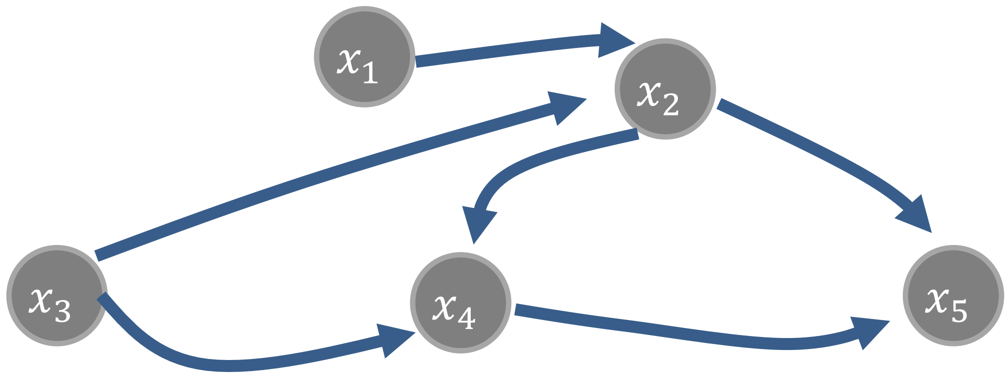
CMSC 471: Reasoning with Bayesian Belief Network

Chapters 12 & 13

KMA Solaiman – ksolaima@umbc.edu

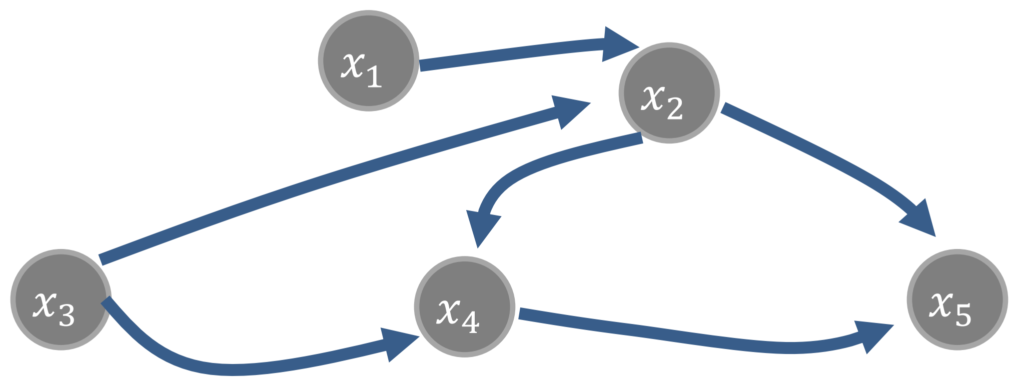
Some slides courtesy Tim Finin and Frank Ferraro

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

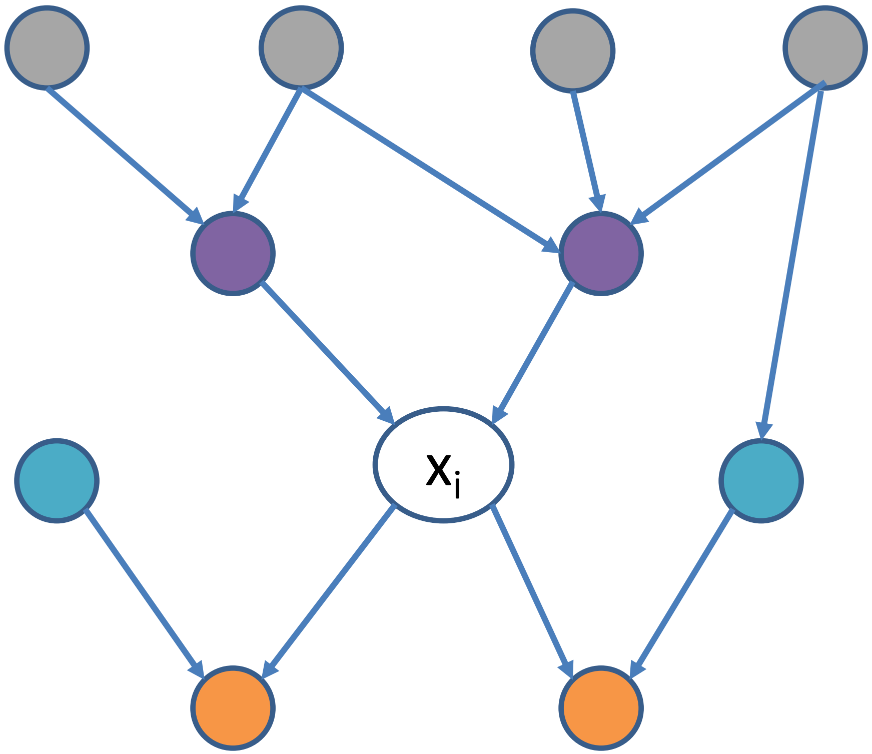


$$p(x_1, x_2, x_3, \dots, x_N) = \prod_i p(x_i \mid \pi(x_i))$$

exact inference in general DAGs is NP-hard

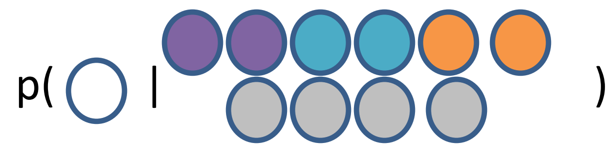
inference in trees can be exact

Markov Blanket

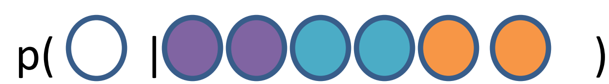


Markov blanket of a node x is its parents, children, and children's parents

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

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

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

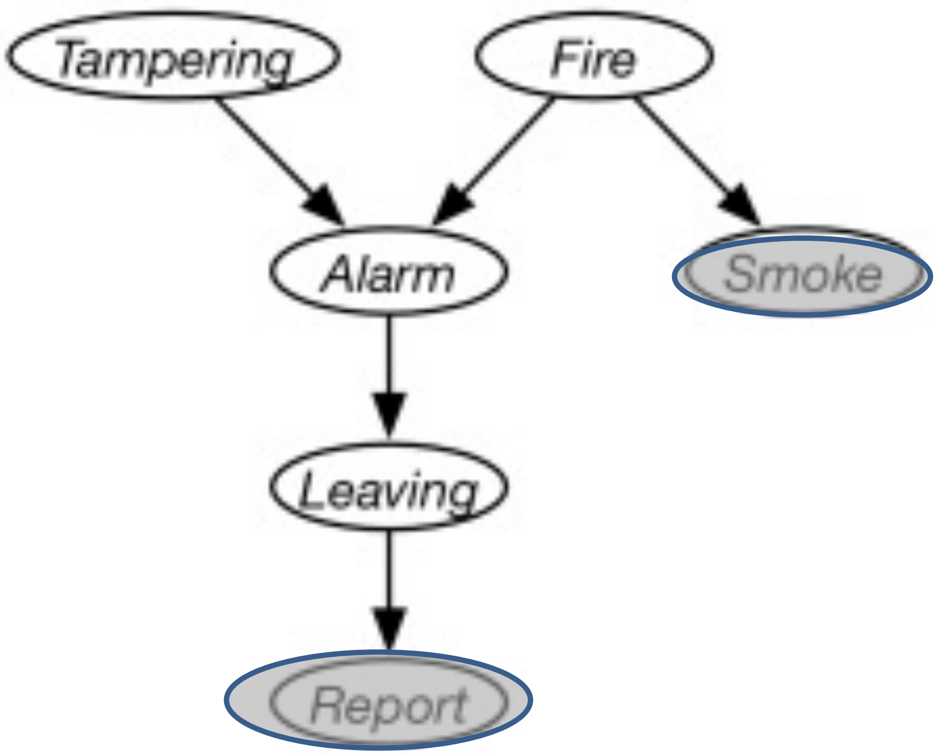
(The word "factor" is used for each CPT.)

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.

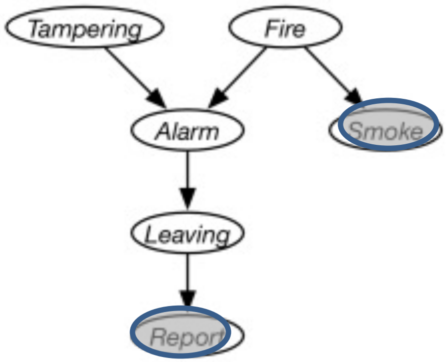


Goal: P(Tampering | Smoke=true ∧ Report=true)

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.



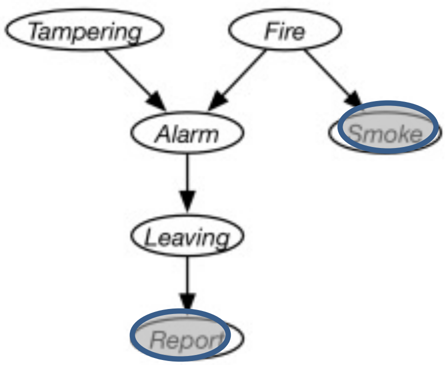
Goal: P(Tampering | Smoke=true ∧ Report=true)

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

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

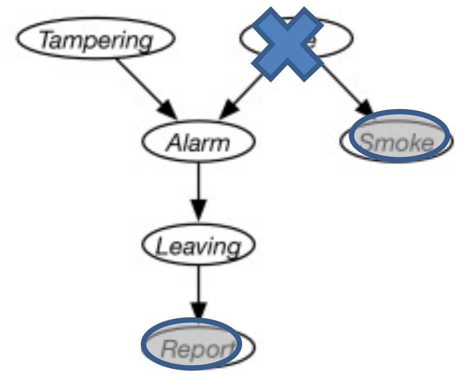
$Conditional Probability$	$Factor$
$P(Tampering)$	$f_0(Tampering)$
$P(Fire)$	$f_1(Fire)$
$P(Alarm \mid Tampering, Fire)$	$f_2(Tampering, Fire, Alarm)$
$P(Smoke = yes \mid Fire)$	$f_3(Fire)$
$P(Leaving \mid Alarm)$	$f_4(Alarm, Leaving)$
$P(Report = yes \mid Leaving)$	$f_5(Leaving)$

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) =

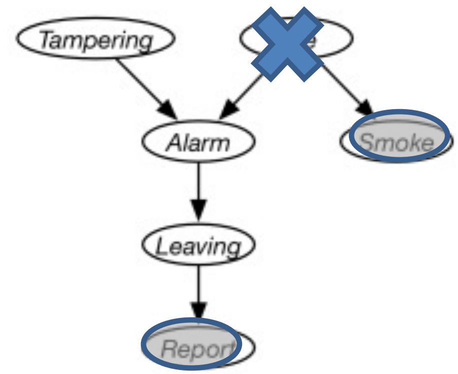
$$= \sum_u f_1(\text{Fire} = u) f_2(T, F = u, A) f_3(F = u)$$

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

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) =

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

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

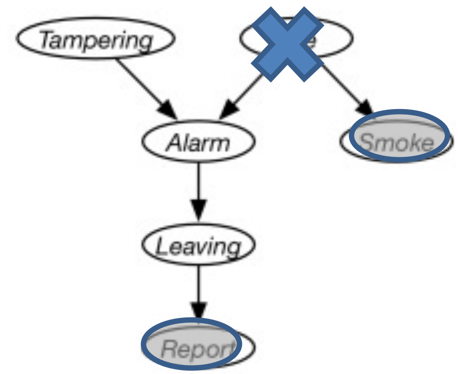
$Conditional Probability$	$Factor$
$P(Tampering)$	$f_0(Tampering)$
$P(Fire)$	$f_1(Fire)$
$P(Alarm \mid Tampering, Fire)$	$f_2(Tampering, Fire, Alarm)$
$P(Smoke = yes \mid Fire)$	$f_3(Fire)$
$P(Leaving \mid Alarm)$	$f_4(Alarm, Leaving)$
$P(Report = yes \mid Leaving)$	$f_5(Leaving)$

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) =

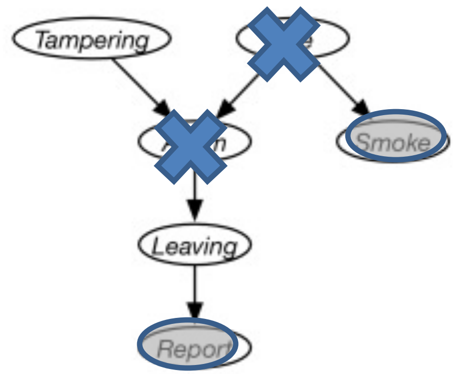
$$= \sum_u p(\text{Fire} = u) p(A \mid T, F = u) p(S = y \mid F = u)$$

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

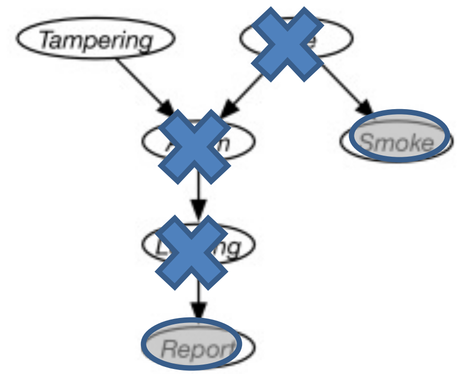
$Conditional Probability$	$Factor$
$P(Tampering)$	$f_0(Tampering)$
$P(Fire)$	$f_1(Fire)$
$P(Alarm \mid Tampering, Fire)$	$f_2(Tampering, Fire, Alarm)$
$P(Smoke = yes \mid Fire)$	$f_3(Fire)$
$P(Leaving \mid Alarm)$	$f_4(Alarm, Leaving)$
$P(Report = yes \mid Leaving)$	$f_5(Leaving)$

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.

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



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

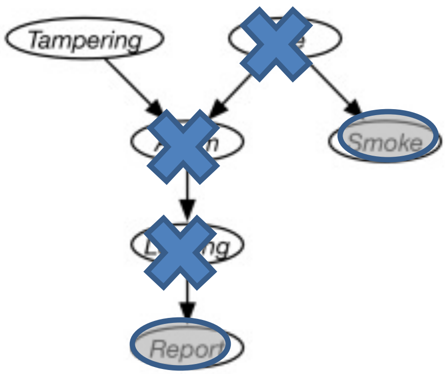
…other computations not shown---see the book or lecture…

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 f8(Tampering):

$$p(T = u) = \frac{f_8(T = u)}{\sum_v f_8(T = v)}$$

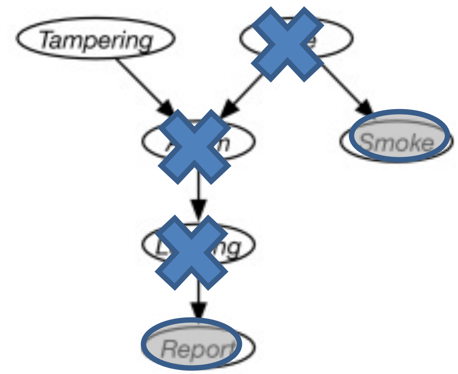
$Conditional Probability$	$Factor$
$P(Tampering)$	$f_0(Tampering)$
$P(Fire)$	$f_1(Fire)$
$P(Alarm \mid Tampering, Fire)$	$f_2(Tampering, Fire, Alarm)$
$P(Smoke = yes \mid Fire)$	$f_3(Fire)$
$P(Leaving \mid Alarm)$	$f_4(Alarm, Leaving)$
$P(Report = yes \mid Leaving)$	$f_5(Leaving)$

Variable Elimination: Example

(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 f8(Tampering):

$$p(T = yes) = \frac{f_8(T = yes)}{f_8(T = yes) + f_8(T = no)}$$

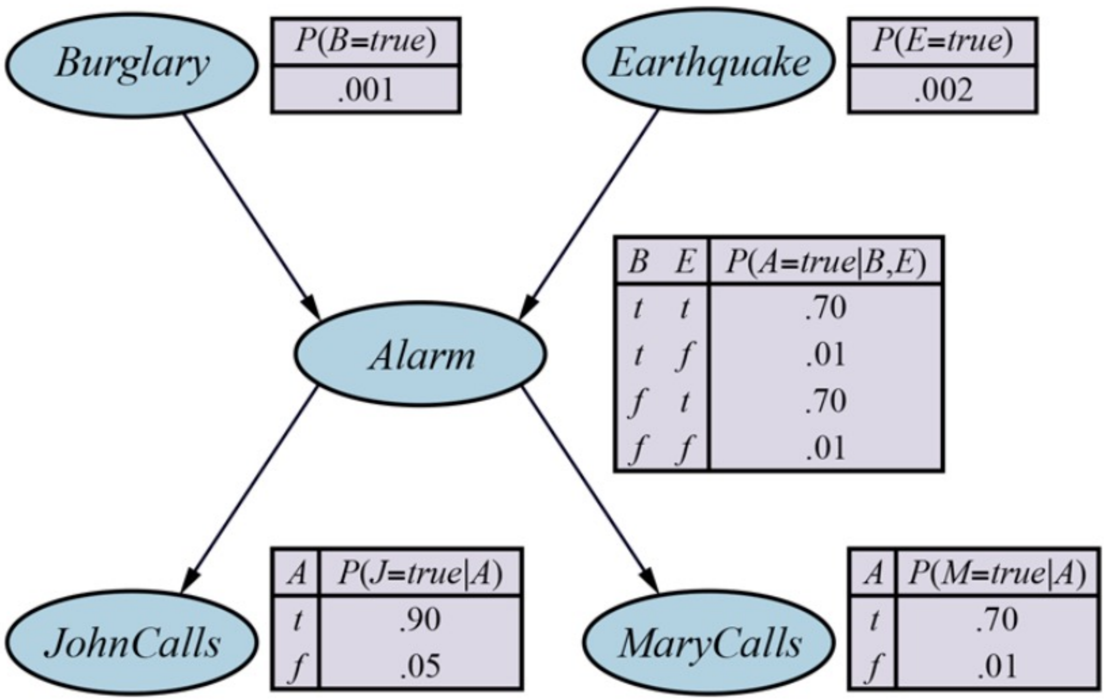
Variable Elimination: Example

- The posterior distribution over *Tampering* is given by

$$\frac{P(Tampering = u)\, f_8(Tampering = u)}{\sum_v P(Tampering = v)\, f_8(Tampering = v)}$$

Another example

Figure 13.2



$$\mathbf{P}(Burglary | JohnCalls = true, MaryCalls = true) = \langle 0.284, 0.716 \rangle.$$

$$\mathbf{P}(B|j,m) = \alpha\,\mathbf{P}(B,j,m) = \alpha \sum_{e}\sum_{a}\mathbf{P}(B,j,m,e,a).$$

$$P(b|j,m) = \alpha \sum_{e}\sum_{a} P(b)P(e)P(a|b,e)P(j|a)P(m|a).$$

$$P(b|j,m) = \alpha\, P(b)\sum_{e} P(e)\sum_{a} P(a|b,e)P(j|a)P(m|a).$$

$$\mathbf{P}(B|j,m) = \alpha \underbrace{\mathbf{P}(B)}_{\mathbf{f}_1(B)} \sum_{e} \underbrace{P(e)}_{\mathbf{f}_2(E)} \sum_{a} \underbrace{\mathbf{P}(a|B,e)}_{\mathbf{f}_3(A,B,E)} \underbrace{P(j|a)}_{\mathbf{f}_4(A)} \underbrace{P(m|a)}_{\mathbf{f}_5(A)}.$$

$$\mathbf{P}(B|j,m) = \alpha\,\mathbf{f}_1(B) \times \sum_{e} \mathbf{f}_2(E) \times \sum_{a} \mathbf{f}_3(A,B,E) \times \mathbf{f}_4(A) \times \mathbf{f}_5(A).$$

$$\begin{aligned}\mathbf{f}_6(B,E) &= \sum_a \mathbf{f}_3(A,B,E) \times \mathbf{f}_4(A) \times \mathbf{f}_5(A) \\ &= (\mathbf{f}_3(a,B,E) \times \mathbf{f}_4(a) \times \mathbf{f}_5(a)) + (\mathbf{f}_3(\neg a,B,E) \times \mathbf{f}_4(\neg a) \times \mathbf{f}_5(\neg a)).\end{aligned}$$

Now we are left with the expression

$$\mathbf{P}(B|j,m) = \alpha\, \mathbf{f}_1(B) \times \sum_e \mathbf{f}_2(E) \times \mathbf{f}_6(B,E).$$

- Next, we sum out E from the product of $\mathbf{f}_2$ and $\mathbf{f}_6$:

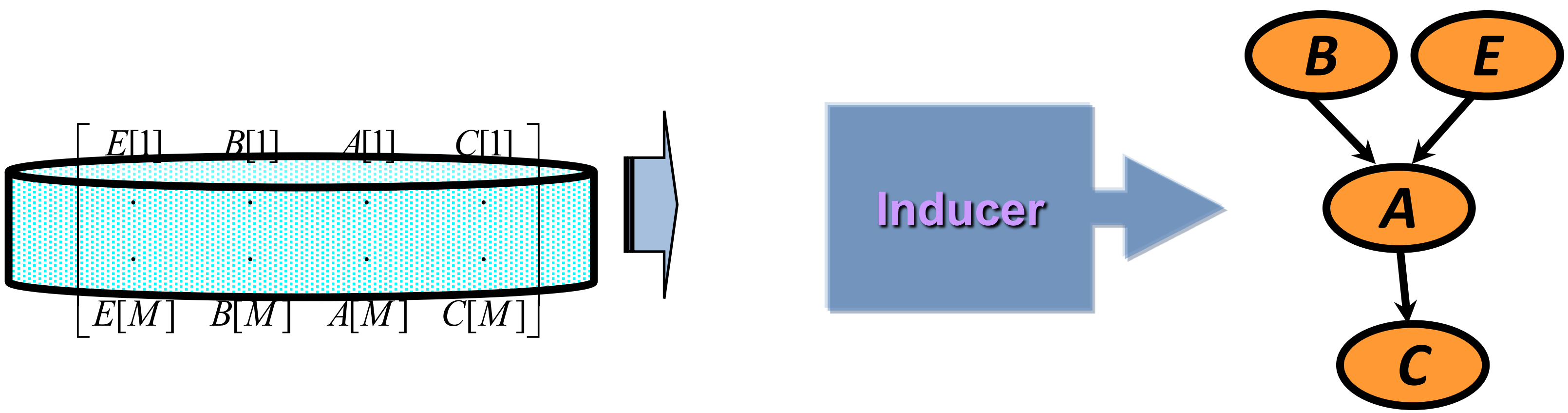
$$\begin{aligned}\mathbf{f}_7(B) &= \sum_e \mathbf{f}_2(E) \times \mathbf{f}_6(B,E) \\ &= \mathbf{f}_2(e) \times \mathbf{f}_6(B,e) + \mathbf{f}_2(\neg e) \times \mathbf{f}_6(B,\neg e).\end{aligned}$$

This leaves the expression

$$\mathbf{P}(B|j,m) = \alpha\, \mathbf{f}_1(B) \times \mathbf{f}_7(B)$$

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

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

- 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

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

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

maximize (log-) likelihood to learn the probability parameters

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

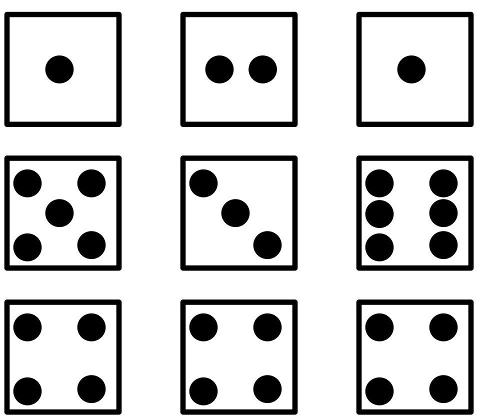
A: Develop a good model for what we observe

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) = ?

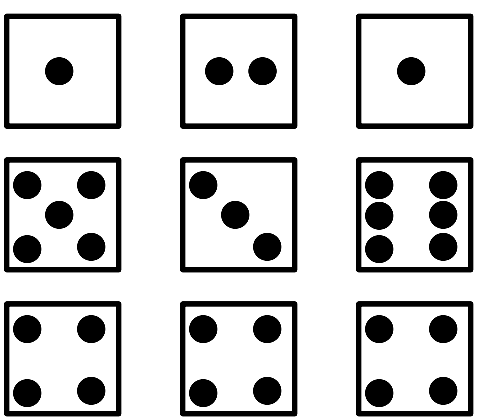
p(6) = ?

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. $g_\phi(\mathcal{X} = \{x_1, \dots, x_N\})$ 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

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

MLE Snowfall Example

Advanced topic

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

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

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?

- $\mathcal{X} = \{x_1, x_2, \dots, x_N\}$ 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? ✓ $p(X = x) = \dfrac{x^{k-1}\exp\left(\frac{-k}{\theta}\right)}{\theta^k \Gamma(k)}$
- 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?

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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)

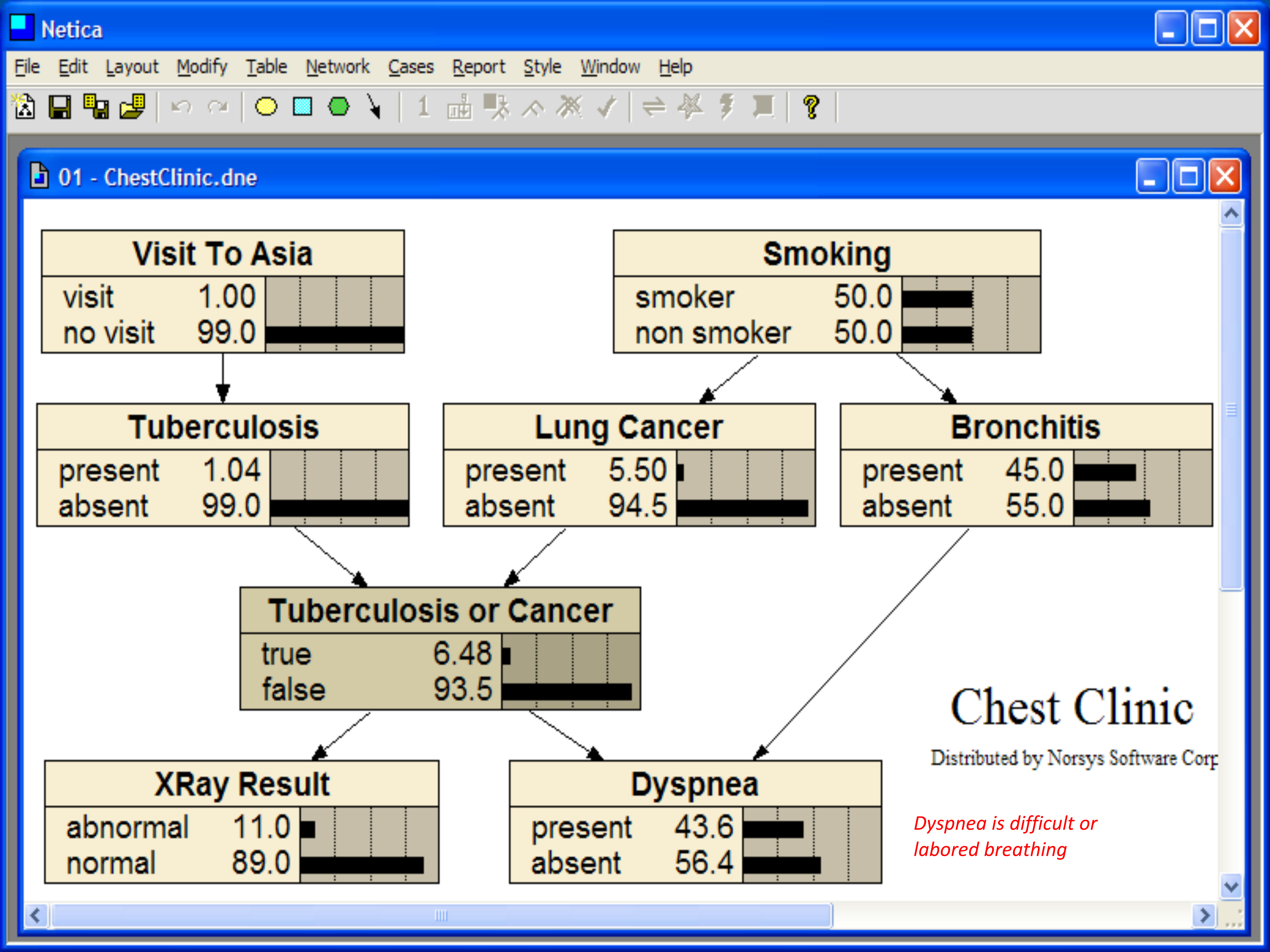
Example: Can I sleep in the next time it snows/is school canceled?

- $\mathcal{X} = \{x_1, x_2, \dots, x_N\}$ 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*?

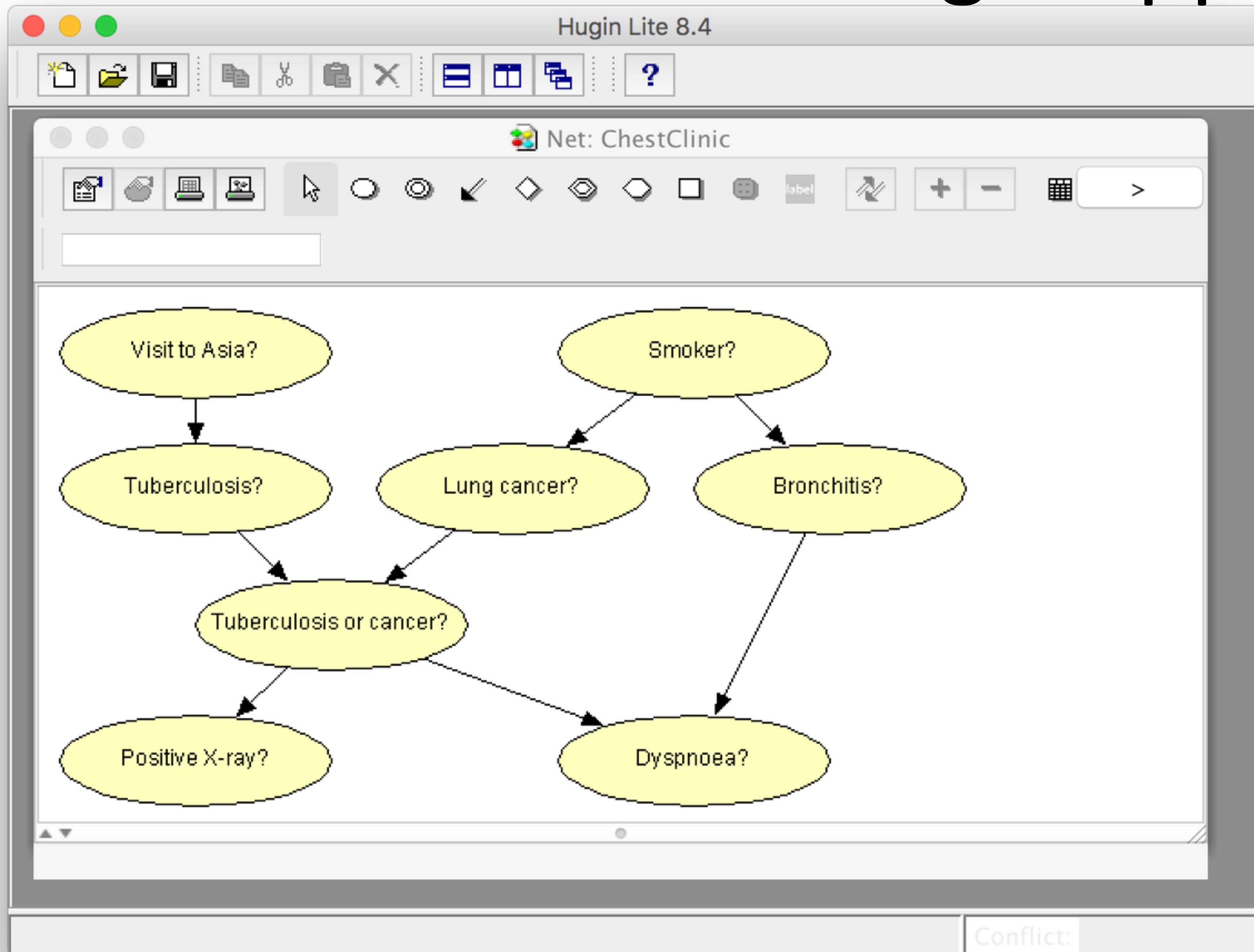
Some software tools

- 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

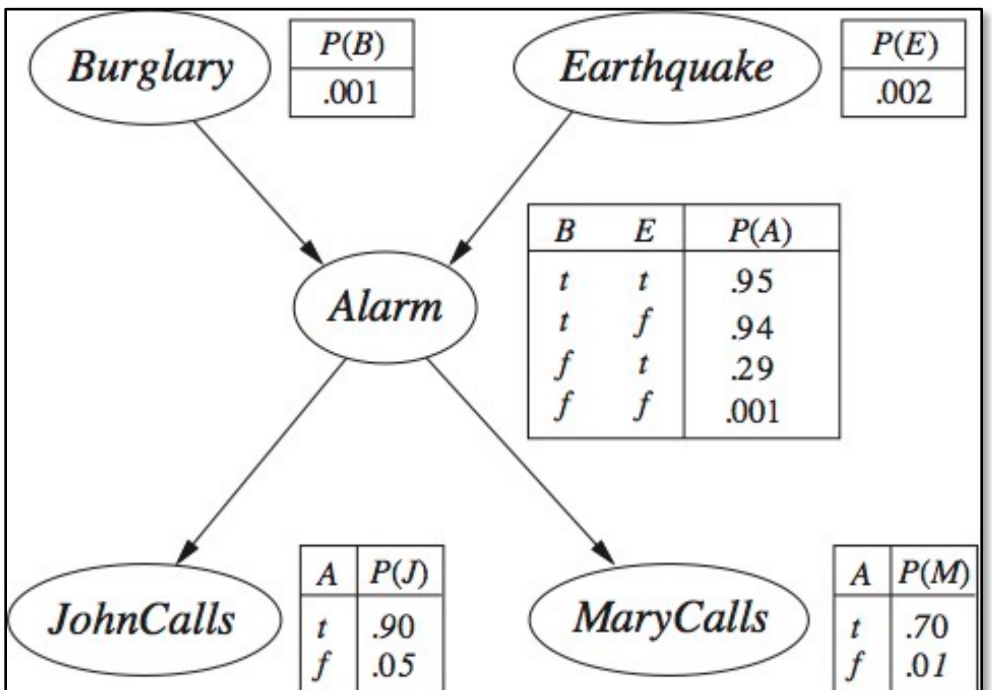
Same BBN model in Hugin app



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.