly models, as accurately as possible, the amount of snow likely

Advanced topic

Learning: Maximum Likelihood Estimation (MLE)

Core concept in intro statistics:

- Observe some data $\mathcal{X}$
- Compute some distribution $g(\mathcal{X})$ to {predict, explain, generate} $\mathcal{X}$
- Assume g is controlled by parameters ϕ, i.e., $g_\phi(\mathcal{X})$
  - Sometimes written $g(\mathcal{X}; \phi)$
- MLE: Find values ϕ s.t. $g_\phi(\mathcal{X} = \{x_1, \dots, x_N\})$ is maximized

Example: How much does it snow?

- $\mathcal{X} = \{x_1, x_2, \dots, x_N\}$ are snowfall values from the previous N storms
- Goal: learn ϕ such that g correctly models, as accurately as possible, the amount of snow likely
- Assumption: each x_i is independent from all others

$$\max_\phi \sum_{i=1}^{N} \log g_\phi(x_i)$$

Advanced topic

MLE Snowfall Example

Example: How much does it snow?

- $\mathcal{X} = \{x_1, x_2, \dots, x_N\}$ are snowfall values from the previous N storms
- Goal: learn ϕ such that g correctly models, as accurately as possible, the amount of snow likely
- Assumption: each x_i is independent from all others

$$\max_{\phi} \sum_{i=1}^{N} \log g_{\phi}(x_i)$$

Q: Why is taking logarithms okay?

Q: What other assumptions, or decisions, do we need to make?

MLE Snowfall Example

Example: How much does it snow?

- $\mathcal{X} = \{x_1, x_2, \dots, x_N\}$ are snowfall values from the previous N storms
- Goal: learn ϕ such that g correctly models, as accurately as possible, the amount of snow likely
- Assumption: each x_i is independent from all others, but all from g

$$\max_{\phi} \sum_{i=1}^{N} \log g_{\phi}(x_i)$$

Q: Why is taking logarithms okay?

Q: What other assumptions, or decisions, do we need to make?

x_i is positive, real-valued. What's a faithful probability distribution for x_i?

- Normal? ✗
- Gamma? ✓
- Exponential? ✓
- Bernoulli? ✗
- Poisson? ✗

MLE Snowfall Example

Example: How much does it snow?

- $\mathcal{X} = \{x_1, x_2, \dots, x_N\}$ are snowfall values from the previous N storms
- Goal: learn ϕ such that g correctly models, as accurately as possible, the amount of snow likely
- Assumption: each x_i is independent from all others, but all from g

$$\max_{\phi} \sum_{i=1}^{N} \log g_{\phi}(x_i)$$

Q: Why is taking logarithms okay?

Q: What other assumptions, or decisions, do we need to make?

x_i is positive, real-valued. What's a faithful probability distribution for x_i?

- Normal? ✗
- Gamma? ✓ $p(X = x) = \frac{x^{k-1} \exp(\frac{-k}{\theta})}{\theta^k \Gamma(k)}$
- Exponential? ✓
- Bernoulli? ✗
- Poisson? ✗

Advanced topic

MLE Snowfall Example

Example: How much does it snow?

- $\mathcal{X} = \{x_1, x_2, \dots, x_N\}$ are snowfall values from the previous N storms
- Goal: learn ϕ such that g correctly models, as accurately as possible, the amount of snow likely
- Assumption: each x_i is independent from all others, but all from g

$$\max_{\phi} \sum_{i=1}^{N} \log g_{\phi}(x_i)$$

Q: Why is taking logarithms okay?

Q: What other assumptions, or decisions, do we need to make?

x_i is positive, real-valued. What's a faithful/nice-to-compute-and-good-enough probability distribution for x_i?

- Normal? ✗ ✓
- Gamma? ✓ ?
- Exponential? ✓ ?
- Bernoulli? ✗ ✗
- Poisson? ✗ ✗

$$p(X = x) = \frac{1}{\sqrt{2\pi\sigma}} \exp\left(\frac{-(x-\mu)^2}{2\sigma^2}\right)$$

Advanced topic

MLE Snowfall Example

Example: How much does it snow?

- $\mathcal{X} = \{x_1, x_2, \ldots, x_N\}$ are snowfall values from the previous N storms
- Goal: learn ϕ such that g correctly models, as accurately as possible, the amount of snow likely
- Assumption: each x_i is independent from all others, but all from g

$$\max_{\phi} \sum_{i=1}^{N} \log g_{\phi}(x_i)$$

$$x_i \sim \mathrm{Normal}(\mu, \sigma^2)$$

$$\max_{(\mu,\sigma^2)} \sum_{i=1}^{N} \log \mathrm{Normal}_{\mu,\sigma^2}(x_i) =$$

Advanced topic

MLE Snowfall Example

Example: How much does it snow?

- $\mathcal{X} = \{x_1, x_2, \ldots, x_N\}$ are snowfall values from the previous N storms
- Goal: learn ϕ such that g correctly models, as accurately as possible, the amount of snow likely
- Assumption: each x_i is independent from all others, but all from g

$$\max_{\phi} \sum_{i=1}^{N} \log g_{\phi}(x_i)$$

$$x_i \sim \text{Normal}(\mu, \sigma^2)$$

$$\max_{(\mu,\sigma^2)} \sum_{i=1}^{N} \log \text{Normal}_{\mu,\sigma^2}(x_i) =$$

$$\max_{(\mu,\sigma^2)} \sum_{i=1}^{N} \left[\frac{-(x_i-\mu)^2}{\sigma^2}\right] - N \log \sigma = F$$

Advanced topic

MLE Snowfall Example

Example: How much does it snow?

- $\mathcal{X} = \{x_1, x_2, \dots, x_N\}$ are snowfall values from the previous N storms
- Goal: learn ϕ such that g correctly models, as accurately as possible, the amount of snow likely
- Assumption: each x_i is independent from all others, but all from g

$$\max_{\phi} \sum_{i=1}^{N} \log g_{\phi}(x_i)$$

$$x_i \sim \text{Normal}(\mu, \sigma^2)$$

$$\max_{(\mu,\sigma^2)} \sum_{i=1}^{N} \log \text{Normal}_{\mu,\sigma^2}(x_i) =$$

$$\max_{(\mu,\sigma^2)} \sum_{i=1}^{N} \left[\frac{-(x_i - \mu)^2}{\sigma^2}\right] - N \log \sigma = F$$

Q: How do we find μ, σ^2?

Advanced topic

MLE Snowfall Example

Example: How much does it snow?

- $\mathcal{X} = \{x_1, x_2, \dots, x_N\}$ are snowfall values from the previous N storms
- Goal: learn ϕ such that g correctly models, as accurately as possible, the amount of snow likely
- Assumption: each x_i is independent from all others, but all from g

$$\max_{\phi} \sum_{i=1}^{N} \log g_{\phi}(x_i)$$

$$x_i \sim \text{Normal}(\mu, \sigma^2)$$

$$\max_{(\mu,\sigma^2)} \sum_{i=1}^{N} \log \text{Normal}_{\mu,\sigma^2}(x_i) =$$

$$\max_{(\mu,\sigma^2)} \sum_{i=1}^{N} \left[\frac{-(x_i - \mu)^2}{\sigma^2}\right] - N \log \sigma = F$$

Q: How do we find μ, σ^2?

A: Differentiate and find that

$$\hat{\mu} = \frac{\sum_i x_i}{N}$$

$$\sigma^2 = \frac{\sum_i (x_i - \hat{\mu})^2}{N}$$

Learning: Maximum Likelihood Estimation (MLE)

Central to machine learning:

- Observe some data $(\mathcal{X}, \mathcal{Y})$
- Compute some function $f(\mathcal{X})$ to {predict, explain, generate} $\mathcal{Y}$
- Assume f is controlled by parameters θ, i.e., $f_\theta(\mathcal{X})$
  - Sometimes written $f(\mathcal{X}; \theta)$

Learning:
Maximum Likelihood Estimation (MLE)

Central to machine learning:

- Observe some data $(\mathcal{X}, \mathcal{Y})$
- Compute some function $f(\mathcal{X})$ to {predict, explain, generate} $\mathcal{Y}$
- Assume f is controlled by parameters θ, i.e., $f_\theta(\mathcal{X})$
  - Sometimes written $f(\mathcal{X}; \theta)$
- Parameters are learned to $\underbrace{\text{minimize error (loss) } \ell}_{\text{Advanced topic}}$

Learning: Maximum Likelihood Estimation (MLE)

Example: Can I sleep in the next time it snows/is school canceled?

- $\mathcal{X} = \{x_1, x_2, \dots, x_N\}$ are snowfall values from the previous N storms
- $\mathcal{Y} = \{y_1, y_2, \dots, y_N\}$ are closure results from the previous N storms
- Goal: learn θ such that f correctly predicts, as accurately as possible, if UMBC will close in the next storm:
  - y^*_{n+1} from x_{n+1}

- If we assume the output of f is a *probability distribution* on $\mathcal{Y}|\mathcal{X}$…
  - $f(\mathcal{X}) \to \{p(\text{yes}|\mathcal{X}), p(\text{no}|\mathcal{X})\}$
- Then re: θ, {predicting, explaining, generating} $\mathcal{Y}$ means… *what*?

Model selection

Goal: Select the best network structure, given the data

Input:

- Training data
- Scoring function

Output:

- A network that maximizes the score

Structure selection: Scoring

- Bayesian: prior over parameters and structure
  - get balance between model complexity and fit to data as a byproduct

Marginal likelihood ***Prior***

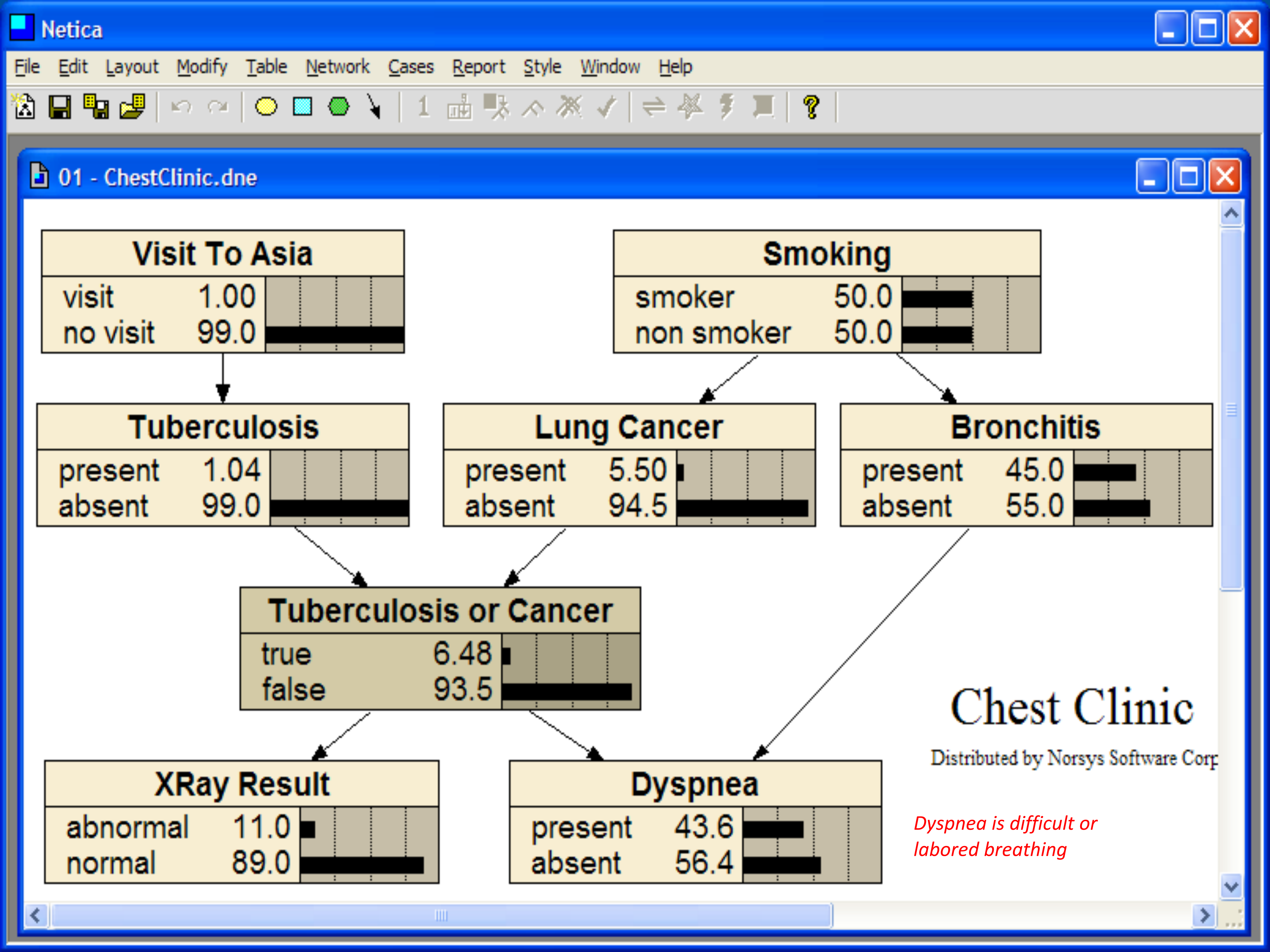
- Score (G:D) = log P(G|D) α log [P(D|G) P(G)]
- Marginal likelihood just comes from our parameter estimates
- Prior on structure can be any measure we want; typically a function of the network complexity

Same key property: Decomposability

$$\text{Score(structure)} = \sum_i \text{Score(family of } X_i)$$

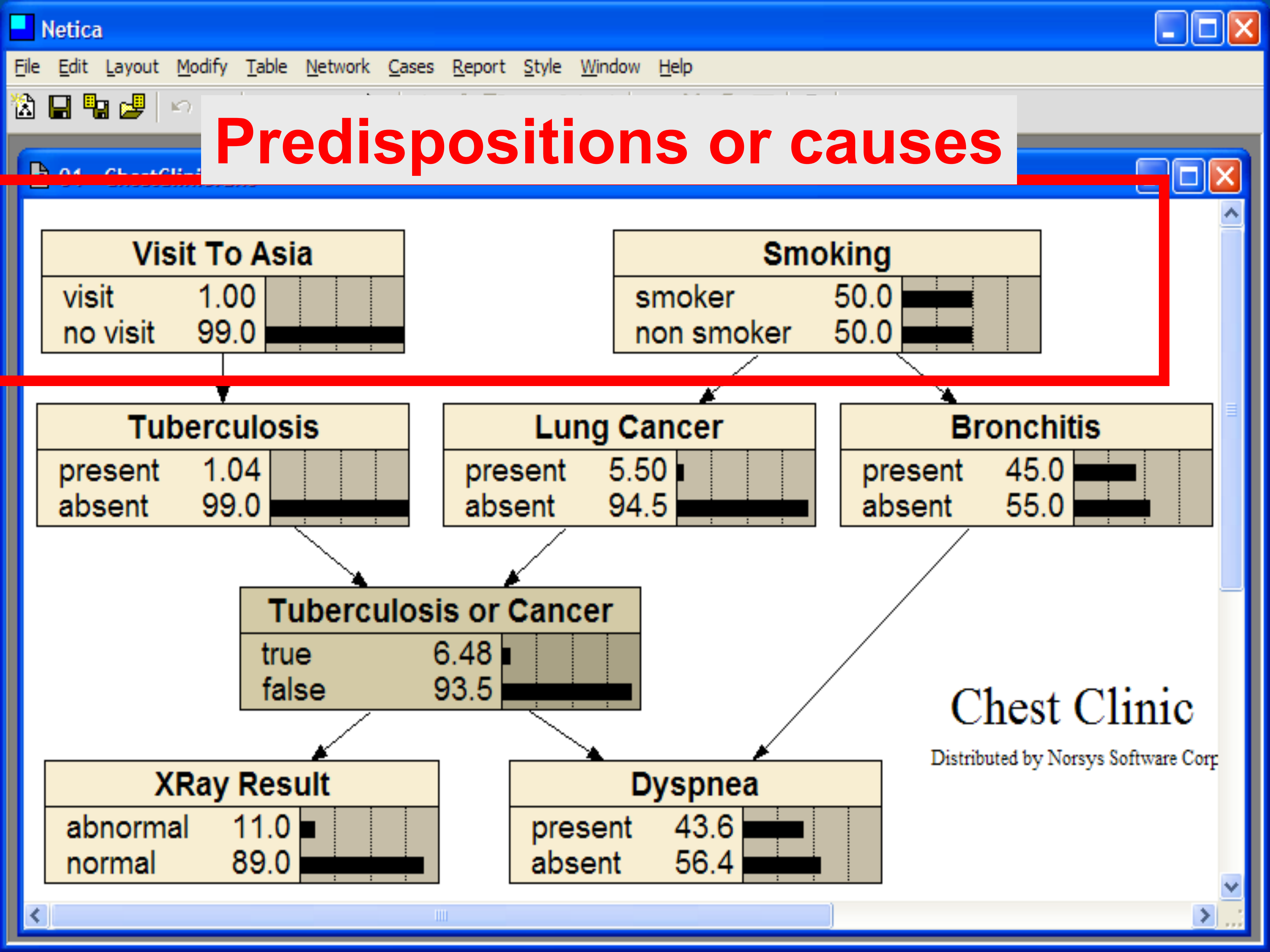
Some software tools

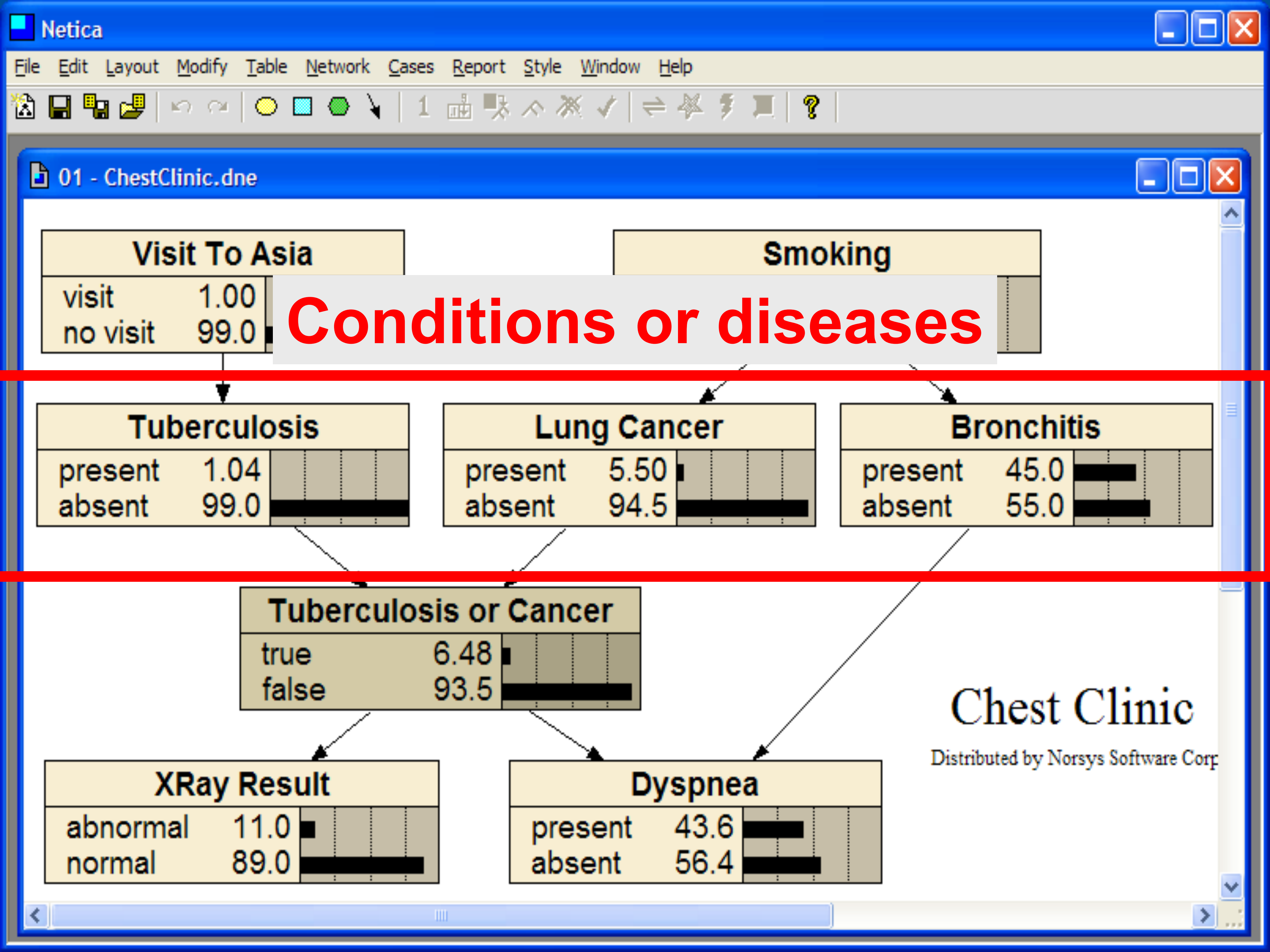
- Netica: Windows app for working with Bayes-ian belief networks and influence diagrams
  - Commercial product, free for small networks
  - Includes graphical editor, compiler, inference engine, etc.
  - To run in OS X or Linus you need Wire or Crossover
- Hugin: free demo versions for Linux, Mac, and Windows are available
- BBN.ipynb based on an AIMA notebook

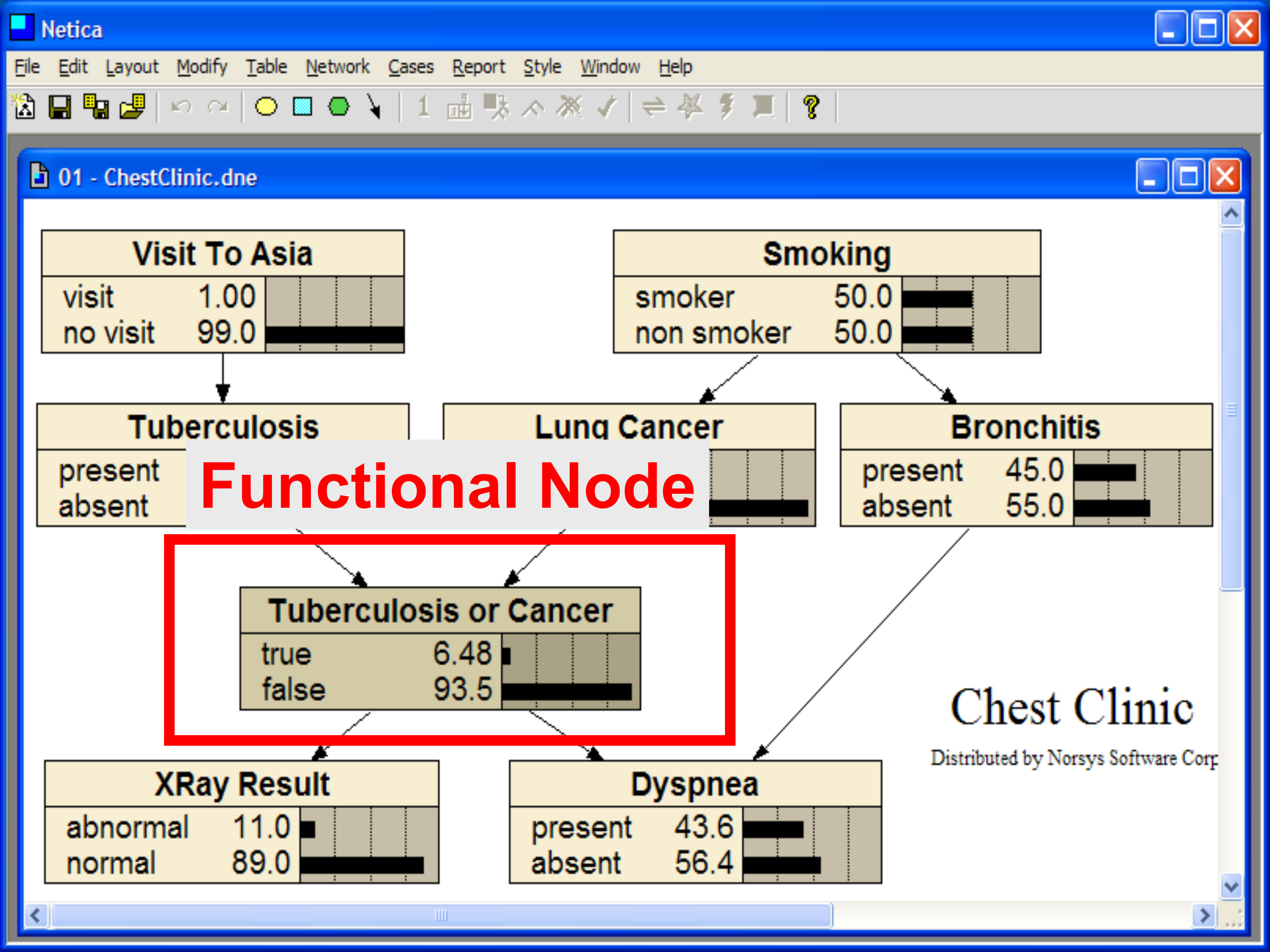


Dyspnea is difficult or labored breathing

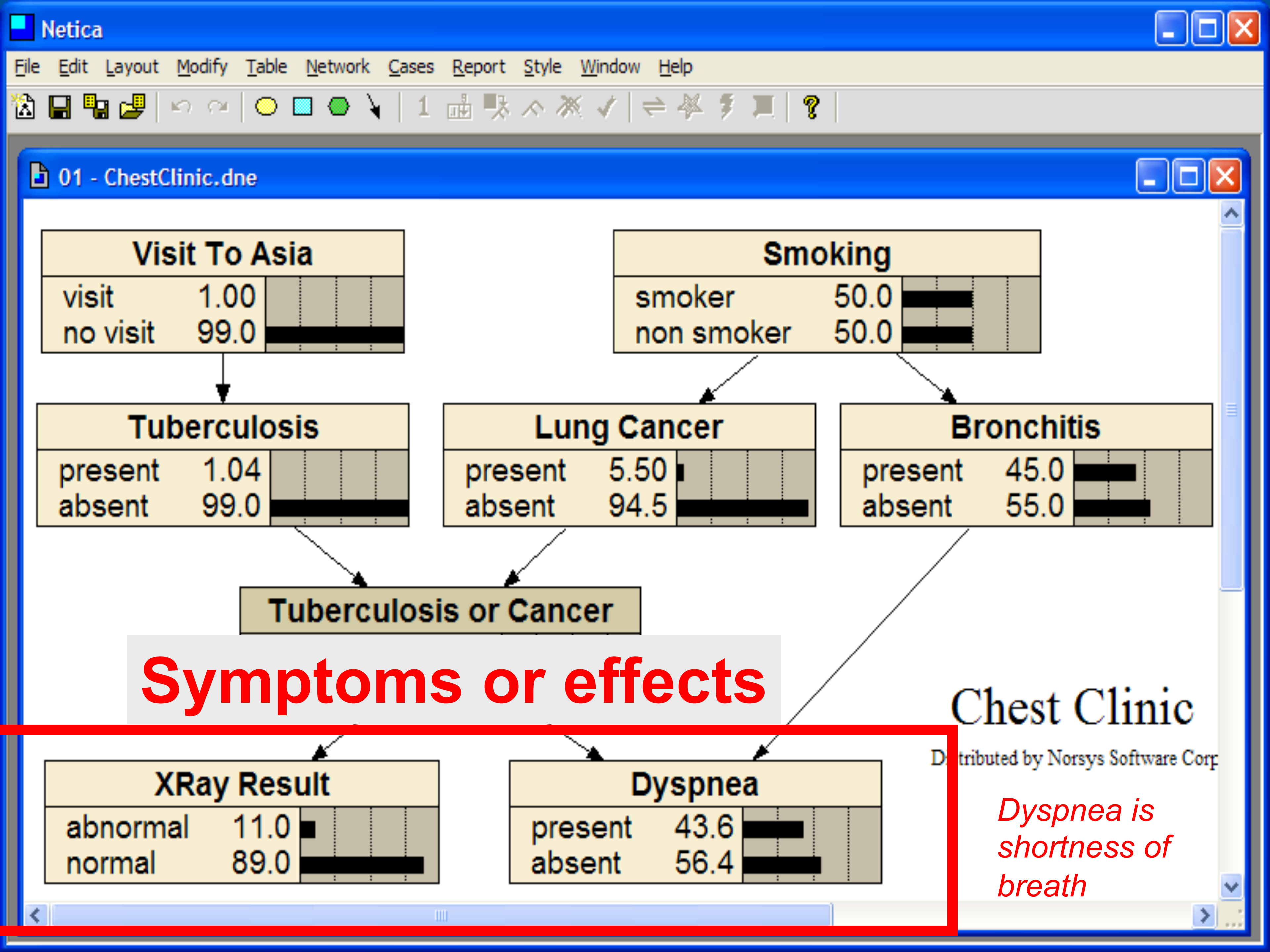
Predispositions or causes



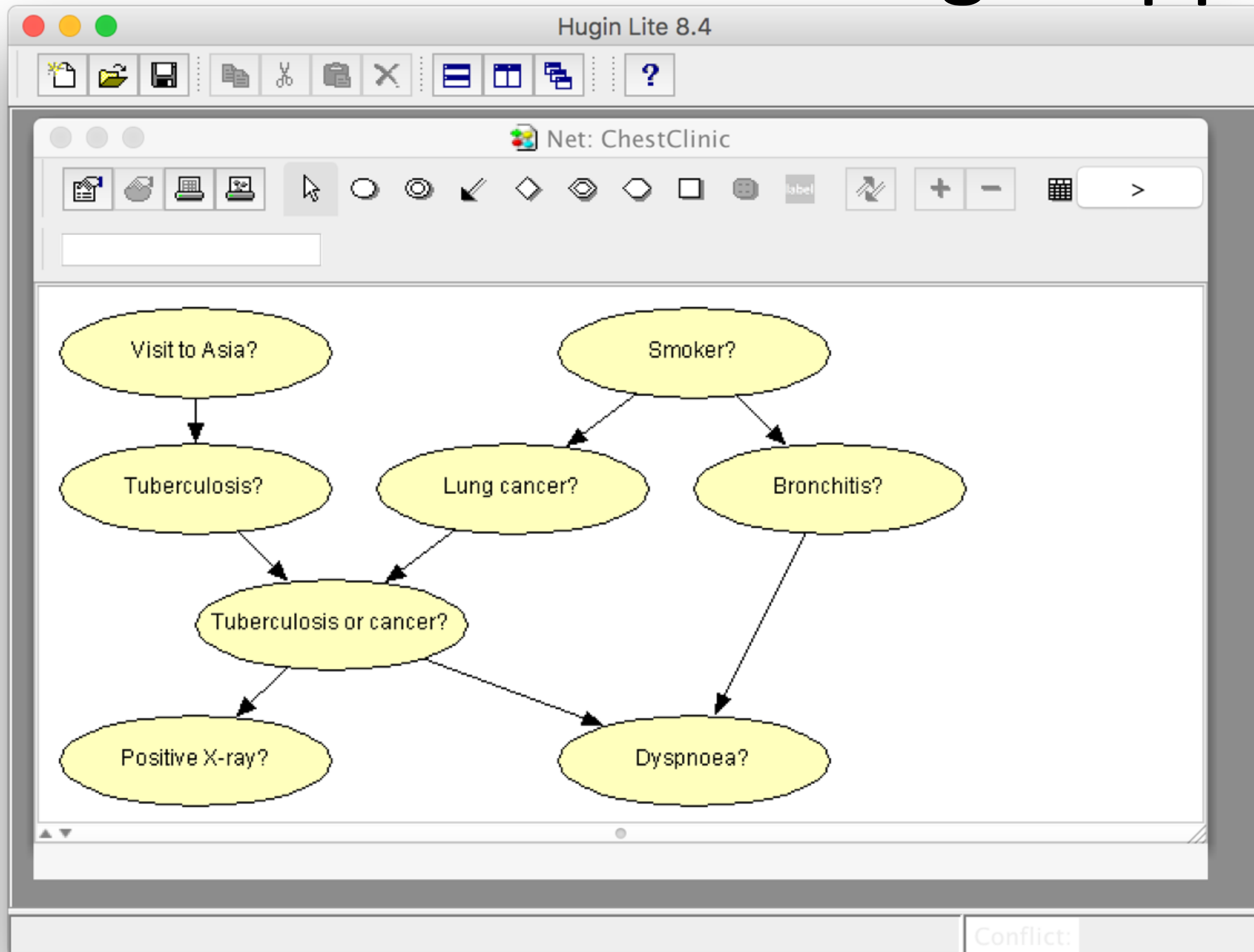

Netica
File Edit Layout Modify Table Network Cases Report Style Window Help
01 - ChestClinic.dne
Visit To Asia
visit 1.00
no visit 99.0
Smoking
Conditions or diseases
Tuberculosis
present 1.04
absent 99.0
Lung Cancer
present 5.50
absent 94.5
Bronchitis
present 45.0
absent 55.0
Tuberculosis or Cancer
true 6.48
false 93.5
XRay Result
abnormal 11.0
normal 89.0
Dyspnea
present 43.6
absent 56.4
Chest Clinic
Distributed by Norsys Software Corp



Netica
File Edit Layout Modify Table Network Cases Report Style Window Help
01 - ChestClinic.dne
Visit To Asia
visit 1.00
no visit 99.0
Smoking
smoker 50.0
non smoker 50.0
Tuberculosis
present
absent
Lung Cancer
Bronchitis
present 45.0
absent 55.0
Functional Node
Tuberculosis or Cancer
true 6.48
false 93.5
Chest Clinic
Distributed by Norsys Software Corp
XRay Result
abnormal 11.0
normal 89.0
Dyspnea
present 43.6
absent 56.4


Netica
File Edit Layout Modify Table Network Cases Report Style Window Help
01 - ChestClinic.dne
Visit To Asia
visit 1.00
no visit 99.0
Smoking
smoker 50.0
non smoker 50.0
Tuberculosis
present 1.04
absent 99.0
Lung Cancer
present 5.50
absent 94.5
Bronchitis
present 45.0
absent 55.0
Tuberculosis or Cancer
Symptoms or effects
Chest Clinic
Distributed by Norsys Software Corp
XRay Result
abnormal 11.0
normal 89.0
Dyspnea
present 43.6
absent 56.4
Dyspnea is shortness of breath

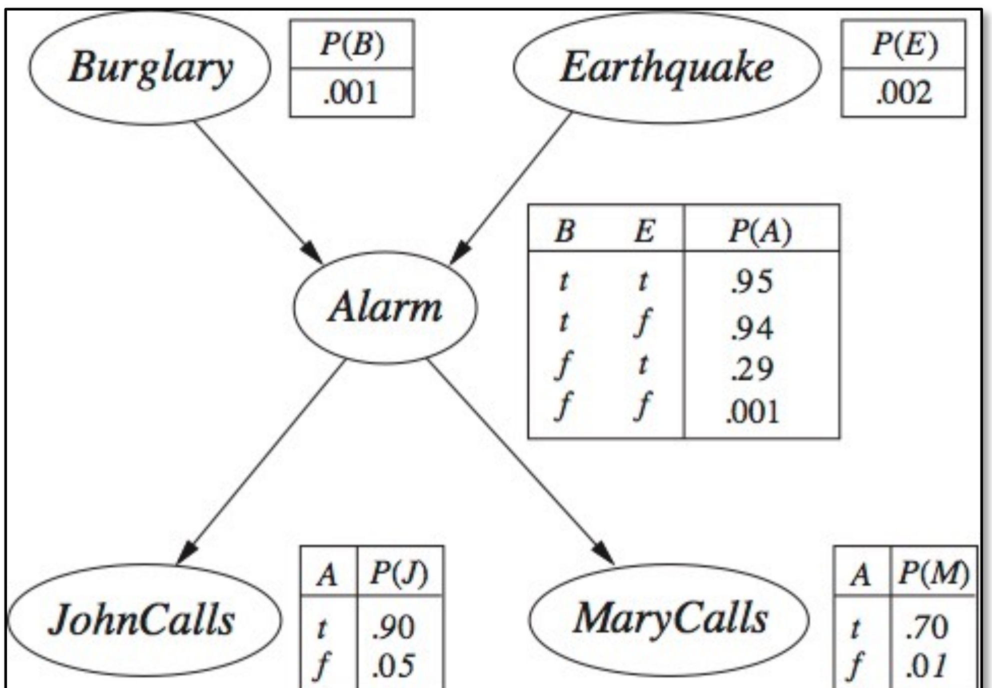
Same BBN model in Hugin app



See the 4-minute [HUGIN Tutorial]on YouTube

Python Code

See this AIMA notebook on colab showing how to construct this BBN Network in Python



Judea Pearl example

There's is a house with a burglar alarm that can be triggered by a burglary or earthquake. If it sounds, one or both neighbors John & Mary, might call the owner to say the alarm is sounding.