CMSC 471:
Reasoning with Bayesian Belief Network

Chapters 12 & 13

KMA Solaiman – ksolaima@umbc.edu

Some slides courtesy Tim Finin and Frank Ferraro

Overview

- Bayesian Belief Networks (BBNs) can reason with networks of propositions and associated probabilities
- Useful for many AI problems
  - Diagnosis
  - Expert systems
  - Planning
  - Learning

Probabilistic Graphical Models

A graph G that represents a probability distribution over N random variables $X_1, \ldots, X_N$

Probabilistic Graphical Models

A graph G that represents a probability distribution over N random variables $X_1, \dots, X_N$

Graph G = (vertices V, edges E)

Distribution $P(X_1, \dots, X_N)$

Probabilistic Graphical Models

A graph G that represents a probability distribution over random N variables $X_1, \ldots, X_N$

Graph G = (vertices V, edges E)

Distribution $P(X_1, \ldots, X_N)$

Vertices $\leftrightarrow$ random variables

Edges show dependencies among random variables

Probabilistic Graphical Models

A graph G that represents a probability distribution over N random variables $X_1, \ldots, X_N$

Graph G = (vertices V, edges E)
Distribution $p(X_1, \ldots, X_N)$

Vertices ↔ random variables
Edges show dependencies among random variables

Two main flavors: *directed* graphical models and *undirected* graphical models

Probabilistic Graphical Models

A graph G that represents a probability distribution over N random variables $X_1, \ldots, X_N$

Graph G = (vertices V, edges E)
Distribution $p(X_1, \ldots, X_N)$

Vertices ↔ random variables
Edges show dependencies among random variables

Two main flavors: ***directed*** **graphical models** and *undirected* graphical models

Directed Graphical Models

A *directed* (acyclic) graph G=(V,E) that represents a probability distribution over random variables $X_1, \ldots, X_N$

Joint probability factorizes into factors of X_i conditioned on the parents of X_i

Directed Graphical Models

A *directed* (acyclic) graph G=(V,E) that represents a probability distribution over random variables $X_1, \ldots, X_N$

Joint probability factorizes into factors of X_i conditioned on the parents of X_i

Benefit: the independence properties are *transparent*

Directed Graphical Models

A *directed* (acyclic) graph G=(V,E) that represents a probability distribution over random variables $X_1, \ldots, X_N$

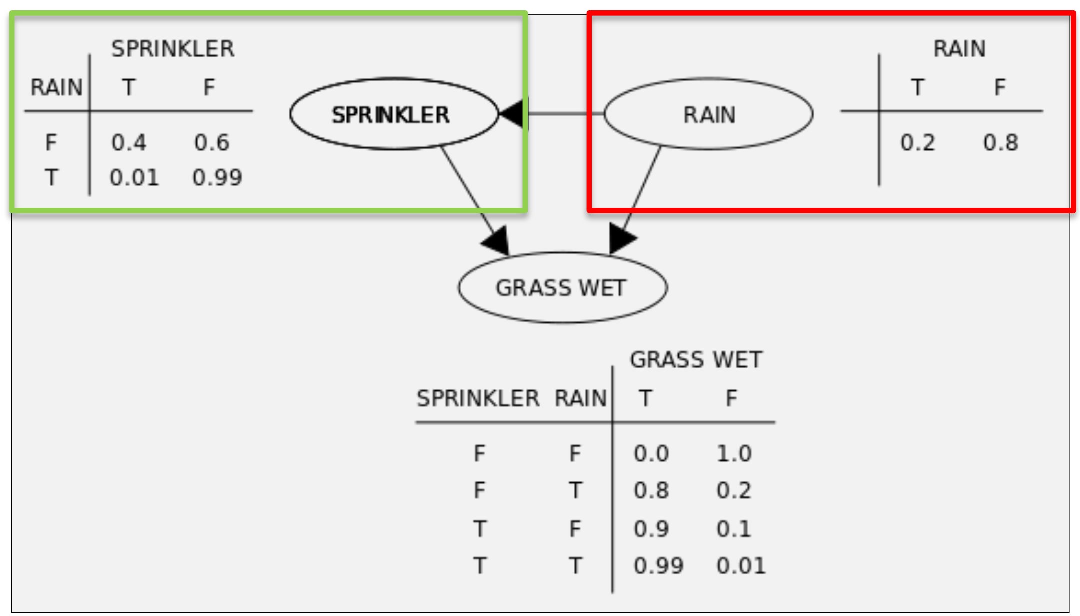
Joint probability factorizes into factors of X_i conditioned on the parents of X_i

A graph/joint distribution that follows this is a **Bayesian network**

BBN Definition

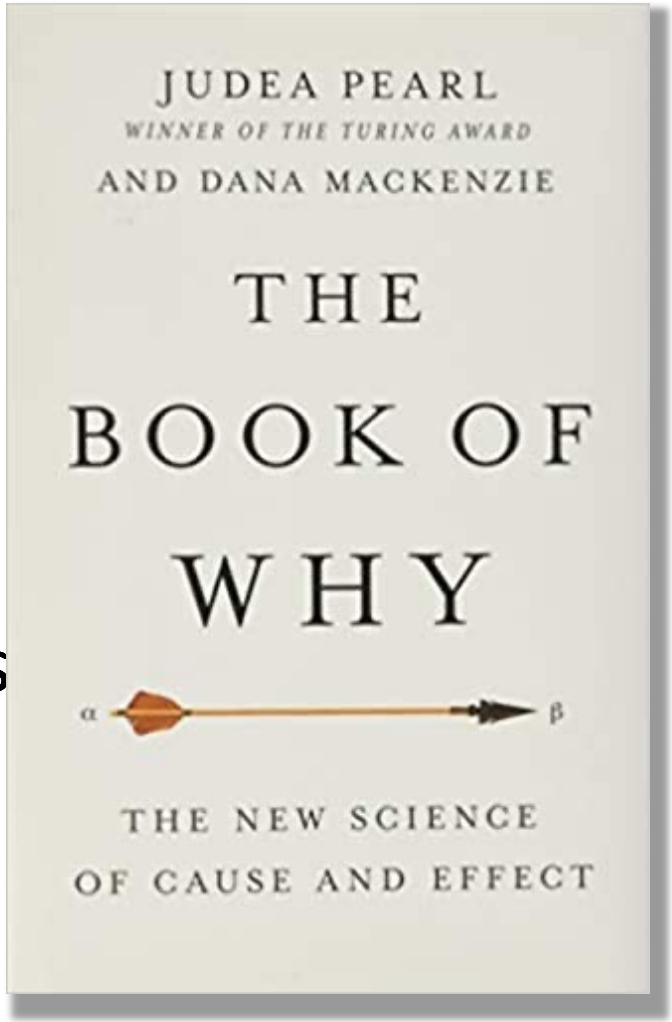
- AKA Bayesian Network, Bayes Net
- A graphical model (as a DAG) of probabilistic relationships among a set of random variables
- Nodes are variables, links represent direct influence of one variable on another
- Nodes have **prior probabilities** or **conditional probability tables** (CPTs)

source



History lesson: Judea Pearl

- UCLA CS professor
- Introduced Bayesian networks in the 1980s
- Pioneer of probabilistic approach to AI reasoning
- First to formalize causal modeling in empirical sciences
- Written many books on the topics, including the popular 2018 Book of Why



Why? 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

Recall Bayes Rule

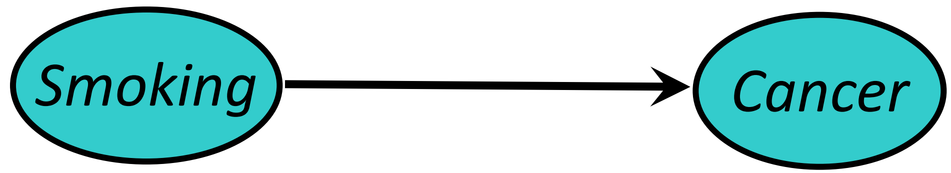
$$P(H,E) = P(H \mid E)P(E) = P(E \mid H)P(H)$$

$$P(H|E) = \frac{P(E|H) * P(H)}{P(E)} \qquad P(E|H) = \frac{P(H|E) * P(E)}{P(H)}$$

Note symmetry: we can compute probability of a ***hypothesis given its evidence*** as well as probability of ***evidence given hypothesis***

Simple Bayesian Network

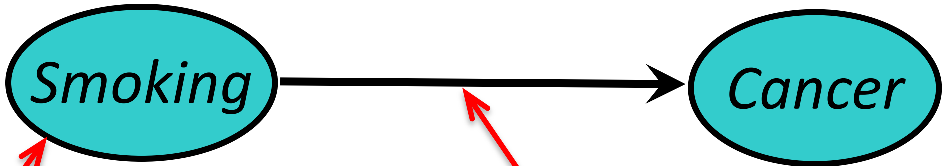
$S \in \{no, light, heavy\}$



$C \in \{none, benign, malignant\}$

Simple Bayesian Network

$S \in \{no, light, heavy\}$



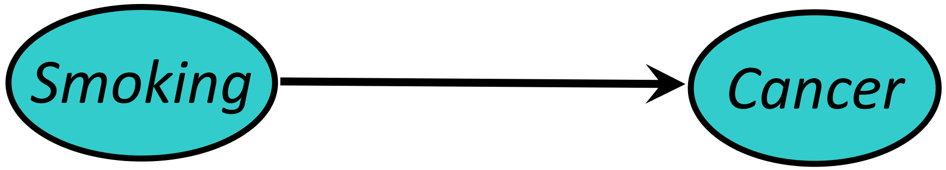
$C \in \{none, benign, malignant\}$

Nodes represent variables

Links represent *"causal"* relations

Simple Bayesian Network

$S \in \{no, light, heavy\}$



$C \in \{none, benign, malignant\}$

Prior probability of S

P(S=no)	0.80
P(S=light)	0.15
P(S=heavy)	0.05

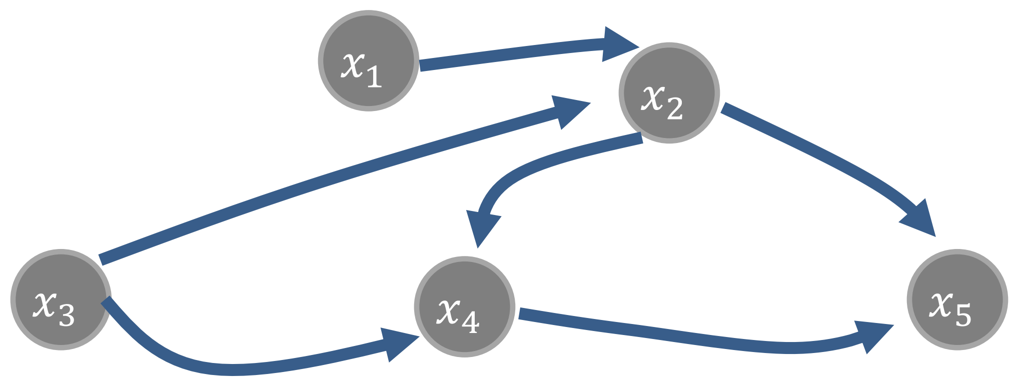
Nodes with no in-links have **prior probabilities**

Conditional distribution of S and C

Smoking=	*no*	*light*	*heavy*
C=none	0.96	0.88	0.60
C=benign	0.03	0.08	0.25
C=malignant	0.01	0.04	0.15

Nodes with in-links have **joint probability distributions**

Bayesian Networks: Directed Acyclic Graphs

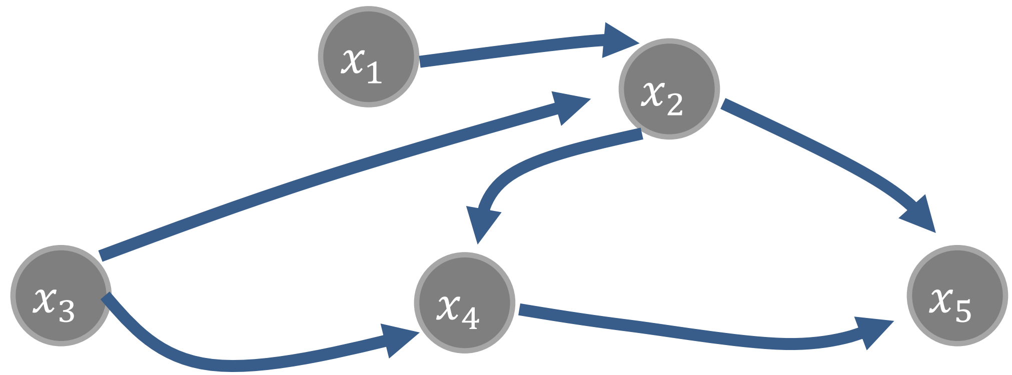


$$p(x_1, x_2, x_3, \ldots, x_N) = \prod_i p(x_i \mid \pi(x_i))$$

topological sort

"parents of"

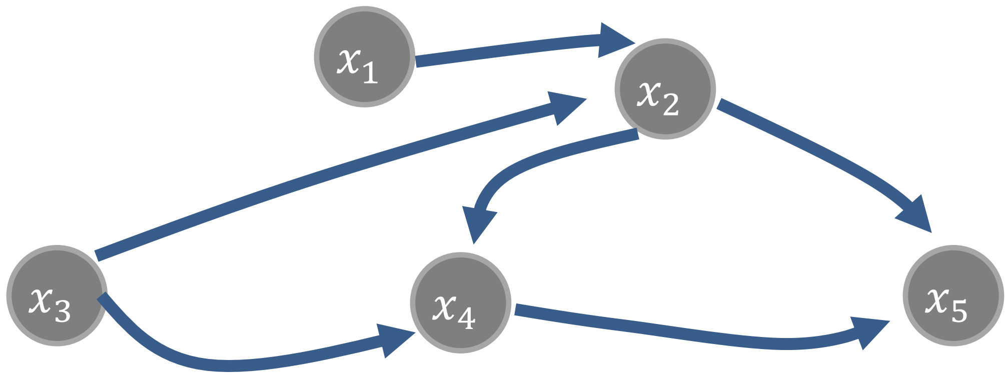
Bayesian Networks: Directed Acyclic Graphs



$$p(x_1, x_2, x_3, \ldots, x_N) = \prod_i p(x_i \mid \pi(x_i))$$

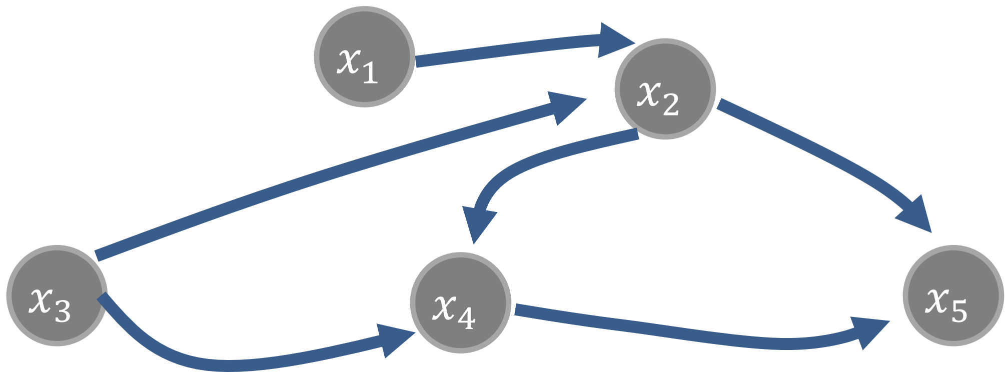
$p(x_1, x_2, x_3, x_4, x_5)$= ???

Bayesian Networks: Directed Acyclic Graphs



$$p(x_1, x_2, x_3, x_4, x_5) =$$
$$p(x_1)p(x_3)p(x_2|x_1, x_3)p(x_4|x_2, x_3)p(x_5|x_2, x_4)$$

Bayesian Networks: Directed Acyclic Graphs

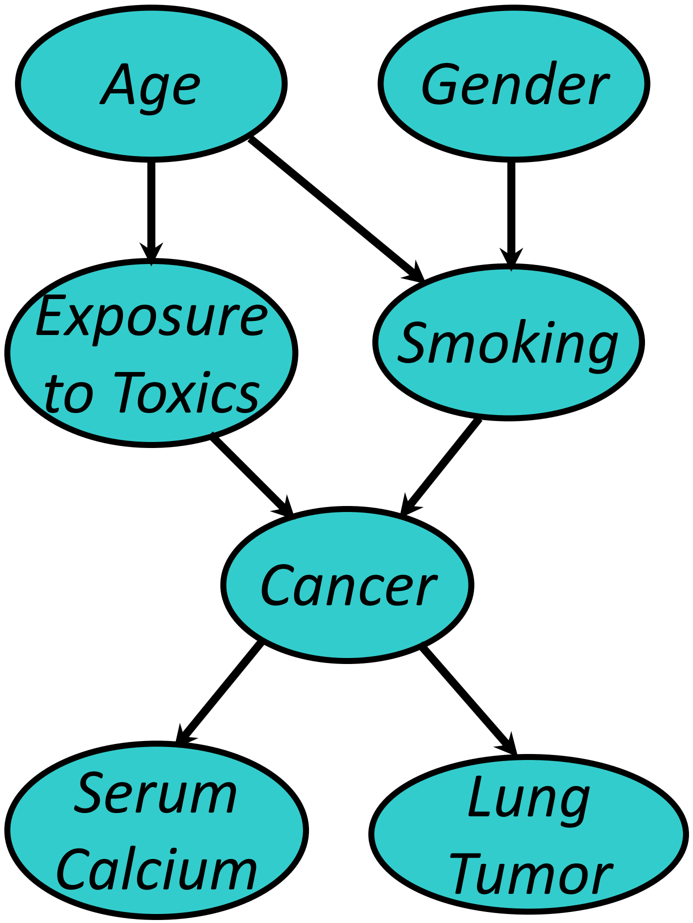


$$p(x_1, x_2, x_3, \ldots, x_N) = \prod_i p(x_i \mid \pi(x_i))$$

exact inference in general DAGs is NP-hard

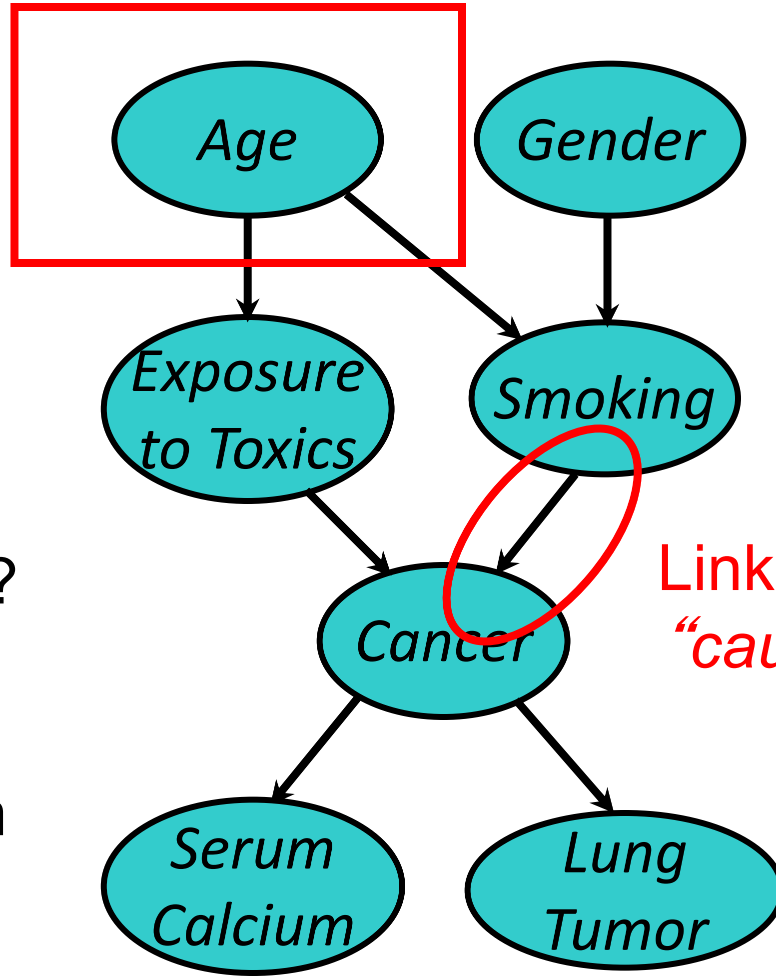
inference in trees can be exact

More Complex Bayesian Network



More Complex Bayesian Network

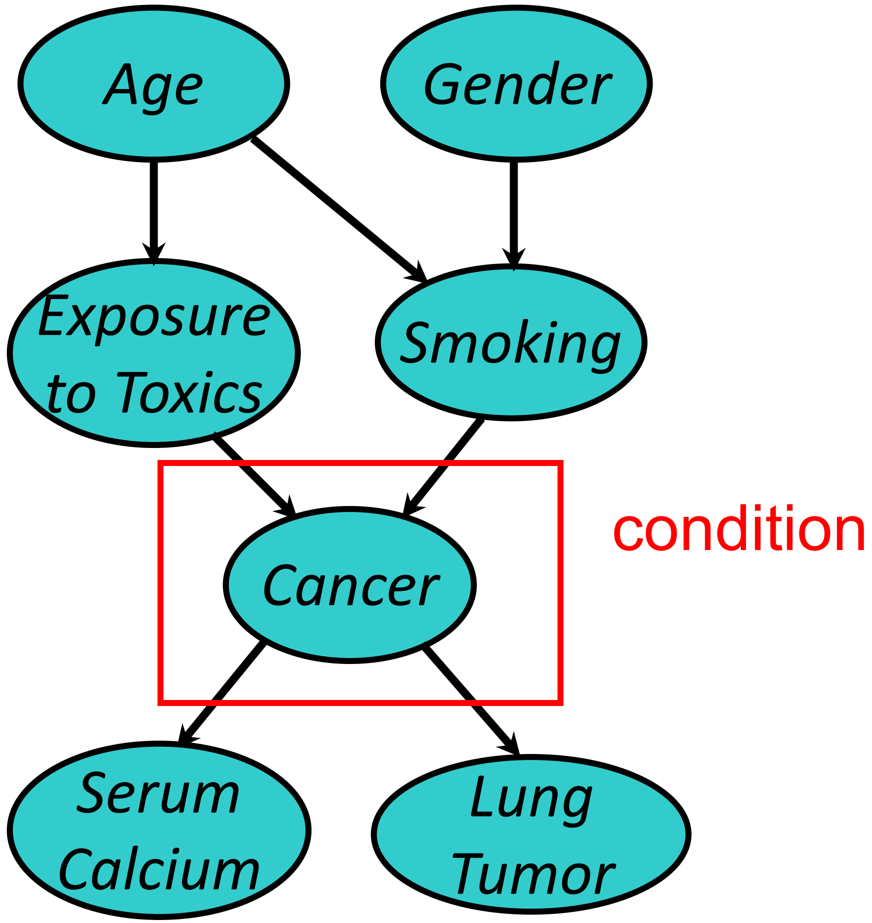
Nodes represent variables



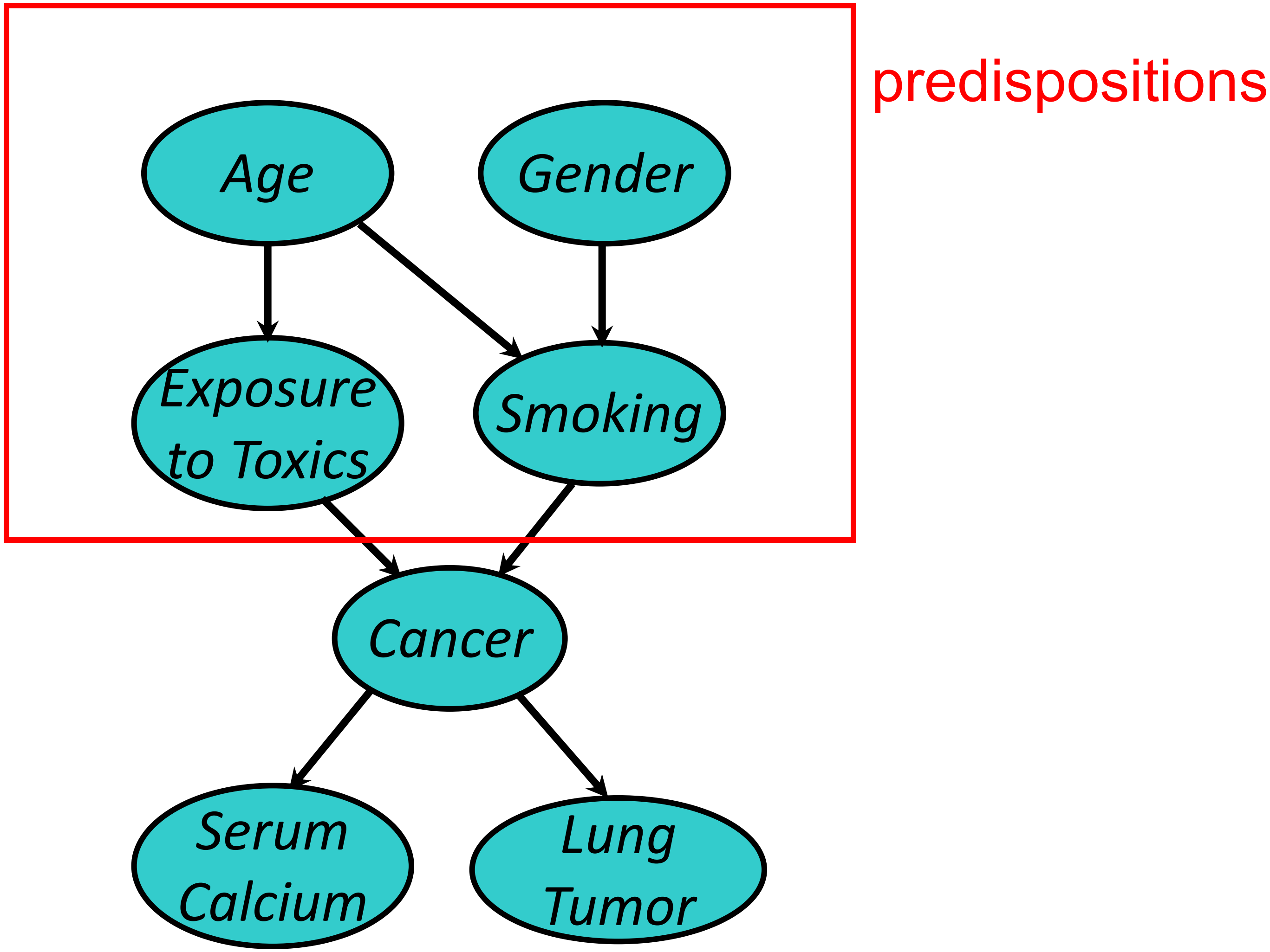
Links represent *"causal"* relations

- Does gender cause smoking?
- Influence might be a better term

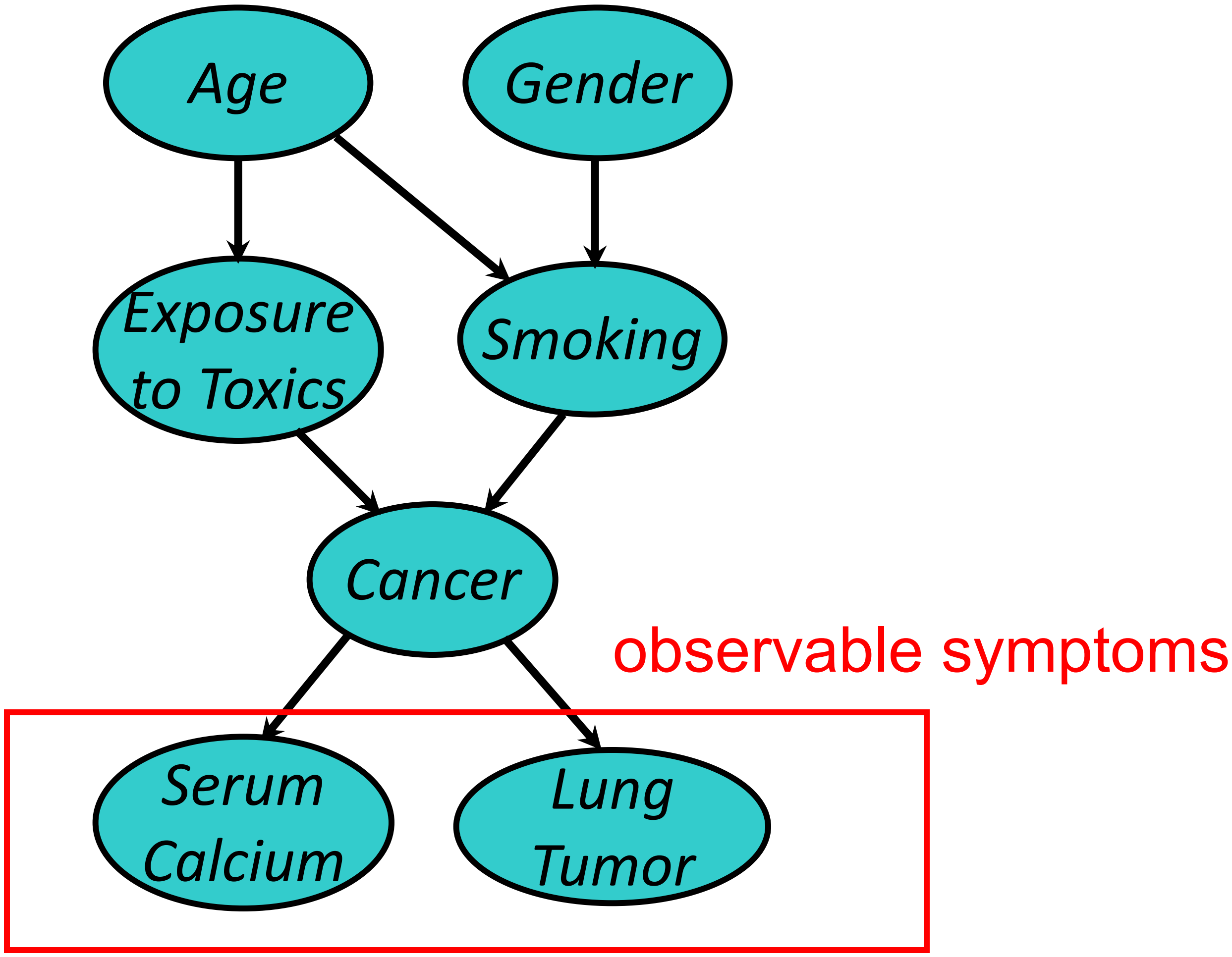
More Complex Bayesian Network



More Complex Bayesian Network

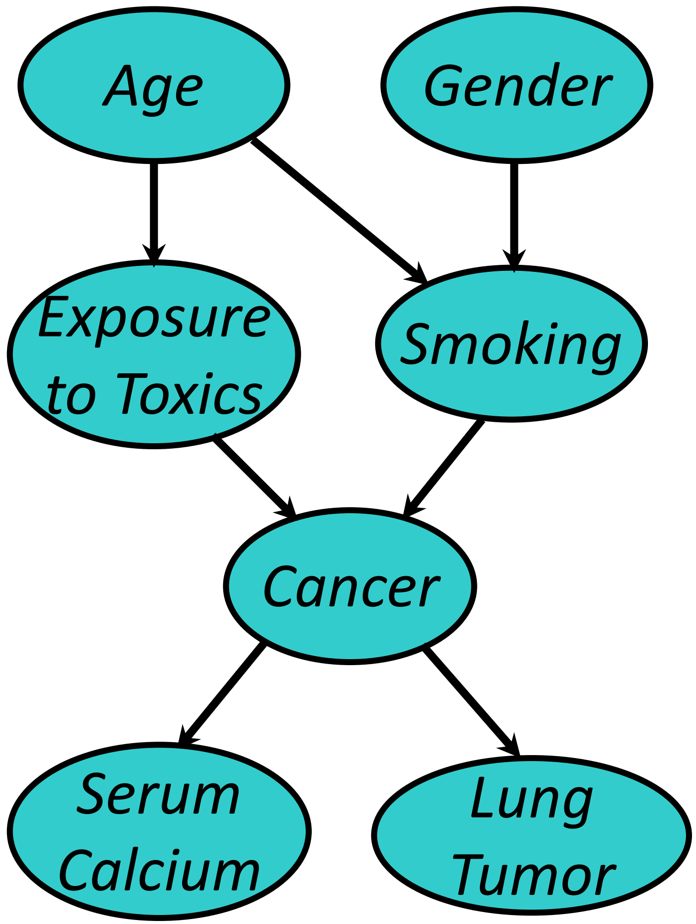


More Complex Bayesian Network



More Complex Bayesian Network

Can we predict likelihood of **lung tumor** given values of other 6 variables?



- Model has 7 variables
- Complete joint probability distribution will have 7 dimensions!
- Too much data required ☹
- BBN simplifies: a node has a CPT with data on itself & parents in graph

CPT = [conditional probability table]

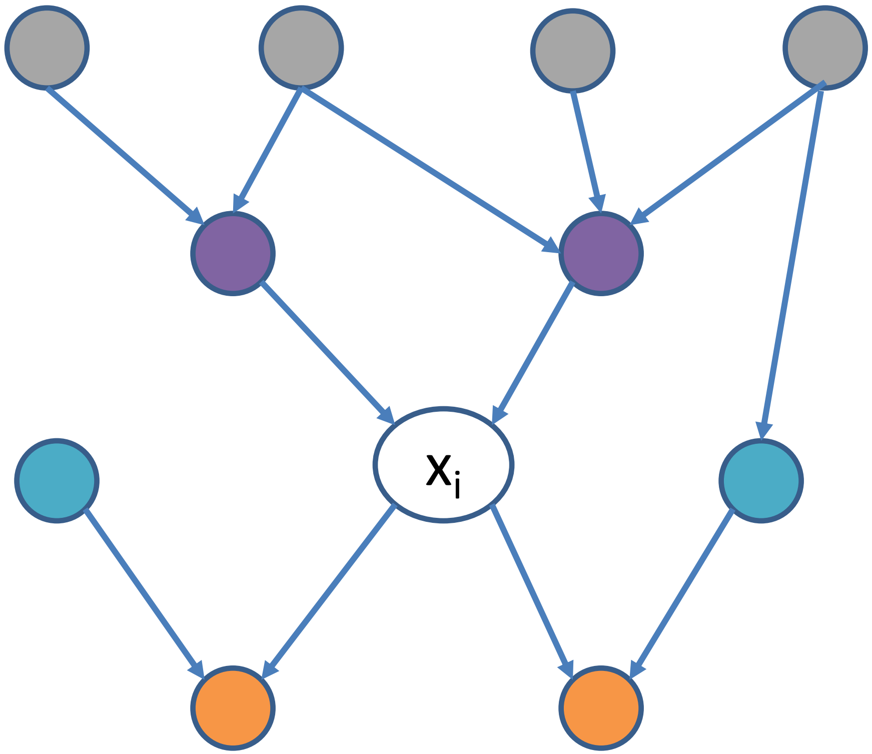
Independence & Conditional Independence in BBNs

Read these independence relationships right from the graph!

There are two common concepts that can help:

1. Markov blanket
2. D-separation (not covering)

Markov Blanket

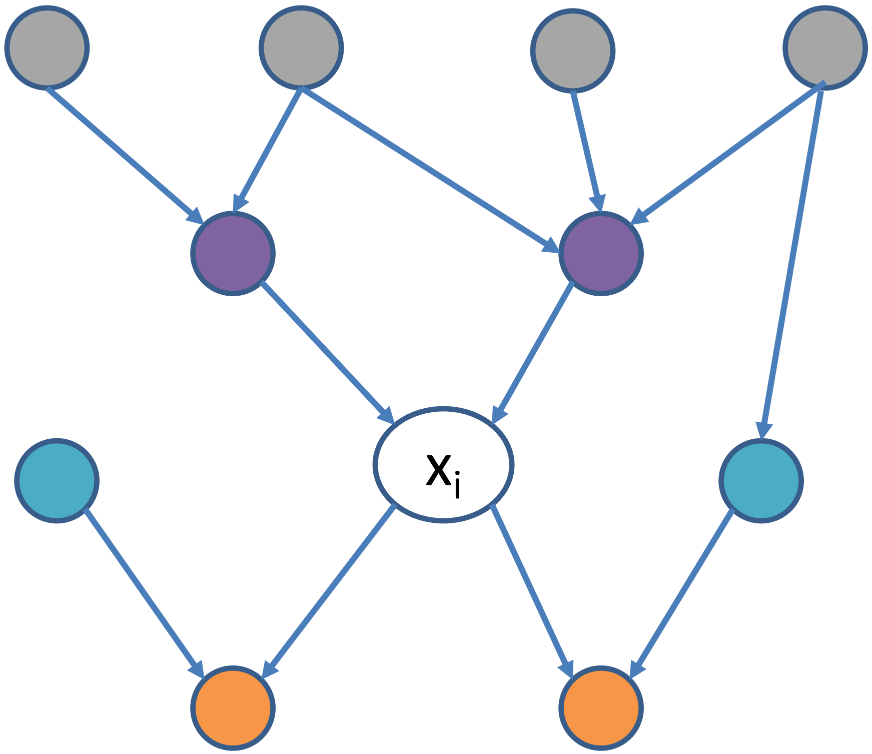


Markov blanket of a node x is its parents, children, and children's parents

(in this example, shading does not show observed/latent)

The **Markov Blanket** of a node x_i the set of nodes needed to form the complete conditional for a variable x_i

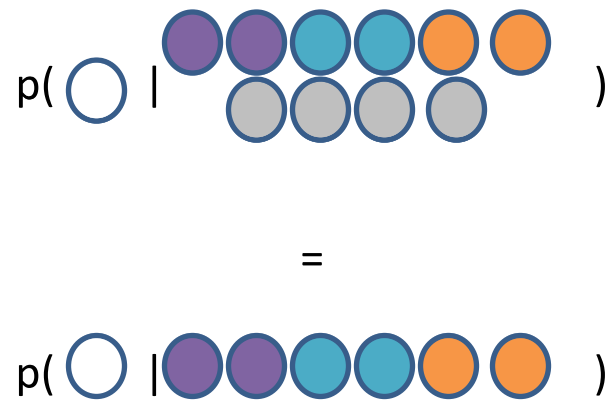
Markov Blanket



Markov blanket of a node x is its parents, children, and children's parents

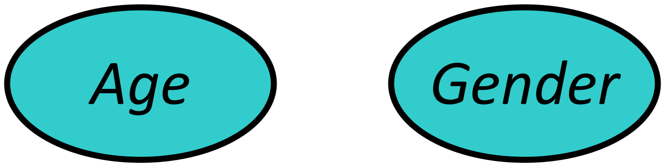
(in this example, shading does not show observed/latent)

The **Markov Blanket** of a node x_i the set of nodes needed to form the complete conditional for a variable x_i



Given its Markov blanket, a node is conditionally independent of all other nodes in the BN

Independence



Age and *Gender* are independent*.

$$P(A,G) = P(G) * P(A)$$

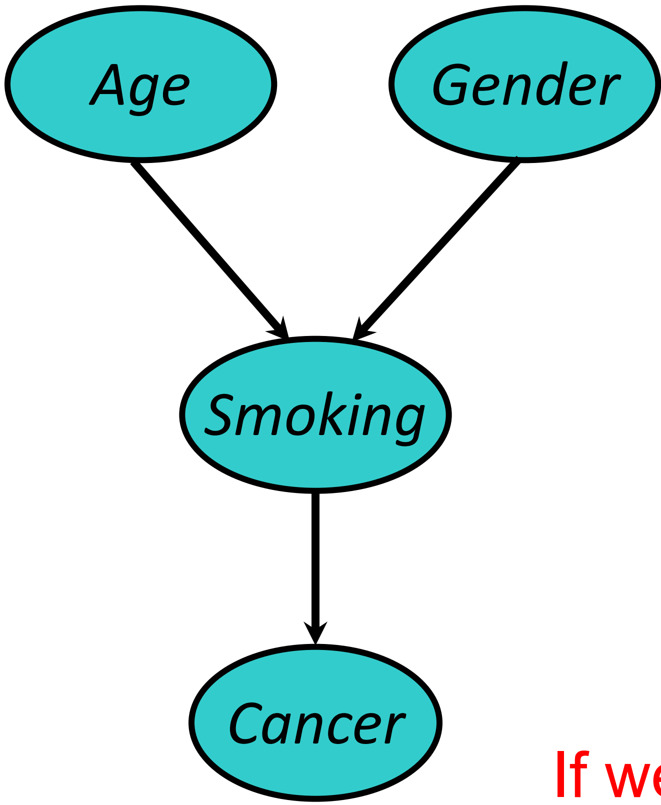
There is no path between them in the graph

$$P(A \mid G) = P(A)$$
$$P(G \mid A) = P(G)$$

$$P(A,G) = P(G \mid A)\, P(A) = P(G)P(A)$$
$$P(A,G) = P(A \mid G)\, P(G) = P(A)P(G)$$

* Not strictly true, but a reasonable approximation

Conditional Independence

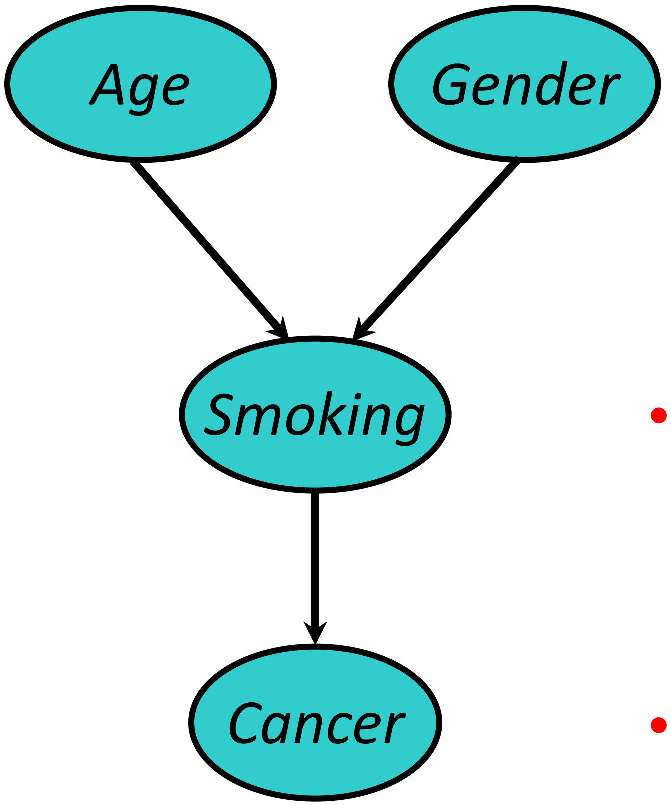


Cancer is independent of *Age* and *Gender* given *Smoking*

$$P(C \mid A,G,S) = P(C \mid S)$$

If we know value of smoking, no need to know values of age or gender

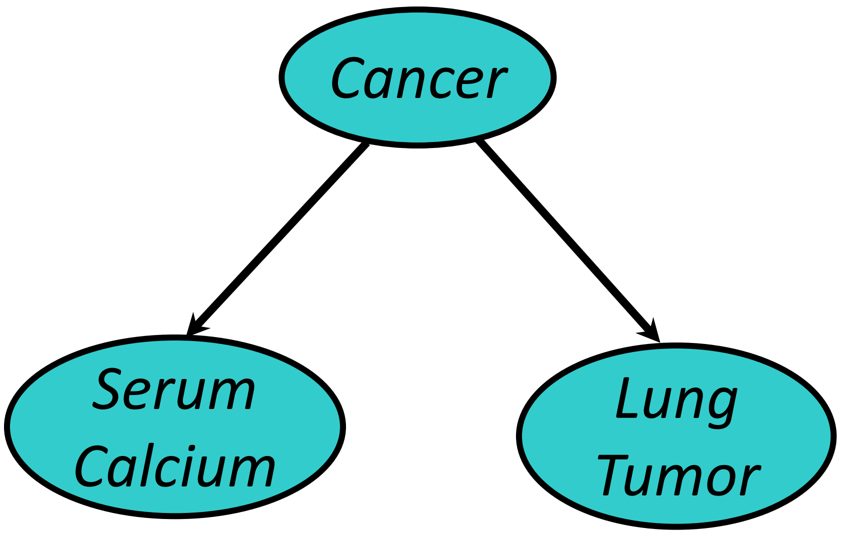
Conditional Independence



Cancer is independent of *Age* and *Gender* given *Smoking*

- Instead of one big CPT with 4 variables, we have two smaller CPTs with 3 and 2 variables
- If all variables binary: 12 models ($2^3 + 2^2$) rather than 16 (2^4)

Conditional Independence: Naïve Bayes



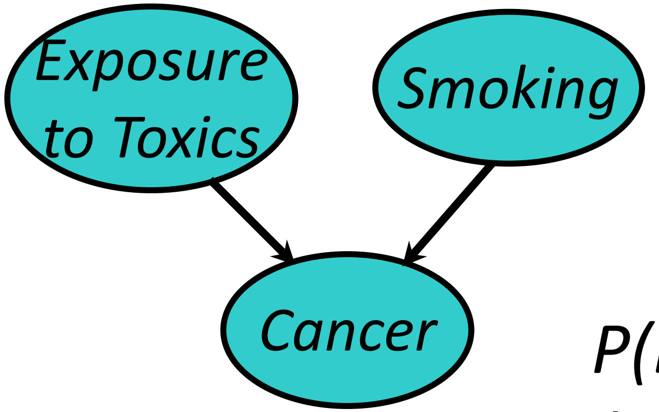
Serum Calcium and *Lung Tumor* are dependent

Serum Calcium is independent of *Lung Tumor*, given *Cancer*

$$P(L \mid SC, C) = P(L \mid C)$$
$$P(SC \mid L, C) = P(SC \mid C)$$

Naïve Bayes assumption: evidence (e.g., symptoms) independent given disease; easy to combine evidence

Explaining Away



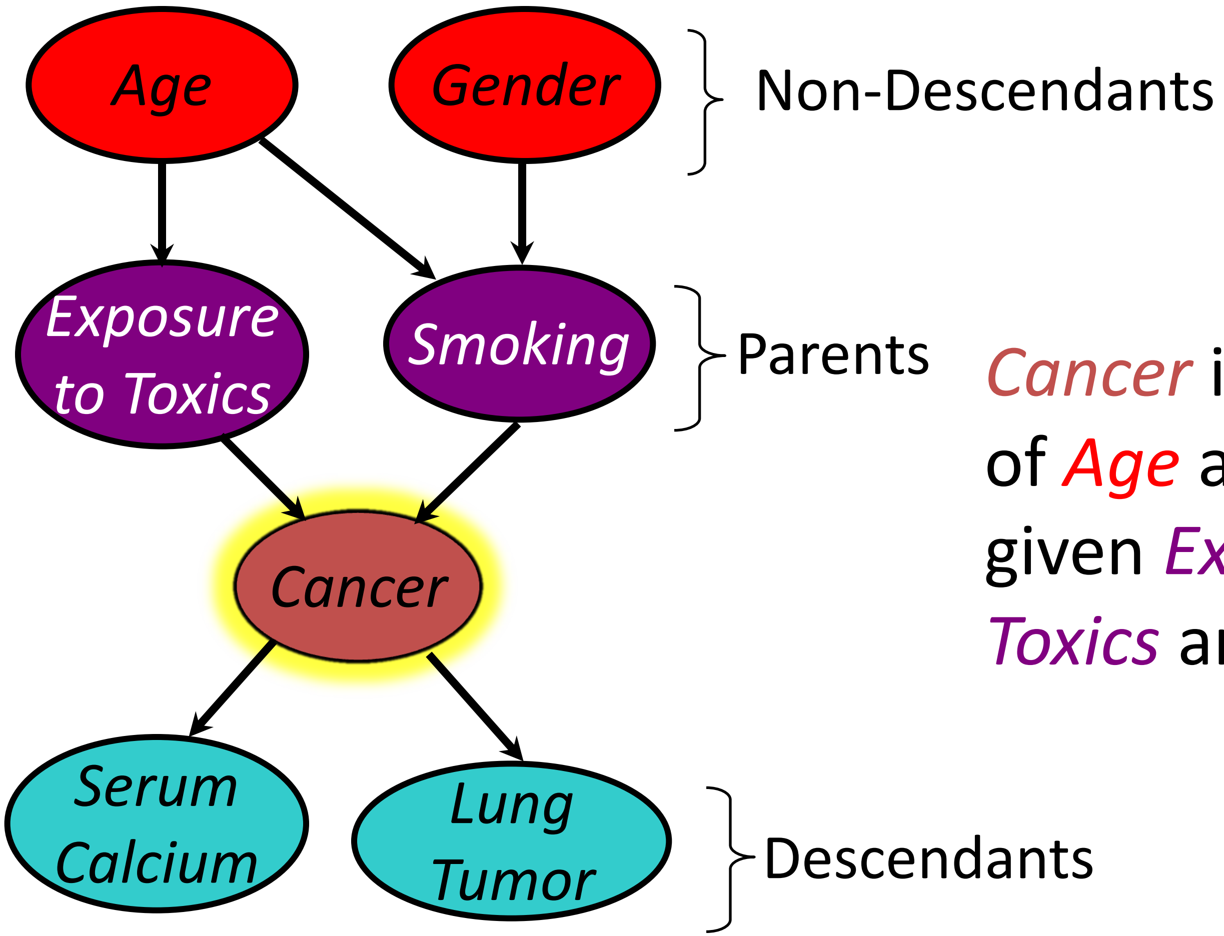
Exposure to Toxics and *Smoking* are independent

Exposure to Toxics is **dependent** on *Smoking*, given *Cancer*

P(E=heavy | C=malignant) > P(E=heavy | C=malignant, S=heavy)

- *Explaining away:* reasoning pattern where confirmation of one cause reduces need to invoke alternatives
- Essence of [Occam's Razor]() (prefer hypothesis with fewest assumptions)
- Relies on independence of causes

Conditional Independence



Cancer is independent of *Age* and *Gender* given *Exposure to Toxics* and *Smoking*.

BBN Construction

The **knowledge acquisition** process for a BBN involves three steps

KA1: Choosing appropriate variables

KA2: Deciding on the network structure

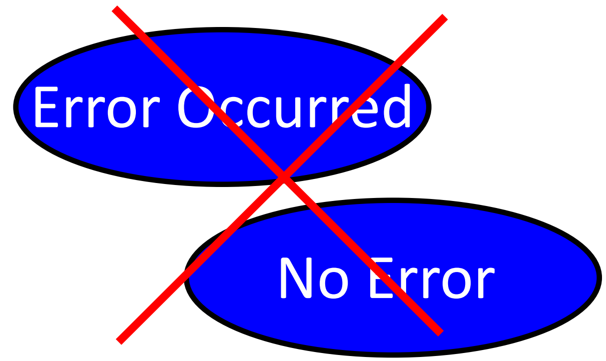
KA3: Obtaining data for the conditional probability tables

KA1: Choosing variables

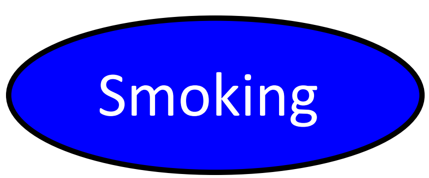
- Variable values: integers, reals or enumerations
- Variable should have collectively *exhaustive*, *mutually exclusive* values

$$x_1 \vee x_2 \vee x_3 \vee x_4$$

$$\neg(x_i \wedge x_j) \quad i \neq j$$



- They should be values, not probabilities

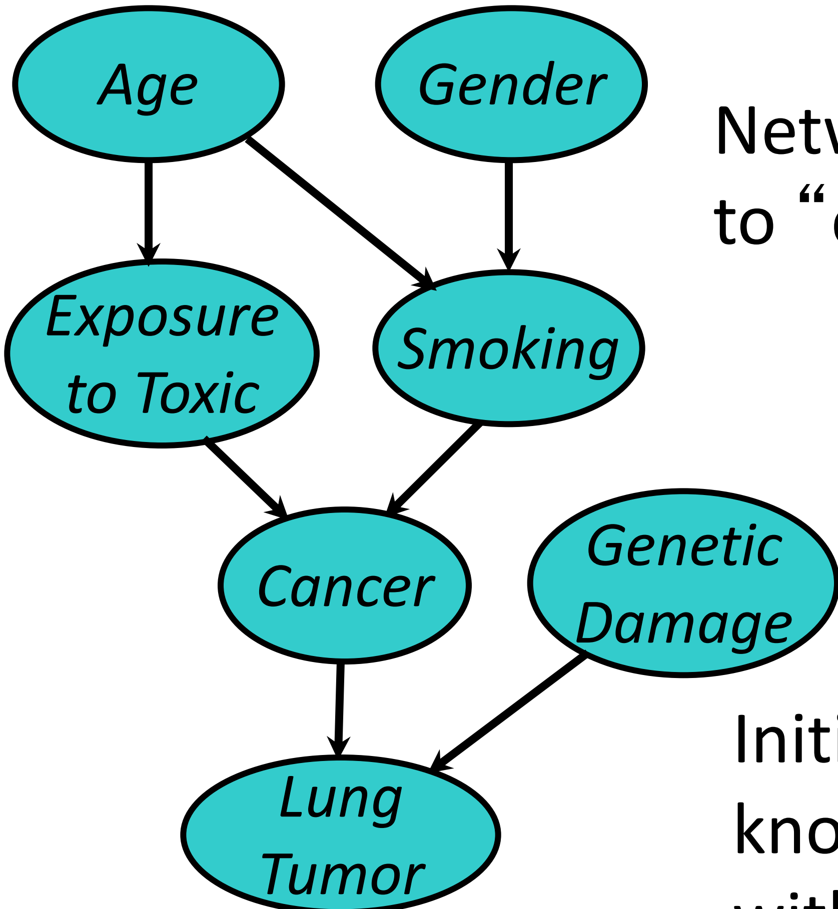


Heuristic: Knowable in Principle

Example of good variables

- Weather: {Sunny, Cloudy, Rain, Snow}
- Gasoline: Cents per gallon {0,1,2…}
- Temperature: { $\geq 100^\circ$ F , $< 100^\circ$ F}
- User needs help on Excel Charts: {Yes, No}
- User's personality: {dominant, submissive}

KA2: Structuring



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

Fundamental Inference & Learning Question

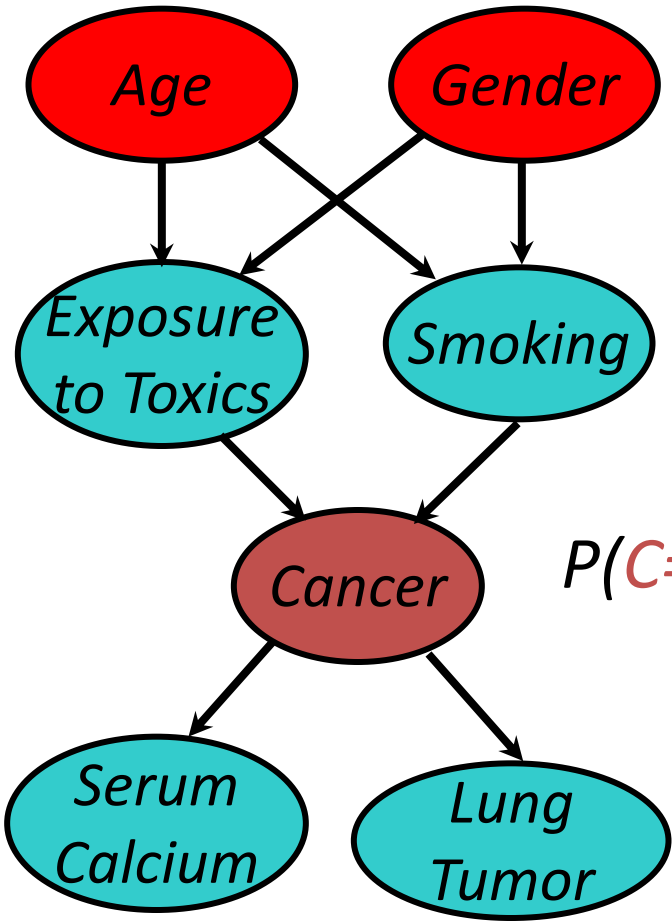
- Compute posterior probability of a node given some other nodes

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

- Some techniques
  - MLE (maximum likelihood estimation)/MAP (maximum a posteriori)
  - Variable Elimination
  - (Loopy) Belief Propagation ((Loopy) BP)
  - Monte Carlo
  - Variational methods
  - …

Advanced topics

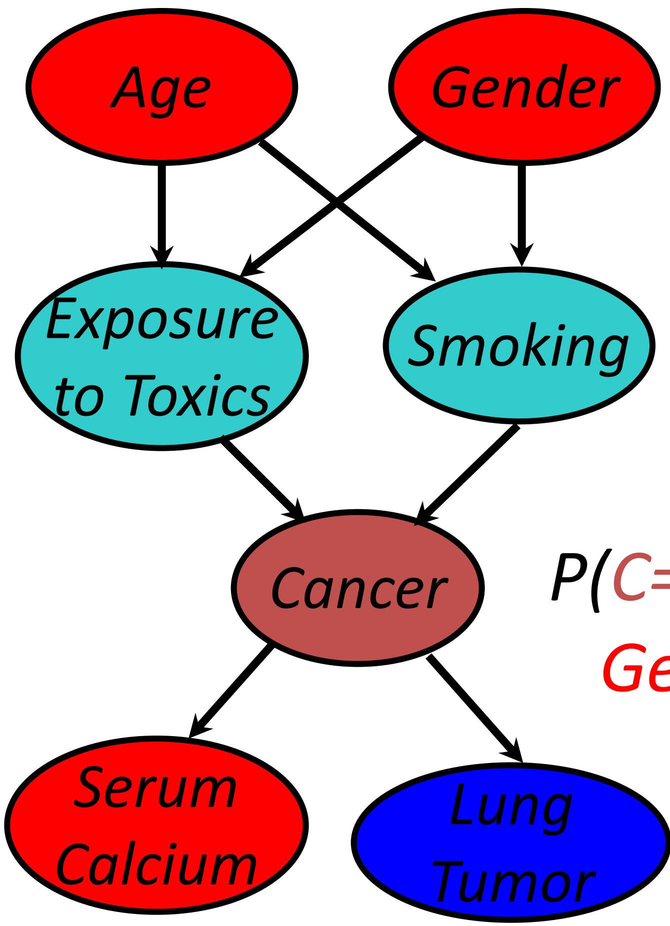
Predictive Inference



How likely are elderly males to get malignant cancer?

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

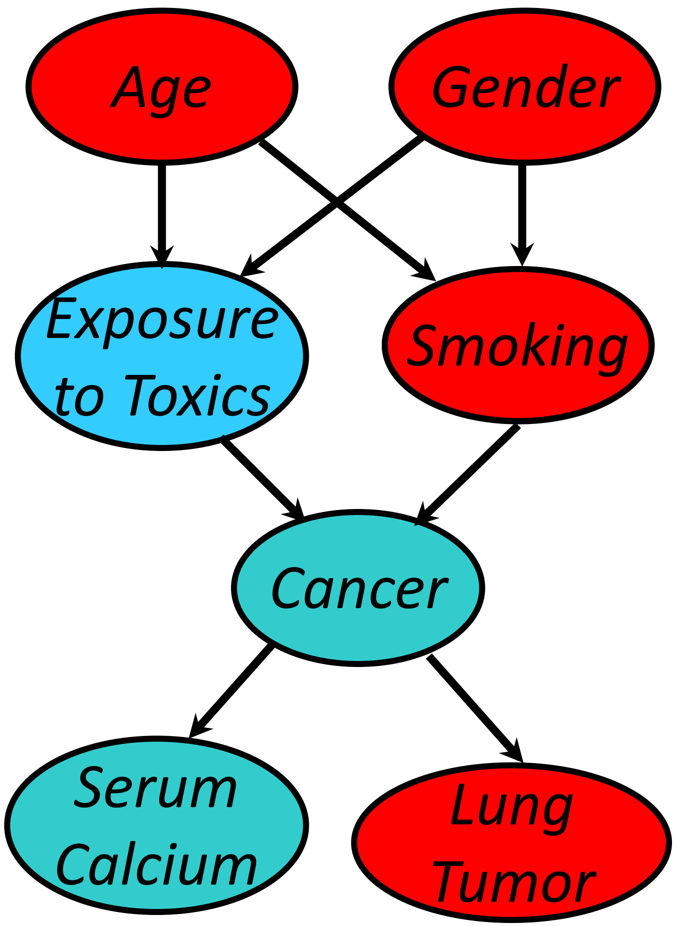
Predictive and diagnostic combined



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

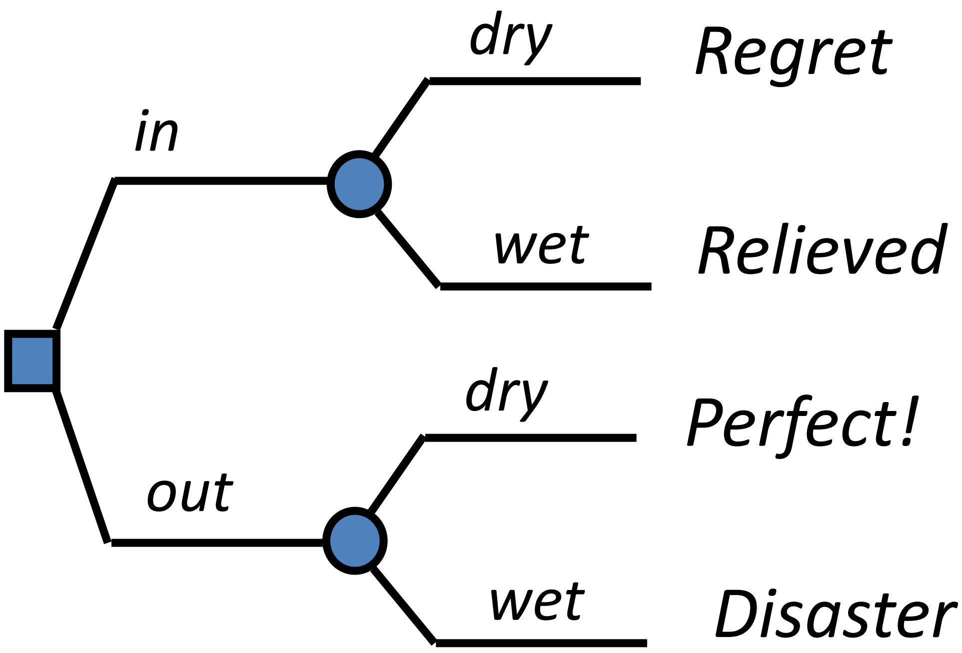
- If we then observe heavy smoking, the probability of exposure to toxics goes back down

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
$$p(Q|x_1, \ldots, x_j)$$
- Variable elimination: An algorithm for exact inference
  - Uses dynamic programming
  - Not necessarily polynomial time!

Variable Elimination (High-level)

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

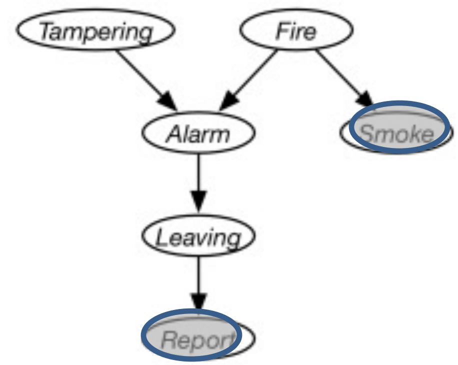
(The word "factor" is used for each CPT.)

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.

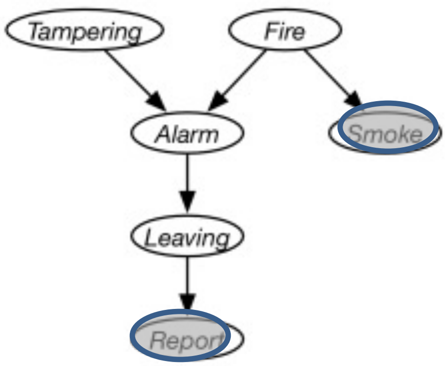


Goal: P(Tampering | Smoke=true ∧ Report=true)

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.



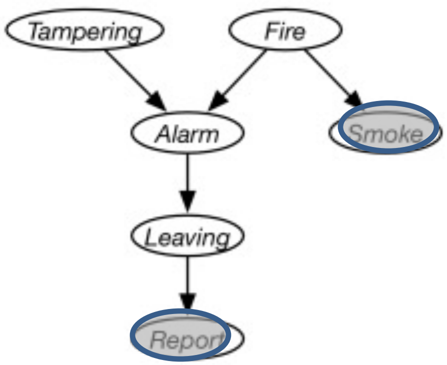
Goal: P(Tampering | Smoke=true ∧ Report=true)

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

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)

Task: Eliminate Fire

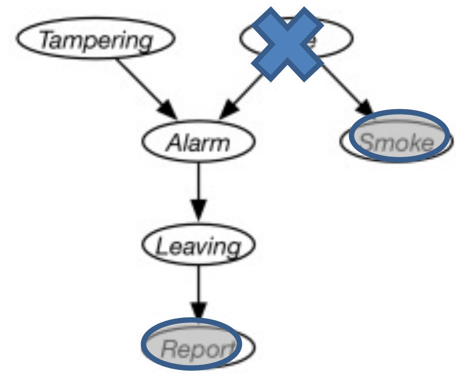
$Conditional Probability$	$Factor$
$P(Tampering)$	$f_0(Tampering)$
$P(Fire)$	$f_1(Fire)$
$P(Alarm \mid Tampering, Fire)$	$f_2(Tampering, Fire, Alarm)$
$P(Smoke = yes \mid Fire)$	$f_3(Fire)$
$P(Leaving \mid Alarm)$	$f_4(Alarm, Leaving)$
$P(Report = yes \mid Leaving)$	$f_5(Leaving)$

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)

f1(Fire)

f2(Tampering, Fire, Alarm)

f3(Fire)

f6(Tampering, Alarm) =

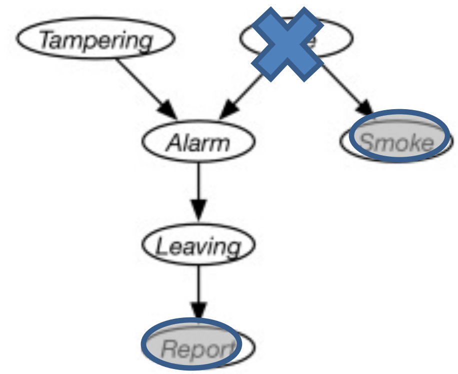
$$= \sum_u f_1(\text{Fire} = u) f_2(T, F = u, A) f_3(F = u)$$

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

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)

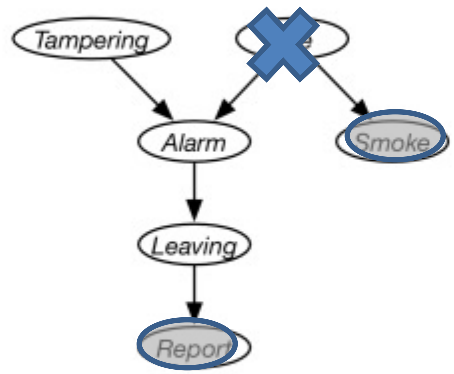
f6(Tampering, Alarm) =

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

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

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) =

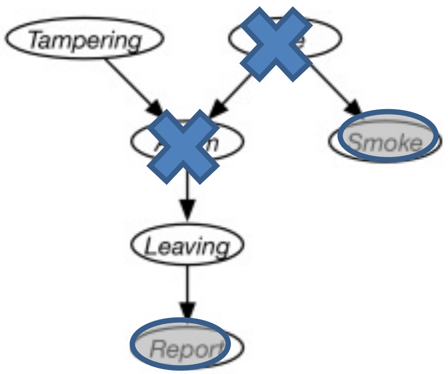
$$= \sum_u p(\text{Fire} = u) p(A \mid T, F = u) p(S = y \mid F = u)$$

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

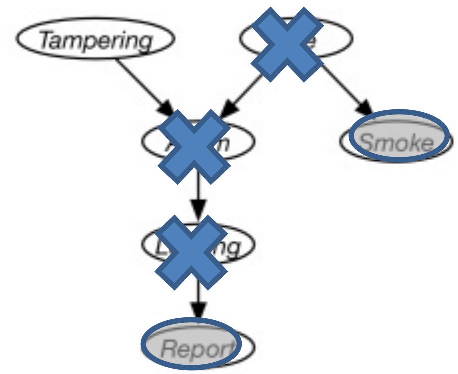
$Conditional Probability$	$Factor$
$P(Tampering)$	$f_0(Tampering)$
$P(Fire)$	$f_1(Fire)$
$P(Alarm \mid Tampering, Fire)$	$f_2(Tampering, Fire, Alarm)$
$P(Smoke = yes \mid Fire)$	$f_3(Fire)$
$P(Leaving \mid Alarm)$	$f_4(Alarm, Leaving)$
$P(Report = yes \mid Leaving)$	$f_5(Leaving)$

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.

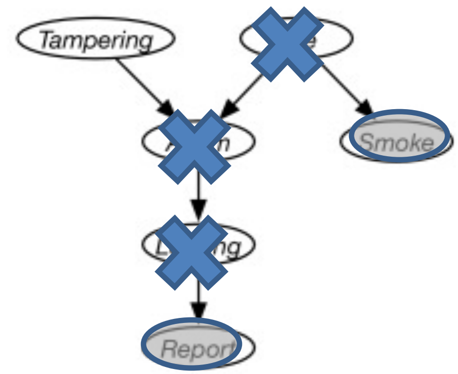
$Conditional Probability$	$Factor$
$P(Tampering)$	$f_0(Tampering)$
$P(Fire)$	$f_1(Fire)$
$P(Alarm \mid Tampering, Fire)$	$f_2(Tampering, Fire, Alarm)$
$P(Smoke = yes \mid Fire)$	$f_3(Fire)$
$P(Leaving \mid Alarm)$	$f_4(Alarm, Leaving)$
$P(Report = yes \mid Leaving)$	$f_5(Leaving)$



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

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

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):

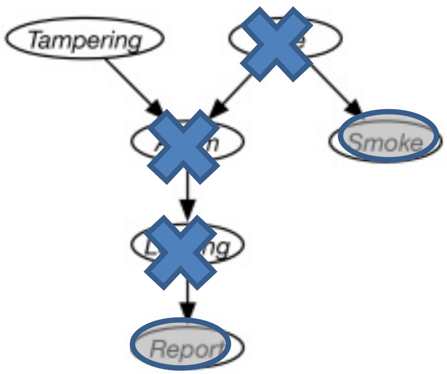
$$p(T = u) = \frac{f_9(T = u)}{\sum_v f_9(T = v)}$$

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)

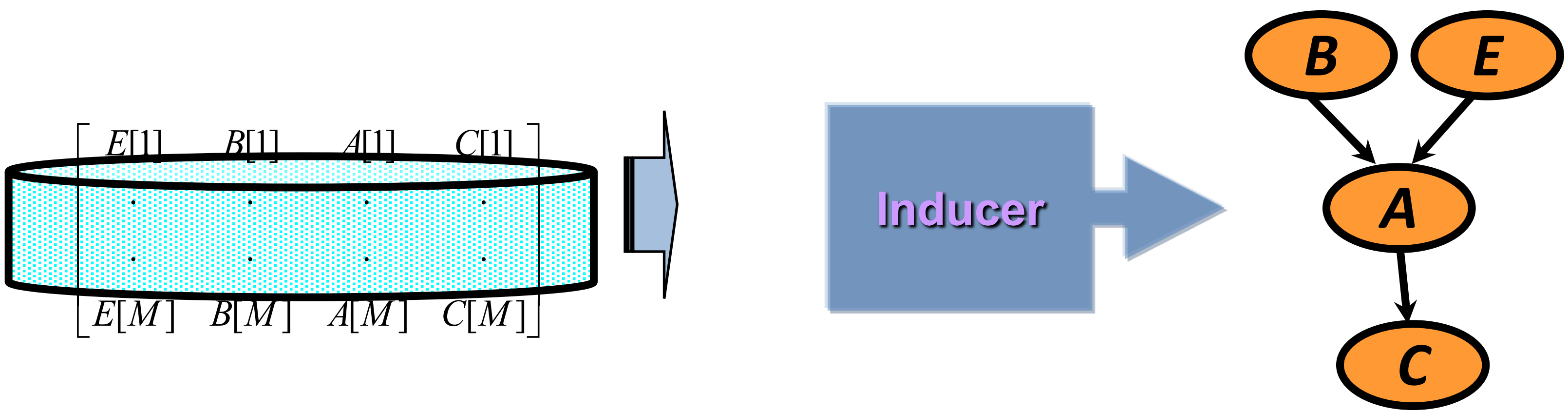
Task: Normalize in order to compute p(Tampering)

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

$$p(T = y) = \frac{f_9(T = y)}{f_9(T = y) + f_9(T = n)}$$

Learning Bayesian networks

- Given training set $\boldsymbol{D} = \{\boldsymbol{x}[1], \ldots, \boldsymbol{x}[M]\}$
- 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

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

- 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

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

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

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

maximize (log-) likelihood to learn the probability parameters

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

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?

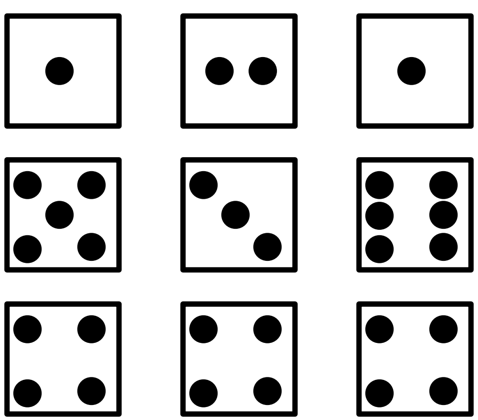
A: Develop a good model for what we observe

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) = ?

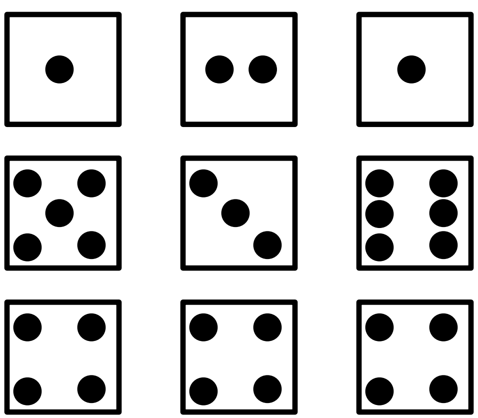
p(6) = ?

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

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

maximize (log-) likelihood to learn the probability parameters

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. $g_\phi(\mathcal{X} = \{x_1, \ldots, x_N\})$ is maximized

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

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. $g_\phi(\mathcal{X} = \{x_1, \ldots, x_N\})$ is maximized

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

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

MLE Snowfall Example

Advanced topic

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?

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

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

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? ✗

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

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

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?

- $\mathcal{X} = \{x_1, x_2, \ldots, x_N\}$ 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

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

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 $\mathcal{Y}|\mathcal{X}$…
  - $f(\mathcal{X}) \rightarrow \{p(\text{yes}|\mathcal{X}), p(\text{no}|\mathcal{X})\}$
- Then re: θ, {predicting, explaining, generating} $\mathcal{Y}$ means… *what*?

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 $\mathcal{Y}|\mathcal{X}$…
- Then re: θ, {predicting, explaining, generating} $\mathcal{Y}$ means… *what*?

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 $\mathcal{Y}|\mathcal{X}$…
- Then re: θ, {predicting, explaining, generating} $\mathcal{Y}$ means finding a value for θ that maximizes the probability of $\mathcal{Y}$ given $\mathcal{X}$

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 $\mathcal{Y}|\mathcal{X}$…
- Then re: θ, {predicting, explaining, generating} $\mathcal{Y}$ means finding a value for θ that maximizes the probability of $\mathcal{Y}$ given $\mathcal{X}$, according to f
- To model $\mathcal{X}$: learn a distribution g, on $\mathcal{X}$

Extended examples of MLE

Advanced topic

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

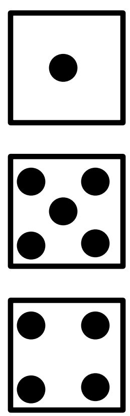
$$p(\overbrace{w_1, w_2, \ldots, w_N}^{\text{N different (independent) rolls}}) = p(w_1)p(w_2)\cdots p(w_N) = \prod_i p(w_i)$$

$w_1 = 1$

$w_2 = 5$

$w_3 = 4$

$\ldots$



Generative Story

for roll $i = 1$ to N:

$w_i \sim \mathrm{Cat}(\theta)$

Maximize Log-likelihood

$$\mathcal{L}(\theta) = \sum_i \log p_\theta(w_i)$$

$$= \sum_i \log \theta_{w_i}$$

Advanced topic

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

$$p(\overbrace{w_1, w_2, \ldots, w_N}^{\text{N different (independent) rolls}}) = p(w_1)p(w_2)\cdots p(w_N) = \prod_i p(w_i)$$

Maximize Log-likelihood (with distribution constraints)

$$\mathcal{L}(\theta) = \sum_i \log \theta_{w_i} \ \text{s.t.} \sum_{k=1}^{6} \theta_k = 1$$

(we can include the inequality constraints $0 \leq \theta_k$, but it complicates the problem and, *right now*, is not needed)

solve using Lagrange multipliers

Advanced topic

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

$$p(\overbrace{w_1, w_2, \dots, w_N}^{\text{N different (independent) rolls}}) = p(w_1)p(w_2)\cdots p(w_N) = \prod_i p(w_i)$$

Maximize Log-likelihood (with distribution constraints)

$$\mathcal{F}(\theta) = \sum_i \log \theta_{w_i} - \lambda\left(\sum_{k=1}^{6} \theta_k - 1\right)$$

(we can include the inequality constraints $0 \leq \theta_k$, but it complicates the problem and, *right now*, is not needed)

$$\frac{\partial \mathcal{F}(\theta)}{\partial \theta_k} = \sum_{i:w_i=k} \frac{1}{\theta_{w_i}} - \lambda \qquad \frac{\partial \mathcal{F}(\theta)}{\partial \lambda} = -\sum_{k=1}^{6} \theta_k + 1$$

Advanced topic

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

$$p(\overbrace{w_1, w_2, \ldots, w_N}^{\text{N different (independent) rolls}}) = p(w_1)p(w_2)\cdots p(w_N) = \prod_i p(w_i)$$

Maximize Log-likelihood (with distribution constraints)

$$\mathcal{F}(\theta) = \sum_i \log \theta_{w_i} - \lambda\left(\sum_{k=1}^{6} \theta_k - 1\right)$$

(we can include the inequality constraints $0 \leq \theta_k$, but it complicates the problem and, *right now*, is not needed)

$$\theta_k = \frac{\sum_{i:w_i=k} 1}{\lambda}$$

$$\text{optimal } \lambda \text{ when } \sum_{k=1}^{6} \theta_k = 1$$

Advanced topic

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

$$p(\overbrace{w_1, w_2, \dots, w_N}^{\text{N different (independent) rolls}}) = p(w_1)p(w_2)\cdots p(w_N) = \prod_i p(w_i)$$

Maximize Log-likelihood (with distribution constraints)

$$\mathcal{F}(\theta) = \sum_i \log \theta_{w_i} - \lambda\left(\sum_{k=1}^{6} \theta_k - 1\right)$$

(we can include the inequality constraints $0 \leq \theta_k$, but it complicates the problem and, *right now*, is not needed)

$$\theta_k = \frac{\sum_{i:w_i=k} 1}{\sum_k \sum_{i:w_i=k} 1} = \frac{N_k}{N} \qquad \text{optimal } \lambda \text{ when } \sum_{k=1}^{6} \theta_k = 1$$

Example: Conditionally Rolling a Die

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

add complexity to better explain what we see

$$p(z_1, w_1, z_2, w_2, \ldots, z_N, w_N) = p(z_1)p(w_1|z_1)\cdots p(z_N)p(w_N|z_N)$$

$$= \prod_i p(w_i|z_i)\, p(z_i)$$

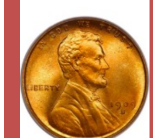
Example: Conditionally Rolling a Die

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

add complexity to better explain what we see

$$p(z_1, w_1, z_2, w_2, \ldots, z_N, w_N) = p(z_1)p(w_1|z_1)\cdots p(z_N)p(w_N|z_N)$$
$$= \prod_i p(w_i|z_i)\, p(z_i)$$

First flip a coin…



$z_1 = T$

$z_2 = H$

…

Example: Conditionally Rolling a Die

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

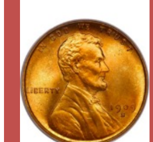
add complexity to better explain what we see

$$p(z_1, w_1, z_2, w_2, \dots, z_N, w_N) = p(z_1)p(w_1|z_1)\cdots p(z_N)p(w_N|z_N)$$

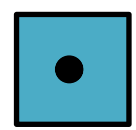
$$= \prod_i p(w_i|z_i)\, p(z_i)$$

First flip a coin…

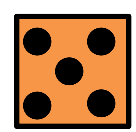
…then roll a different die depending on the coin flip



$z_1 = T$ $\quad w_1 = 1$



$z_2 = H$ $\quad w_2 = 5$



…

Learning in Conditional Die Roll Model: Maximize (Log-)Likelihood

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

$\Big\downarrow$ *add complexity to better explain what we see*

$$p(z_1, w_1, z_2, w_2, \dots, z_N, w_N) = p(z_1)p(w_1|z_1)\cdots p(z_N)p(w_N|z_N)$$
$$= \prod_i p(w_i|z_i)\, p(z_i)$$

If you observe the z_i values, this is easy!

Advanced topic

Learning in Conditional Die Roll Model: Maximize (Log-)Likelihood

$$p(z_1, w_1, z_2, w_2, \dots, z_N, w_N) = \prod_i p(w_i | z_i)\, p(z_i)$$

If you observe the z_i values, this is easy!

First: Write the Generative Story

$\lambda =$ distribution over coin (z)

$\gamma^{(H)} =$ distribution for die when coin comes up heads

$\gamma^{(T)} =$ distribution for die when coin comes up tails

for item $i = 1$ to N:

$\quad z_i \sim \text{Bernoulli}(\lambda)$

$\quad w_i \sim \text{Cat}\left(\gamma^{(z_i)}\right)$

Advanced topic

Learning in Conditional Die Roll Model: Maximize (Log-)Likelihood

$$p(z_1, w_1, z_2, w_2, \ldots, z_N, w_N) = \prod_i p(w_i | z_i)\, p(z_i)$$

If you observe the z_i values, this is easy!

First: Write the Generative Story

$\lambda =$ distribution over coin (z)

$\gamma^{(H)} =$ distribution for H die

$\gamma^{(T)} =$ distribution for T die

for item $i = 1$ to N:

  $z_i \sim \text{Bernoulli}(\lambda)$

  $w_i \sim \text{Cat}\left(\gamma^{(z_i)}\right)$

Second: Generative Story → Objective

$$\mathcal{F}(\theta) = \sum_i^n \left(\log \lambda_{z_i} + \log \gamma_{w_i}^{(z_i)}\right)$$

$-$ Lagrange multiplier constraints

Advanced topic

Learning in Conditional Die Roll Model: Maximize (Log-)Likelihood

$$p(z_1, w_1, z_2, w_2, \dots, z_N, w_N) = \prod_i p(w_i | z_i)\, p(z_i)$$

If you observe the z_i values, this is easy!

First: Write the Generative Story

$\lambda =$ distribution over coin (z)

$\gamma^{(H)} =$ distribution for H die

$\gamma^{(T)} =$ distribution for T die

for item $i = 1$ to N:

$\quad z_i \sim \text{Bernoulli}(\lambda)$

$\quad w_i \sim \text{Cat}\left(\gamma^{(z_i)}\right)$

Second: Generative Story → Objective

$$\mathcal{F}(\theta) = \sum_i^n (\log \lambda_{z_i} + \log \gamma_{w_i}^{(z_i)})$$

$$-\eta\left(\sum_{k=1}^{2} \lambda_k - 1\right) - \sum_{k=1}^{2} \delta_k \left(\sum_{j=1}^{6} \gamma_j^{(k)} - 1\right)$$

Advanced topic

Learning in Conditional Die Roll Model: Maximize (Log-)Likelihood

$$p(z_1, w_1, z_2, w_2, \ldots, z_N, w_N) = \prod_i p(w_i | z_i)\, p(z_i)$$

If you observe the z_i values, this is easy!

But if you don't observe the z_i values, this is not easy!

First: Write the Generative Story

$\lambda = \text{distribution over coin (z)}$

$\gamma^{(H)} = \text{distribution for H die}$

$\gamma^{(T)} = \text{distribution for T die}$

for item $i = 1$ to N:

$z_i \sim \text{Bernoulli}(\lambda)$

$w_i \sim \text{Cat}\left(\gamma^{(z_i)}\right)$

Second: Generative Story → Objective

$$\mathcal{F}(\theta) = \sum_i^n \left(\log \lambda_{z_i} + \log \gamma_{w_i}^{(z_i)}\right)$$

$$-\eta\left(\sum_{k=1}^{2} \lambda_k - 1\right) - \sum_{k=1}^{2} \delta_k \left(\sum_{j=1}^{6} \gamma_j^{(k)} - 1\right)$$

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

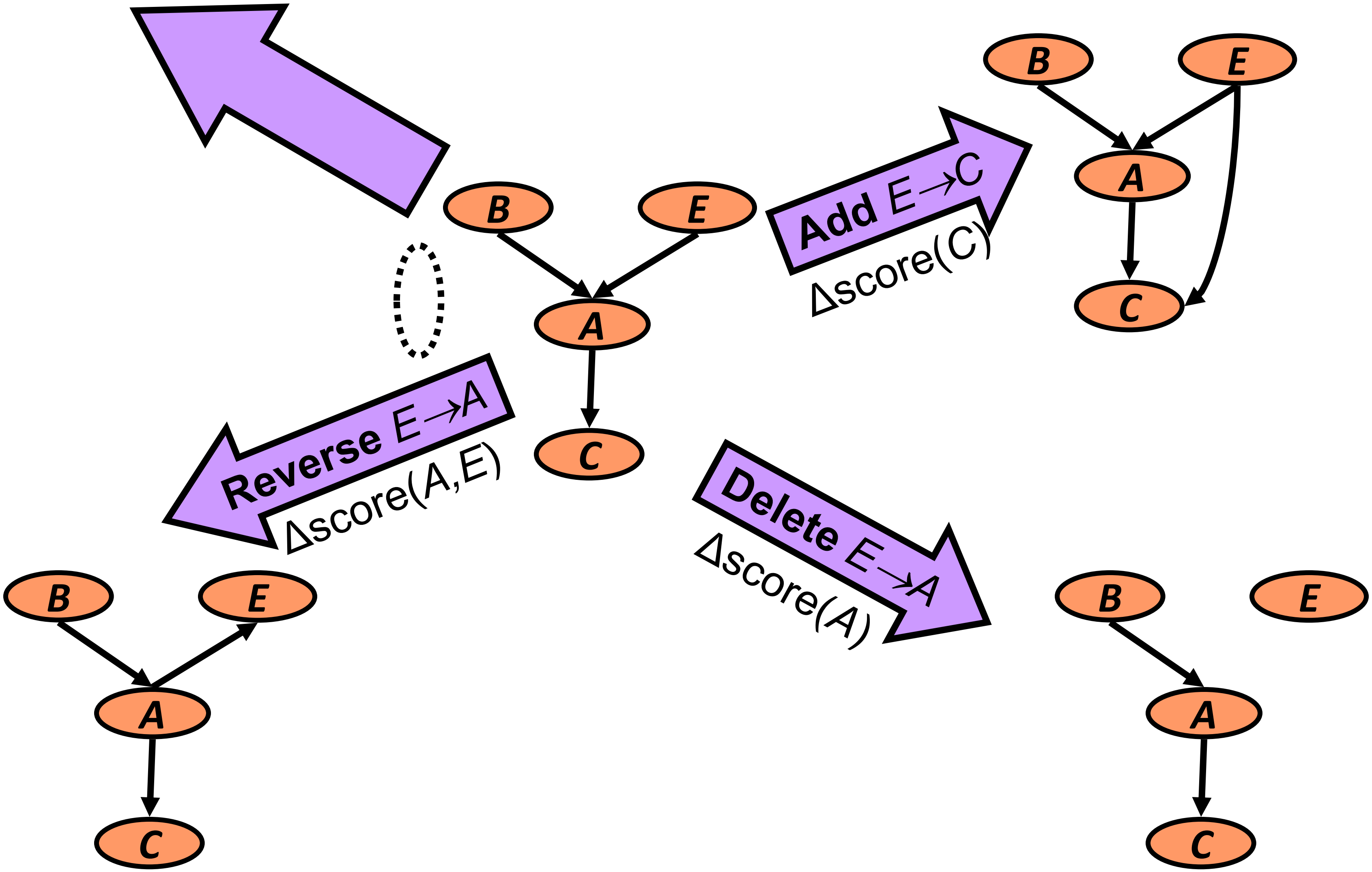
$$\text{Score}(G:D) = \log P(G|D)\ \alpha\ \log [\underbrace{P(D|G)}_{\textbf{\textit{Marginal likelihood}}}\ \underbrace{P(G)}_{\textbf{\textit{Prior}}}]$$

- Marginal likelihood just comes from our parameter estimates
- Prior on structure can be any measure we want; typically a function of the network complexity

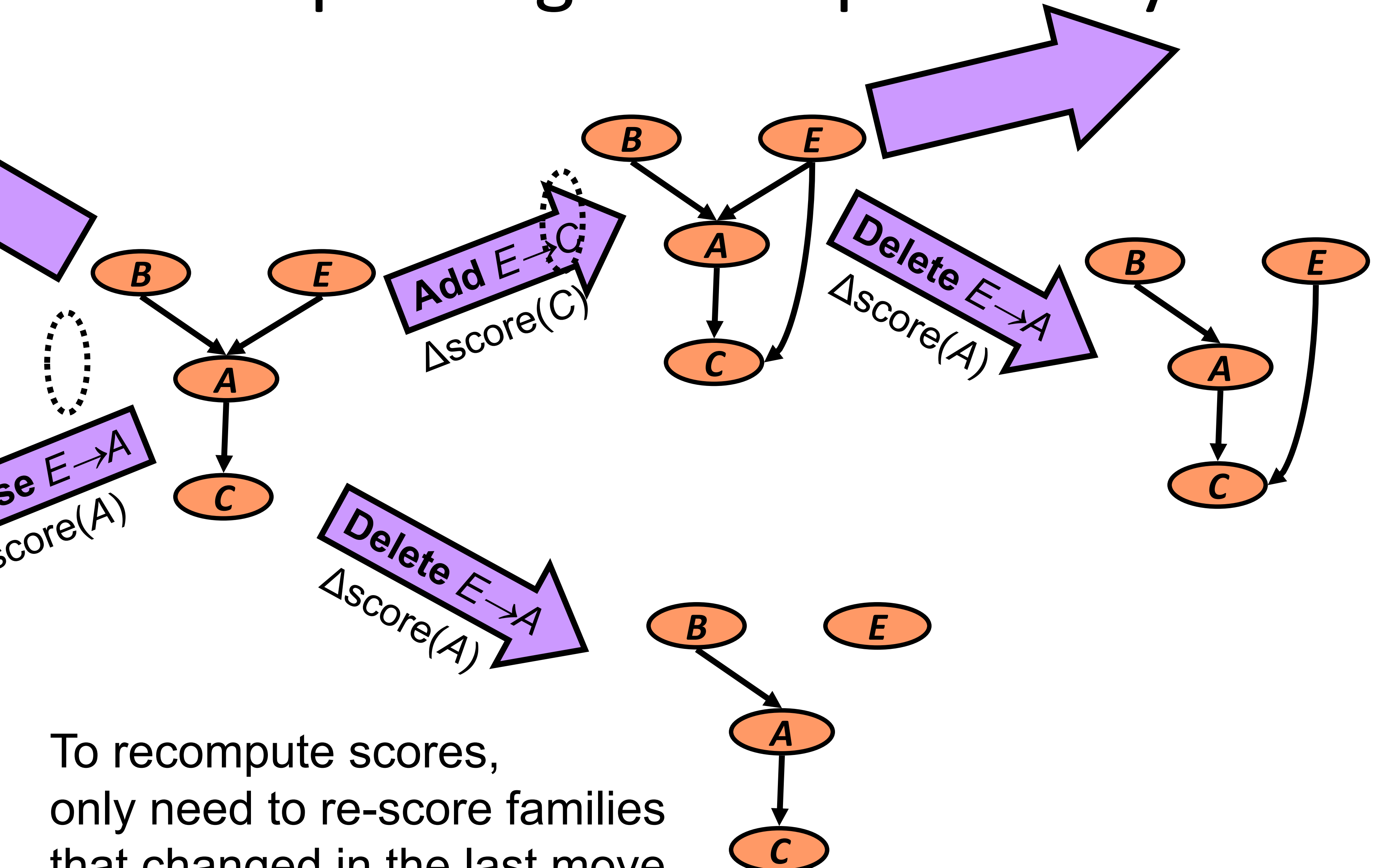
Same key property: Decomposability

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

Heuristic search



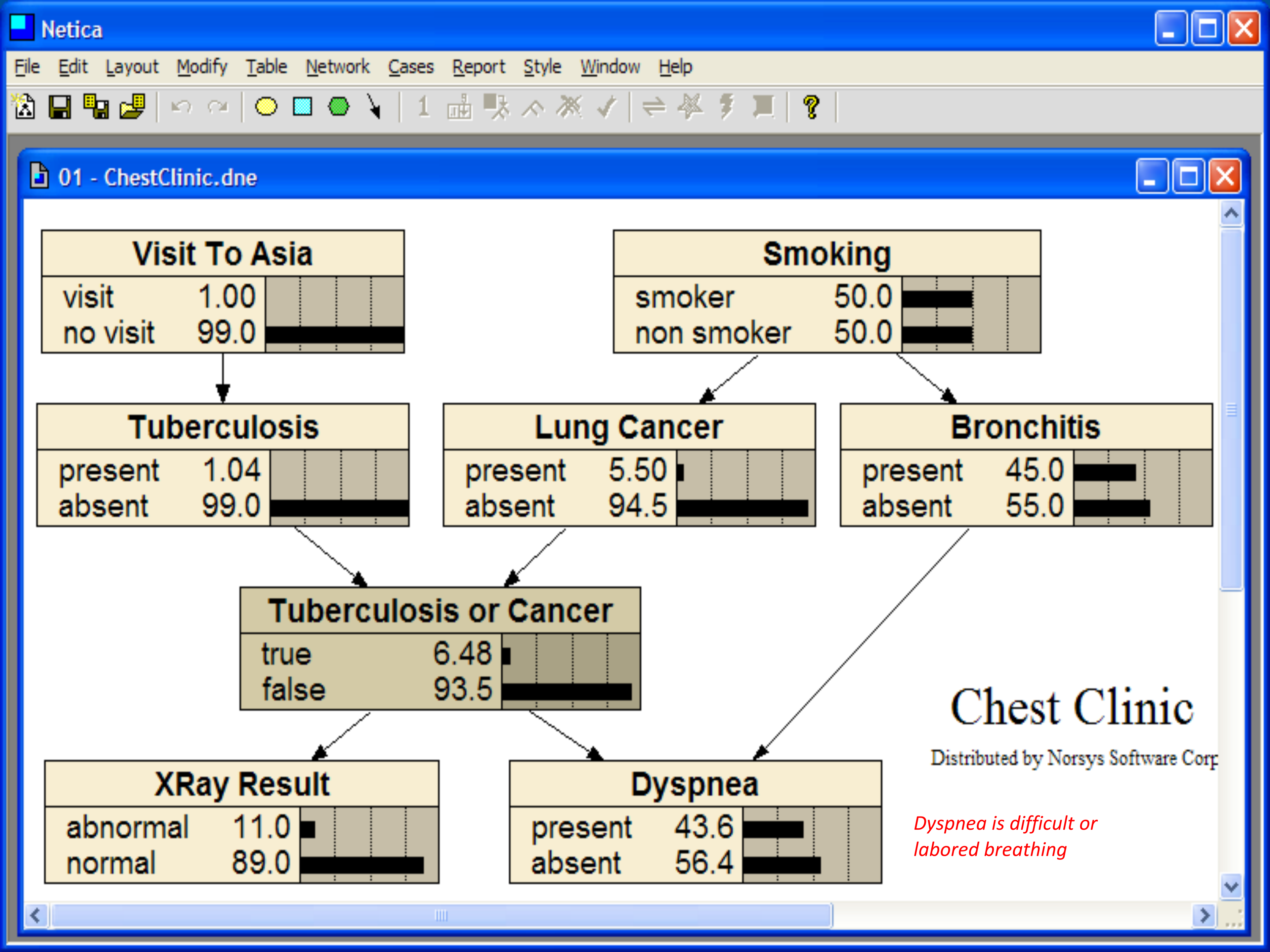
Exploiting decomposability



To recompute scores,
only need to re-score families
that changed in the last move

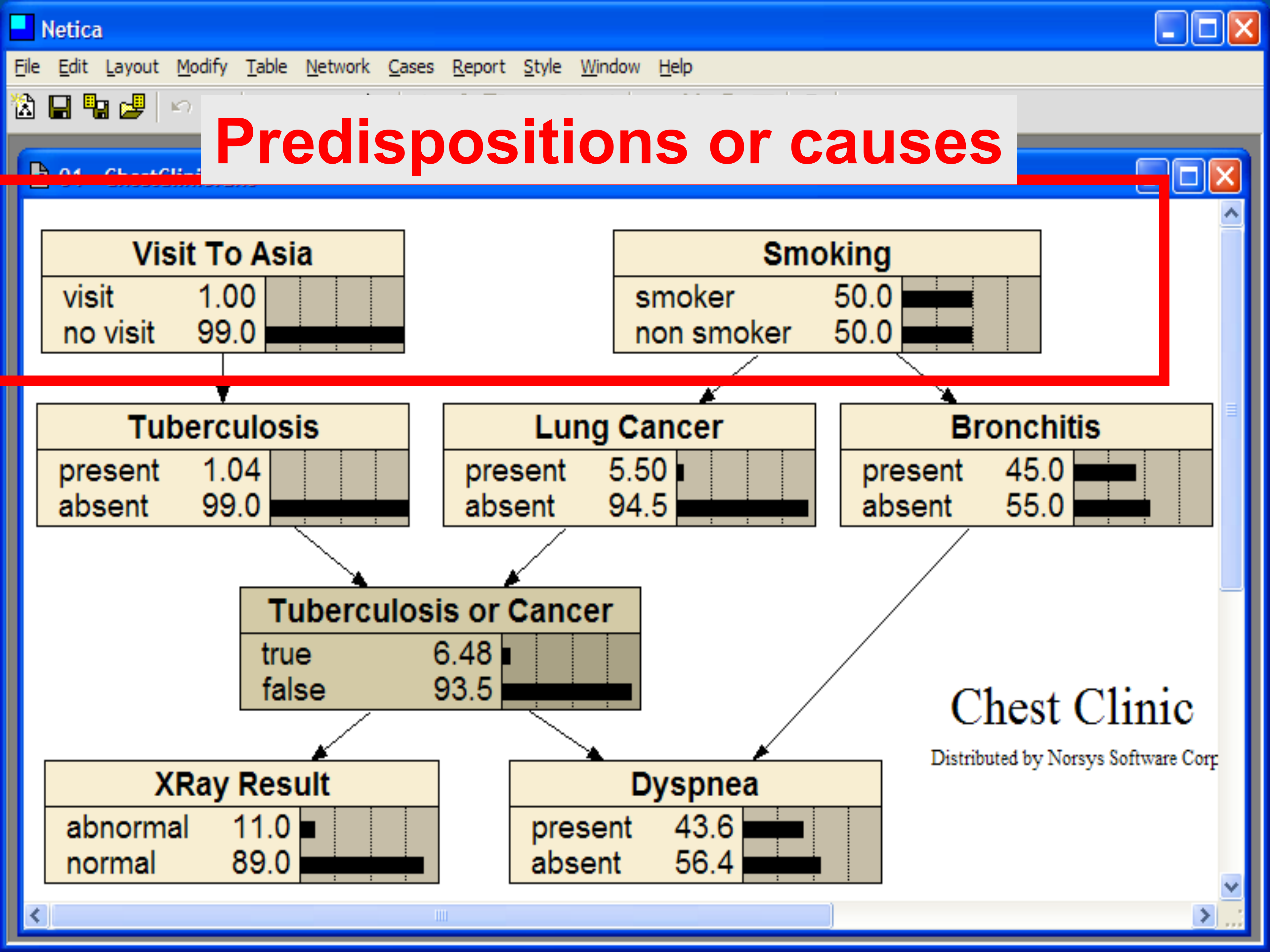
Some software tools

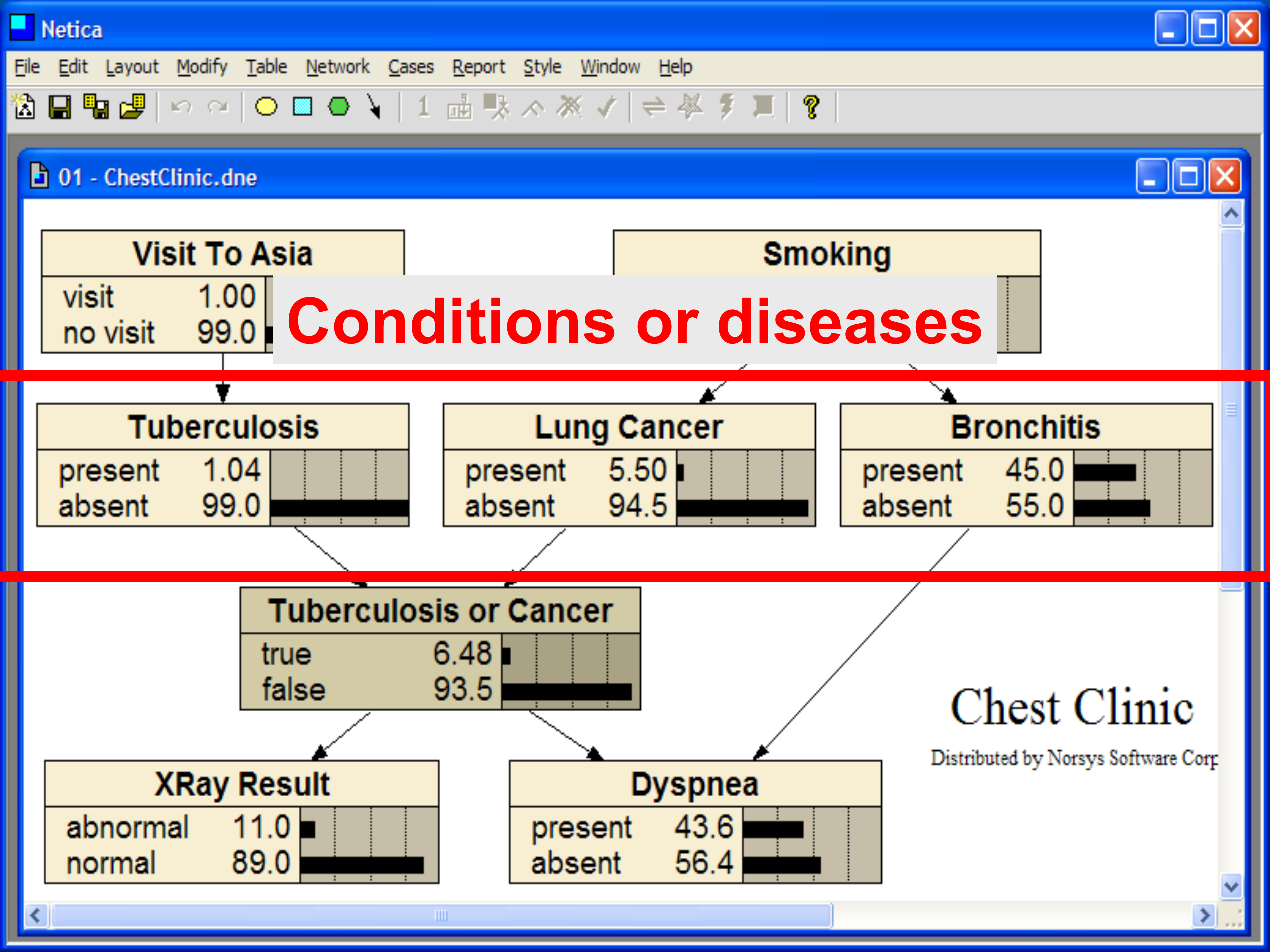
- Netica: Windows app for working with Bayes-ian belief networks and influence diagrams
  - Commercial product, free for small networks
  - Includes graphical editor, compiler, inference engine, etc.
  - To run in OS X or Linus you need Wire or Crossover
- Hugin: free demo versions for Linux, Mac, and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

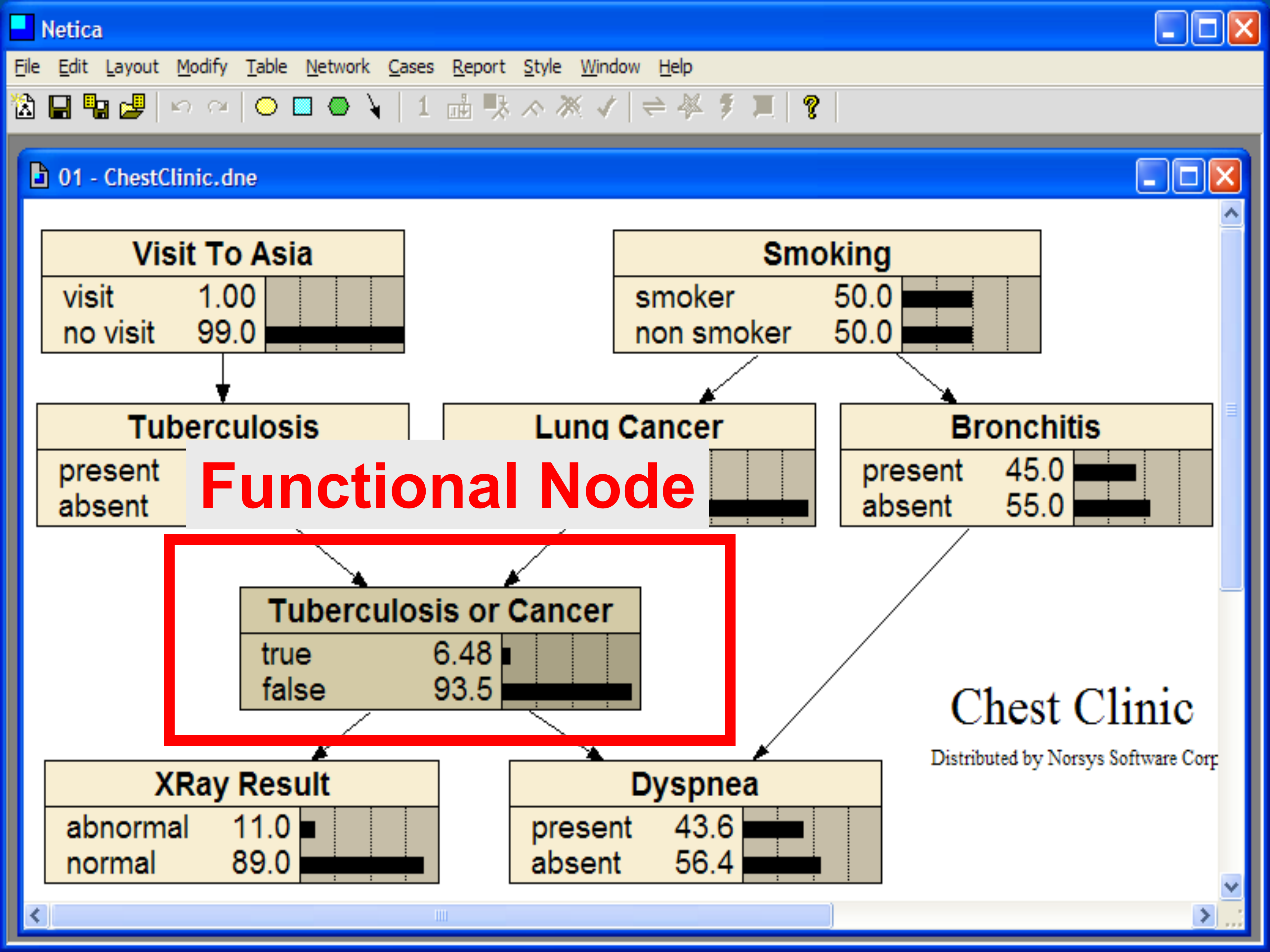


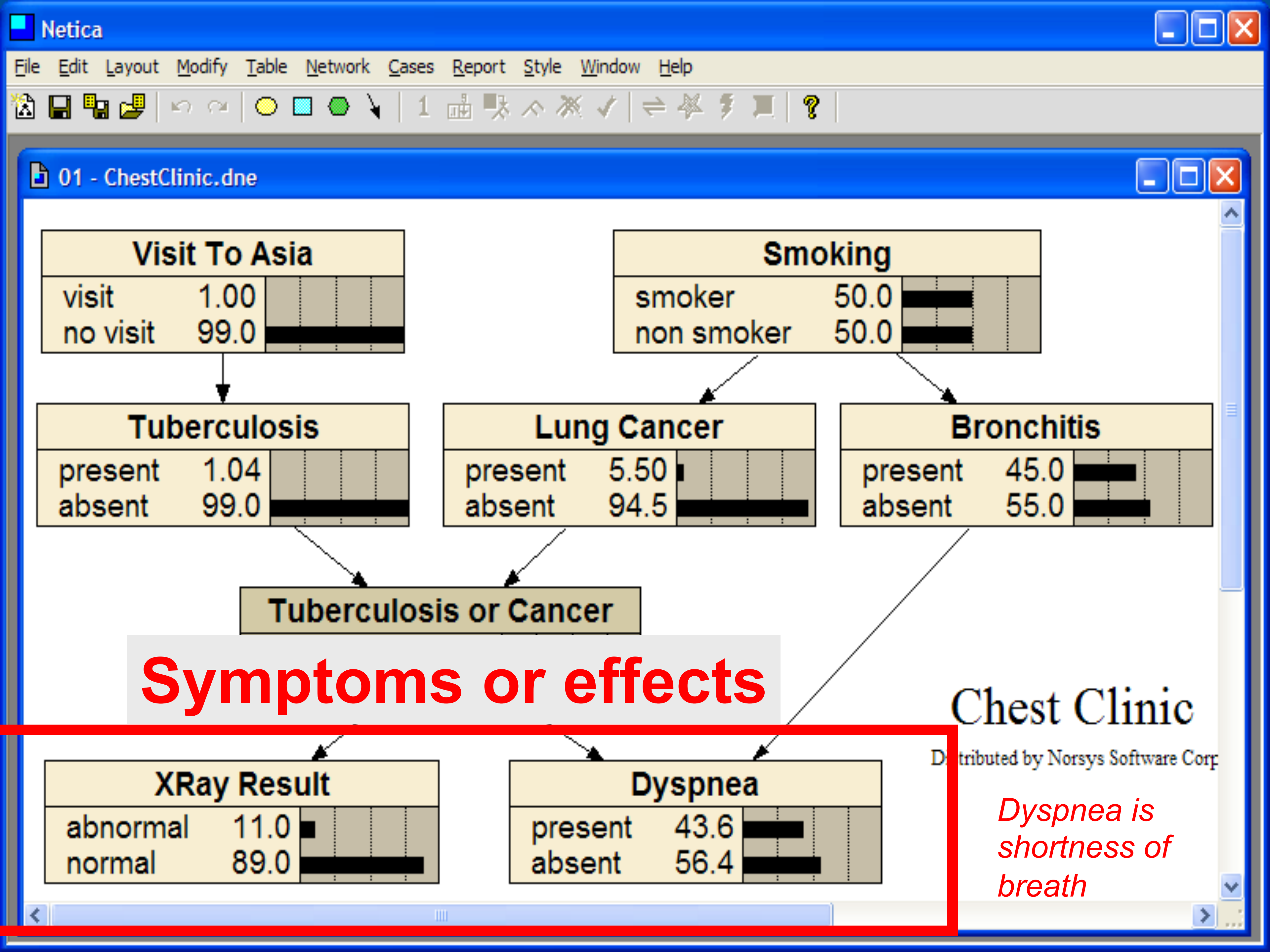
Netica
File Edit Layout Modify Table Network Cases Report Style Window Help
01 - ChestClinic.dne
Visit To Asia
visit 1.00
no visit 99.0
Smoking
smoker 50.0
non smoker 50.0
Tuberculosis
present 1.04
absent 99.0
Lung Cancer
present 5.50
absent 94.5
Bronchitis
present 45.0
absent 55.0
Tuberculosis or Cancer
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
Dyspnea is difficult or
labored breathing

Predispositions or causes



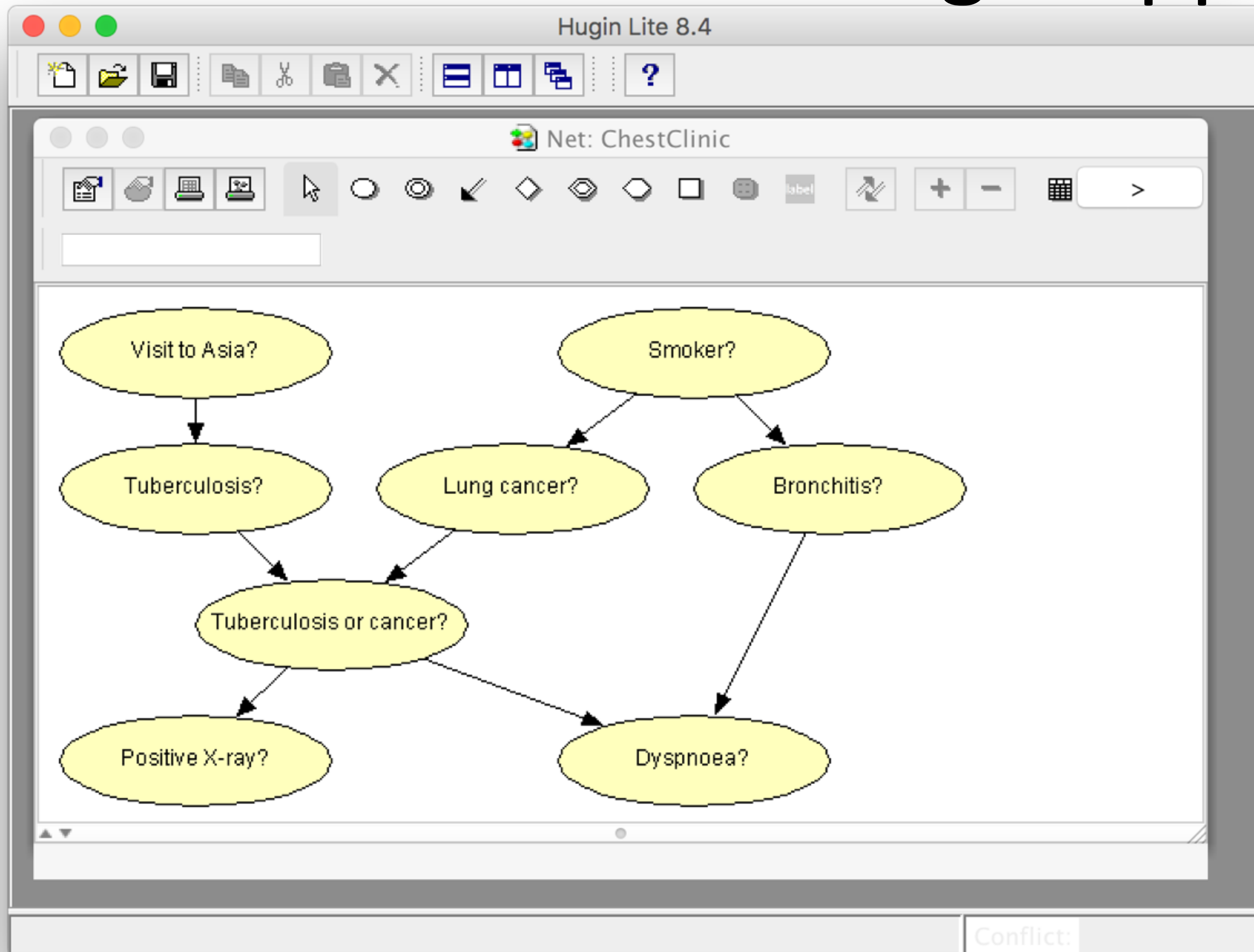

Netica
File Edit Layout Modify Table Network Cases Report Style Window Help
01 - ChestClinic.dne
Visit To Asia
visit 1.00
no visit 99.0
Smoking
Conditions or diseases
Tuberculosis
present 1.04
absent 99.0
Lung Cancer
present 5.50
absent 94.5
Bronchitis
present 45.0
absent 55.0
Tuberculosis or Cancer
true 6.48
false 93.5
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
absent
Lung Cancer
Bronchitis
present 45.0
absent 55.0
Functional Node
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 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

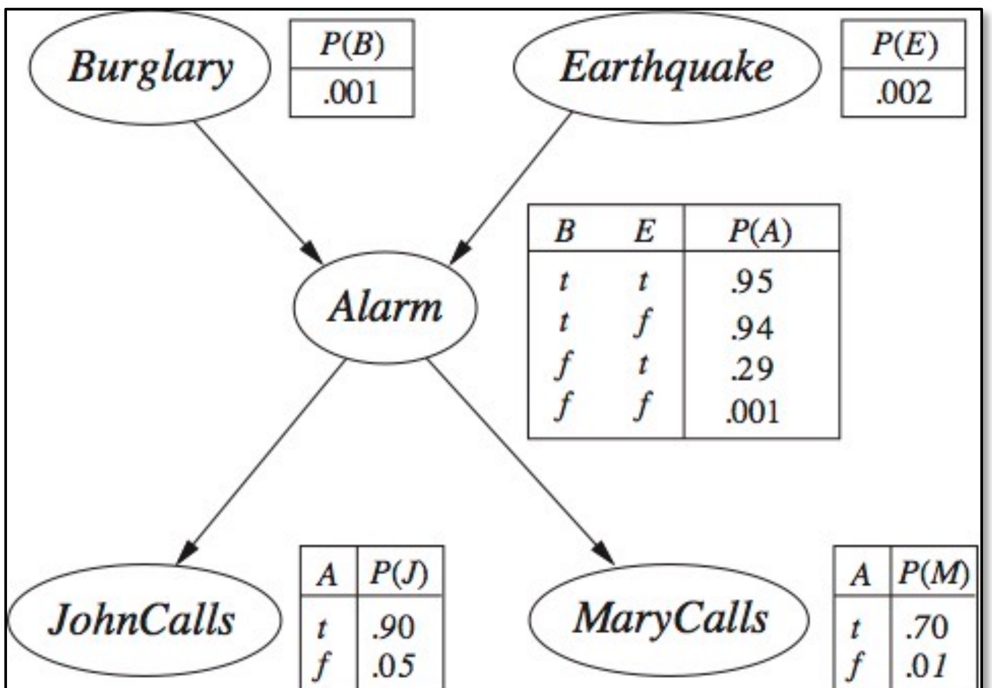
Same BBN model in Hugin app



See the 4-minute HUGIN Tutorial on YouTube

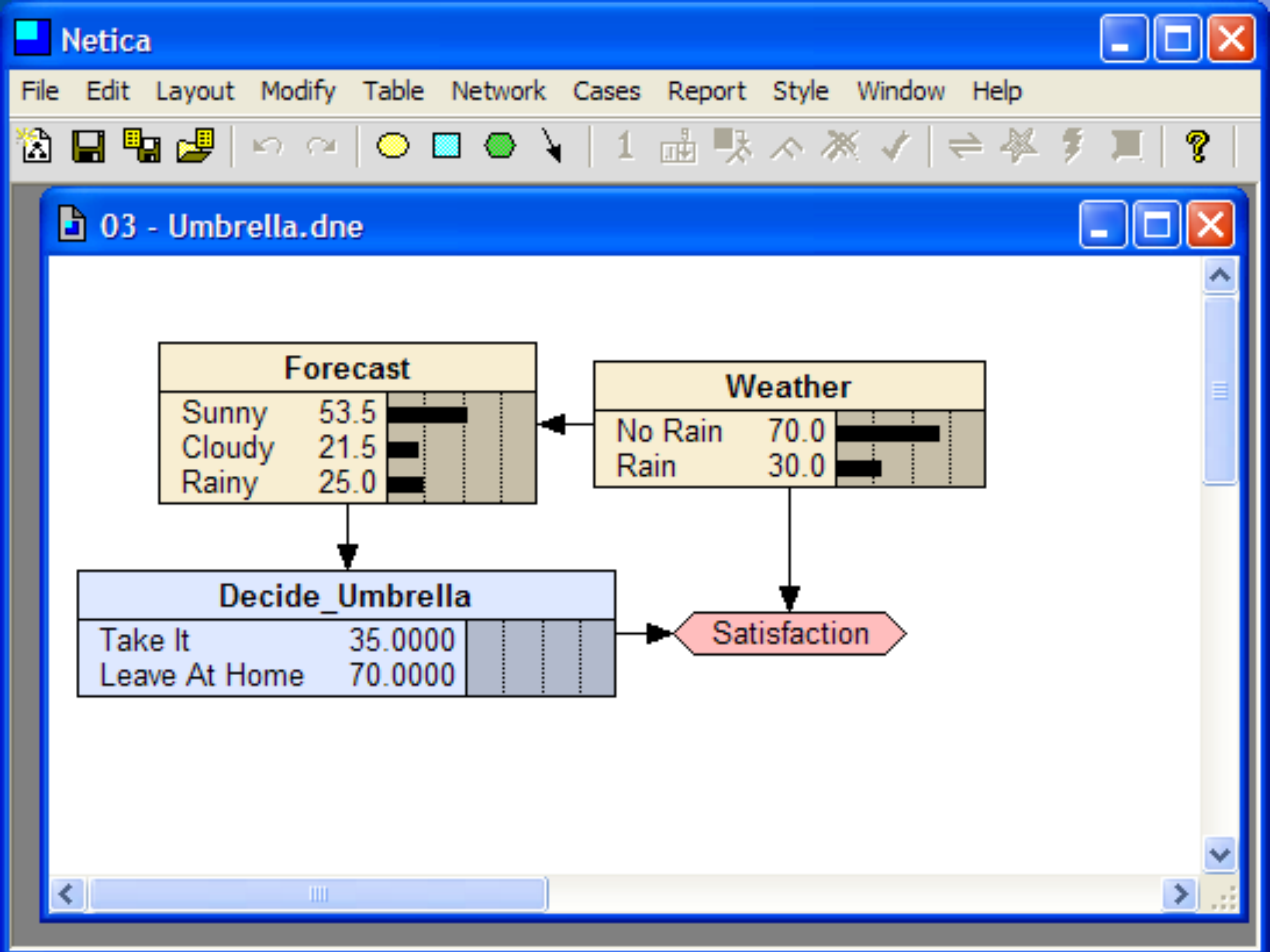
Python Code

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

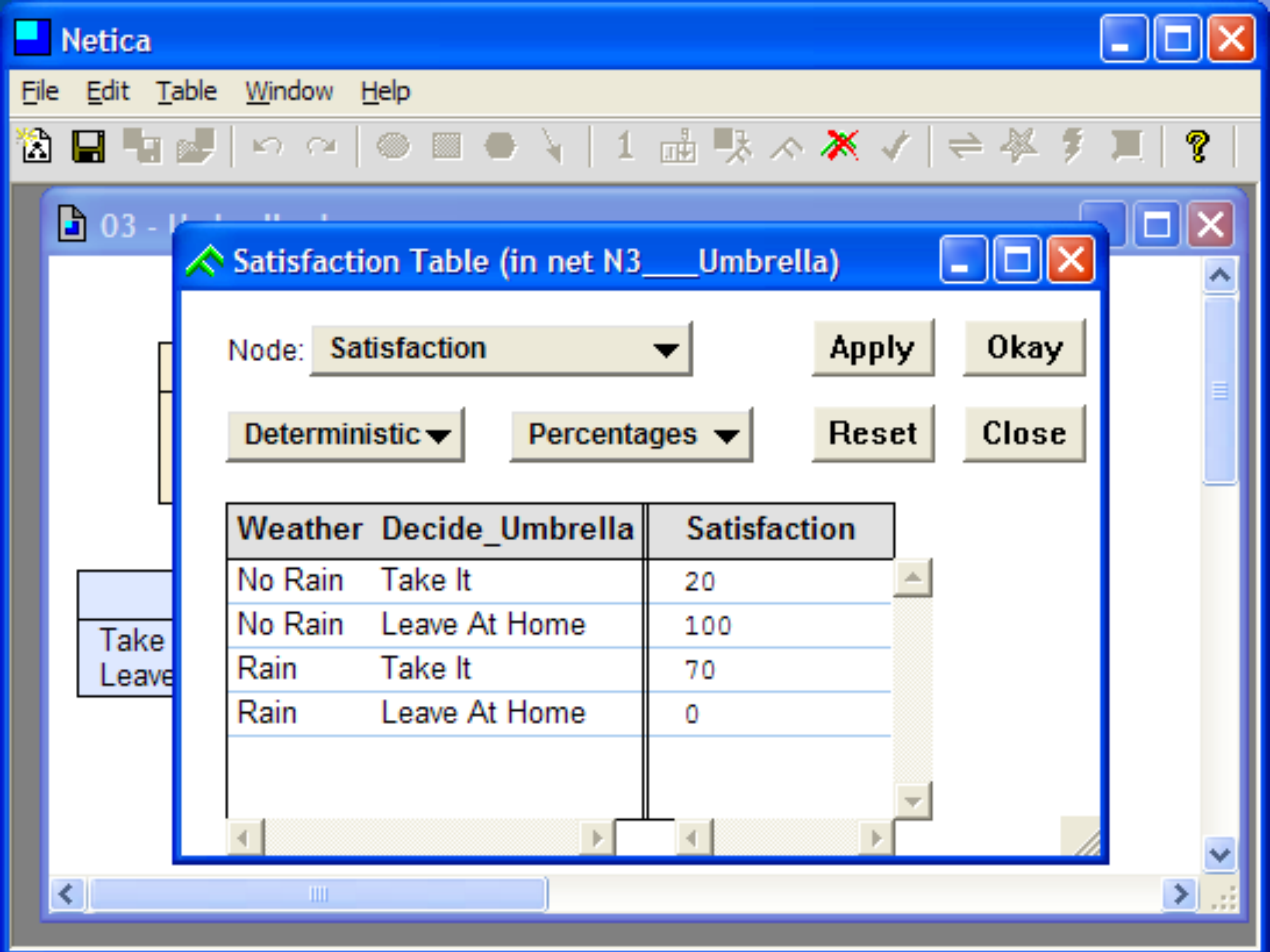


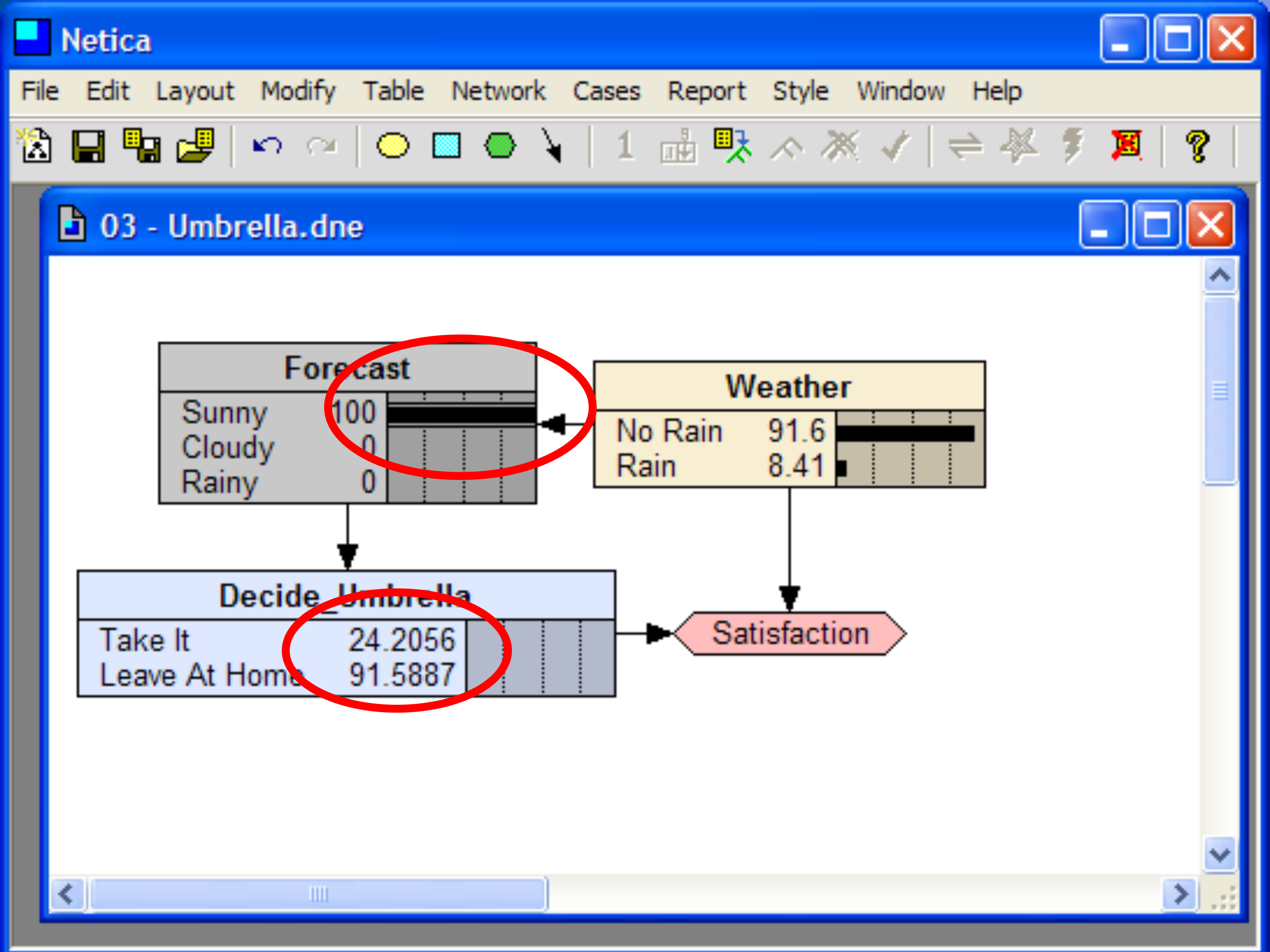
Judea Pearl example

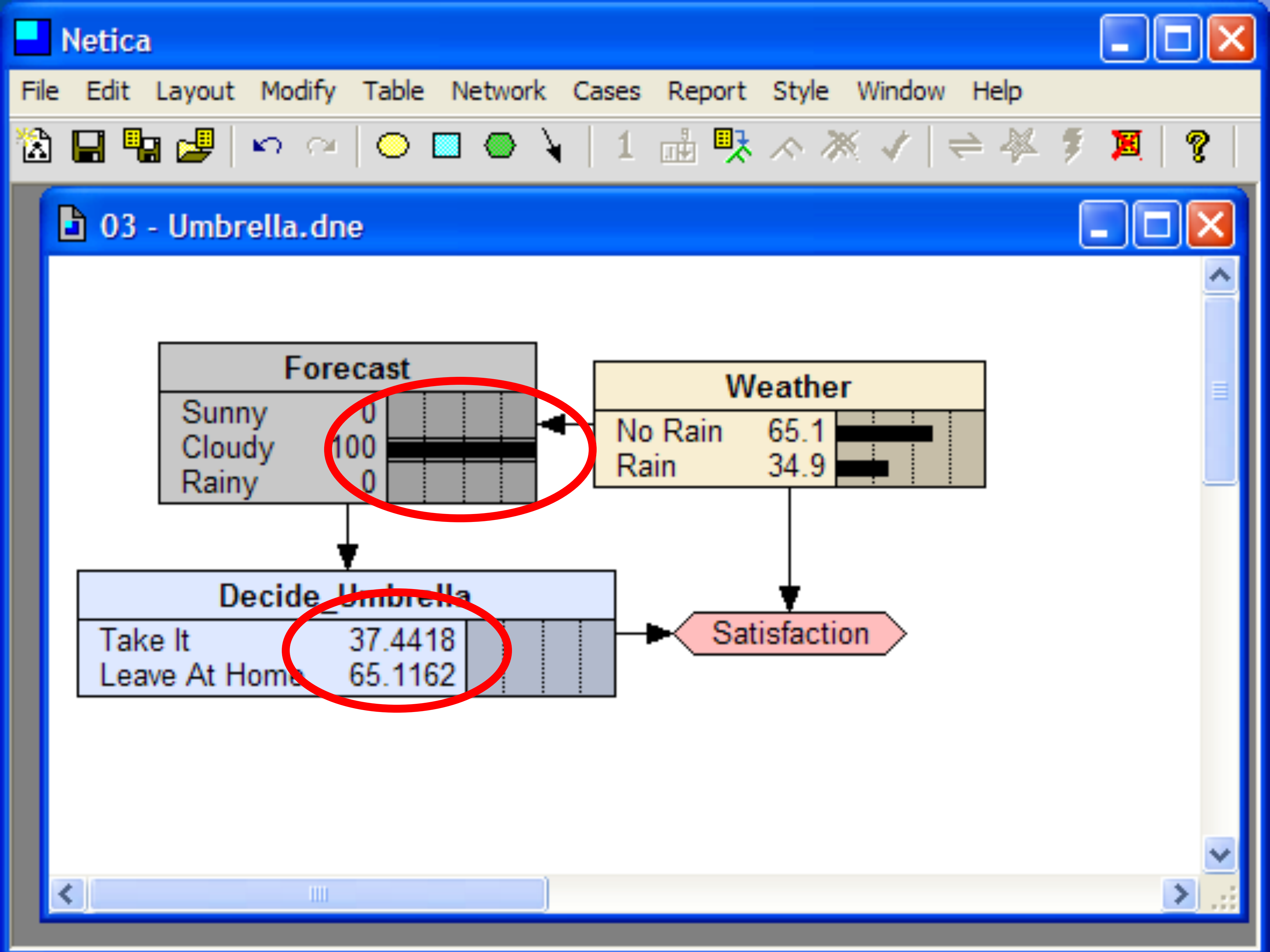
There's is a house with a burglar alarm that can be triggered by a burglary or earthquake. If it sounds, one or both neighbors John & Mary, might call the owner to say the alarm is sounding.



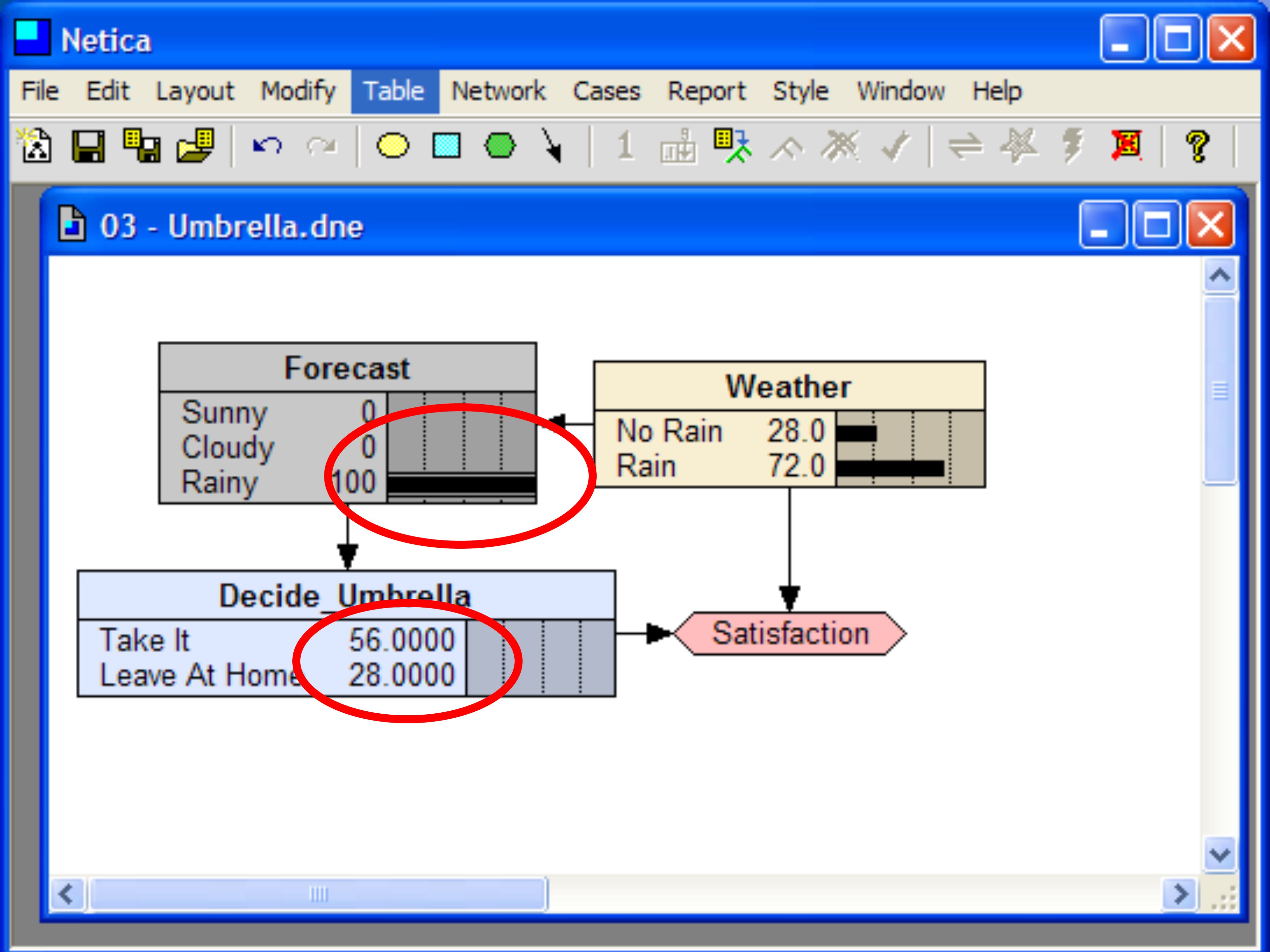
Netica
File Edit Layout Modify Table Network Cases Report Style Window Help
03 - Umbrella.dne
Forecast
Sunny 53.5
Cloudy 21.5
Rainy 25.0
Weather
No Rain 70.0
Rain 30.0
Decide_Umbrella
Take It 35.0000
Leave At Home 70.0000
Satisfaction


Netica
File Edit Table Window Help
03 - U
Satisfaction Table (in net N3___Umbrella)
Node: Satisfaction
Apply
Okay
Deterministic
Percentages
Reset
Close
Weather Decide_Umbrella Satisfaction
No Rain Take It 20
No Rain Leave At Home 100
Rain Take It 70
Rain Leave At Home 0
Take
Leave


Netica
File Edit Layout Modify Table Network Cases Report Style Window Help
03 - Umbrella.dne
Forecast
Sunny 100
Cloudy 0
Rainy 0
Weather
No Rain 91.6
Rain 8.41
Decide_Umbrella
Take It 24.2056
Leave At Home 91.5887
Satisfaction



Netica
File Edit Layout Modify Table Network Cases Report Style Window Help
03 - Umbrella.dne
Forecast
Sunny 0
Cloudy 100
Rainy 0
Weather
No Rain 65.1
Rain 34.9
Decide_Umbrella
Take It 37.4418
Leave At Home 65.1162
Satisfaction


Netica
File
Edit
Layout
Modify
Table
Network
Cases
Report
Style
Window
Help
03 - Umbrella.dne
Forecast
Sunny 0
Cloudy 0
Rainy 100
Weather
No Rain 28.0
Rain 72.0
Decide_Umbrella
Take It 56.0000
Leave At Home 28.0000
Satisfaction