# Artificial Intelligence (CS303)

## Lecture 12: Representing and Inference with Uncertainty

# Hints for this lecture

- **An agent can seldom precisely knows the state, knowledge should be represented such that wise decisions/actions can still be made.**

# I. Uncertainty and Rational Decisions

# The World is Uncertain

- Things don’t always happen with simple true or false.
- We never know what “state” we are in exactly, because the world is only partially observable.
- We (agents) seldom make decisions with full certainty, while more often make **rational** decision based on **utility**.

| Language | Ontological Commitment | Epistemological Commitment |
|---|---|---|
| Propositional logic | facts | true/false/unknown |
| First-order logic | facts, objects, relations | true/false/unknown |
| Probability theory | facts | degree of belief |
| Fuzzy logic | facts | degree of truth known interval value |

# Alternative to Logic

- Utility theory: Assign utility to each state/actions
- Probability theory: Summarize the uncertainty associated with each state
- Rational Decisions: Maximize the expected utility (Probability + Utility)
- Thus we need to represent states in the language of probability.

# II. Basic Probability Theory and Its Use

# Basic Probability Theory and Its Use

- Joint probability distribution specifies probability of every atomic event.

- Queries can be answered by summing over atomic events.

| | *toothache* | | ¬ *toothache* | |
|---|---|---|---|---|
| | *catch* | ¬ *catch* | *catch* | ¬ *catch* |
| *cavity* | .108 | .012 | .072 | .008 |
| ¬ *cavity* | .016 | .064 | .144 | .576 |

# Basic Probability Theory and Its Use

## Prior probability

Prior or unconditional probabilities of propositions
e.g., $P(Cavity = true) = 0.1$ and $P(Weather = sunny) = 0.72$
correspond to belief prior to arrival of any (new) evidence

Probability distribution gives values for all possible assignments:
$\mathbf{P}(Weather) = \langle 0.72, 0.1, 0.08, 0.1 \rangle$ (normalized, i.e., sums to 1)

Joint probability distribution for a set of r.v.s gives the
probability of every atomic event on those r.v.s (i.e., every sample point)
$\mathbf{P}(Weather, Cavity) =$ a $4 \times 2$ matrix of values:

| $Weather =$ | $sunny$ | $rain$ | $cloudy$ | $snow$ |
|---|---|---|---|---|
| $Cavity = true$ | 0.144 | 0.02 | 0.016 | 0.02 |
| $Cavity = false$ | 0.576 | 0.08 | 0.064 | 0.08 |

**Every question about a domain can be answered by the joint distribution because every event is a sum of sample points**

# Basic Probability Theory and Its Use

## Conditional probability

Conditional or posterior probabilities

e.g., $P(cavity|toothache) = 0.8$

i.e., **given that** *toothache* **is all I know**

**NOT** "if *toothache* then 80% chance of *cavity*"

(Notation for conditional distributions:

$\mathbf{P}(Cavity|Toothache)$ = 2-element vector of 2-element vectors)

If we know more, e.g., *cavity* is also given, then we have

$P(cavity|toothache, cavity) = 1$

Note: the less specific belief **remains valid** after more evidence arrives, but is not always **useful**

New evidence may be irrelevant, allowing simplification, e.g.,

$P(cavity|toothache, 49ersWin) = P(cavity|toothache) = 0.8$

This kind of inference, sanctioned by domain knowledge, is crucial

# Basic Probability Theory and Its Use

## Conditional probability

Definition of conditional probability:

$$P(a|b) = \frac{P(a \wedge b)}{P(b)} \text{ if } P(b) \neq 0$$

Product rule gives an alternative formulation:

$$P(a \wedge b) = P(a|b)P(b) = P(b|a)P(a)$$

A general version holds for whole distributions, e.g.,

$$\mathbf{P}(Weather, Cavity) = \mathbf{P}(Weather|Cavity)\mathbf{P}(Cavity)$$

(View as a $4 \times 2$ set of equations, **not** matrix mult.)

Chain rule is derived by successive application of product rule:

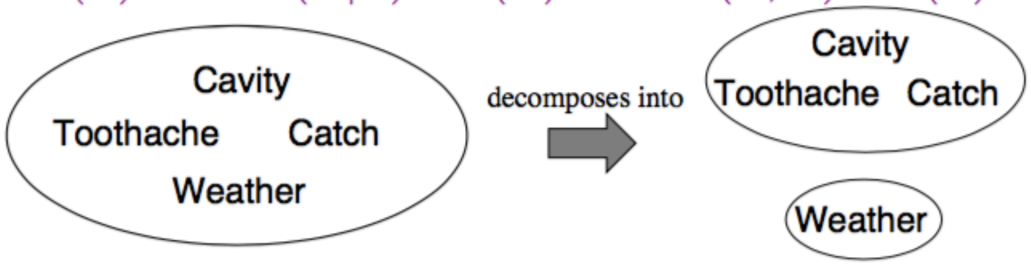
$$\begin{aligned}
\mathbf{P}(X_1, \ldots, X_n) &= \mathbf{P}(X_1, \ldots, X_{n-1})\, \mathbf{P}(X_n|X_1, \ldots, X_{n-1}) \\
&= \mathbf{P}(X_1, \ldots, X_{n-2})\, \mathbf{P}(X_{n_1}|X_1, \ldots, X_{n-2})\, \mathbf{P}(X_n|X_1, \ldots, X_{n-1}) \\
&= \ldots \\
&= \prod_{i=1}^{n} \mathbf{P}(X_i|X_1, \ldots, X_{i-1})
\end{aligned}$$

# Basic Probability Theory and Its Use

## Independence

$A$ and $B$ are independent iff

$\mathbf{P}(A|B) = \mathbf{P}(A)$ or $\mathbf{P}(B|A) = \mathbf{P}(B)$ or $\mathbf{P}(A, B) = \mathbf{P}(A)\mathbf{P}(B)$




$$\mathbf{P}(Toothache, Catch, Cavity, Weather)$$
$$= \mathbf{P}(Toothache, Catch, Cavity)\mathbf{P}(Weather)$$

32 entries reduced to 12; for $n$ independent biased coins, $2^n \to n$

Absolute independence powerful but rare

Dentistry is a large field with hundreds of variables,
none of which are independent. What to do?

# Basic Probability Theory and Its Use

## Bayes' Rule

Product rule $P(a \wedge b) = P(a|b)P(b) = P(b|a)P(a)$

$\Rightarrow$ Bayes' rule $P(a|b) = \dfrac{P(b|a)P(a)}{P(b)}$

or in distribution form

$$\mathbf{P}(Y|X) = \frac{\mathbf{P}(X|Y)\mathbf{P}(Y)}{\mathbf{P}(X)} = \alpha\mathbf{P}(X|Y)\mathbf{P}(Y)$$

Useful for assessing diagnostic probability from causal probability:

$$P(Cause|Effect) = \frac{P(Effect|Cause)P(Cause)}{P(Effect)}$$

# Basic Probability Theory and Its Use

**Bayes' Rule and conditional independence**

$$\mathbf{P}(Cavity|toothache \wedge catch)$$
$$= \alpha\,\mathbf{P}(toothache \wedge catch|Cavity)\mathbf{P}(Cavity)$$
$$= \alpha\,\mathbf{P}(toothache|Cavity)\mathbf{P}(catch|Cavity)\mathbf{P}(Cavity)$$

This is an example of a naive Bayes model:

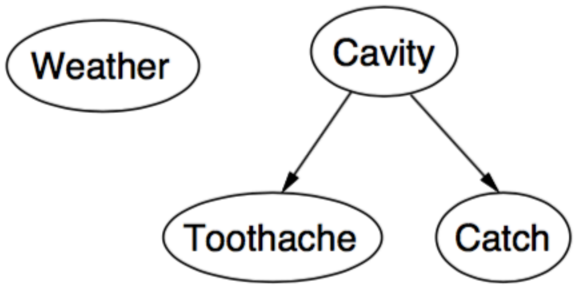
$$\mathbf{P}(Cause, Effect_1, \ldots, Effect_n) = \mathbf{P}(Cause)\Pi_i\mathbf{P}(Effect_i|Cause)$$


Total number of parameters is **linear** in $n$

# III. Bayesian Networks

# What is a BN?

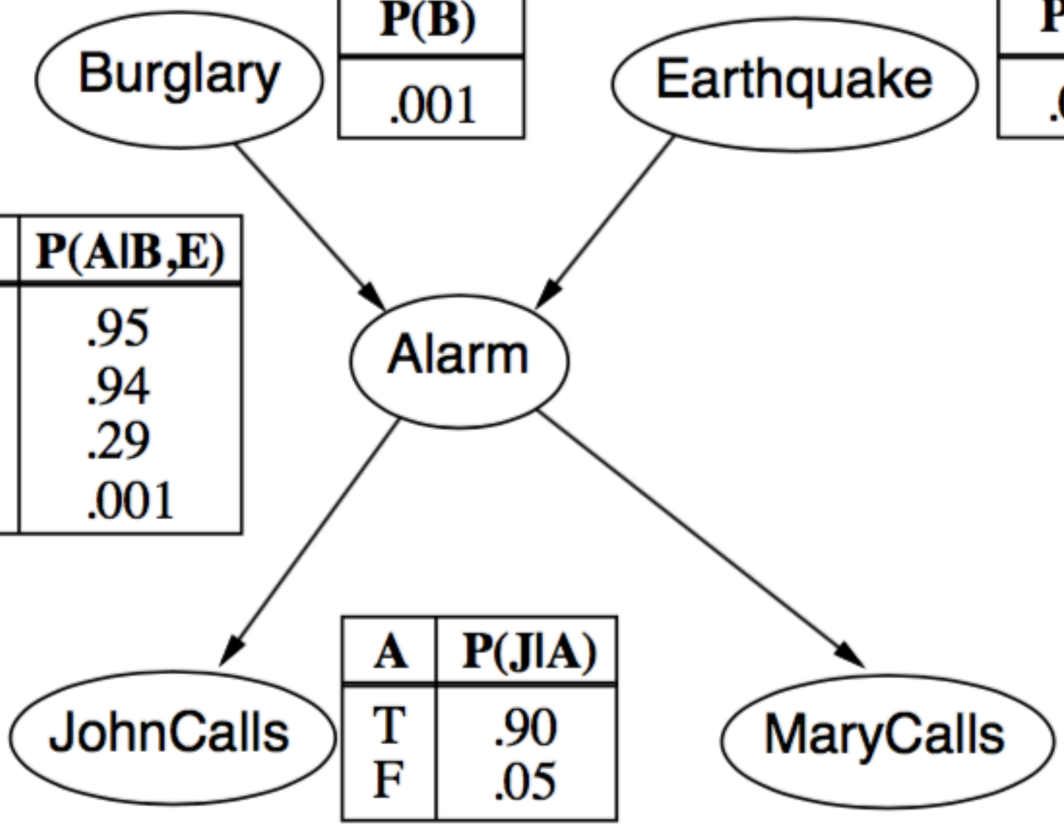
- A Directed Acyclic Graph (DAG).
- Each node is a random variable, associated with conditional distribution.
- Each arc (link) represent **direct influence** of a parent node to a child node.




$Weather$ is independent of the other variables

$Toothache$ and $Catch$ are conditionally independent given $Cavity$

# A Simple Example of BN




| P(B) |
|---|
| .001 |

| P(E) |
|---|
| .002 |

| B | E | P(A\|B,E) |
|---|---|---|
| T | T | .95 |
| T | F | .94 |
| F | T | .29 |
| F | F | .001 |

| A | P(J\|A) |
|---|---|
| T | .90 |
| F | .05 |

| A | P(M\|A) |
|---|---|
| T | .70 |
| F | .01 |

# BN in the form of Conditional Probability Table

- More compact representation (compared to propositional logic).

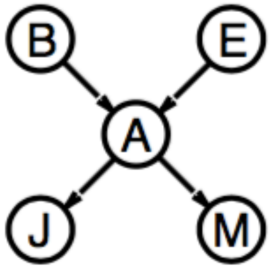
A CPT for Boolean $X_i$ with $k$ Boolean parents has
$2^k$ rows for the combinations of parent values

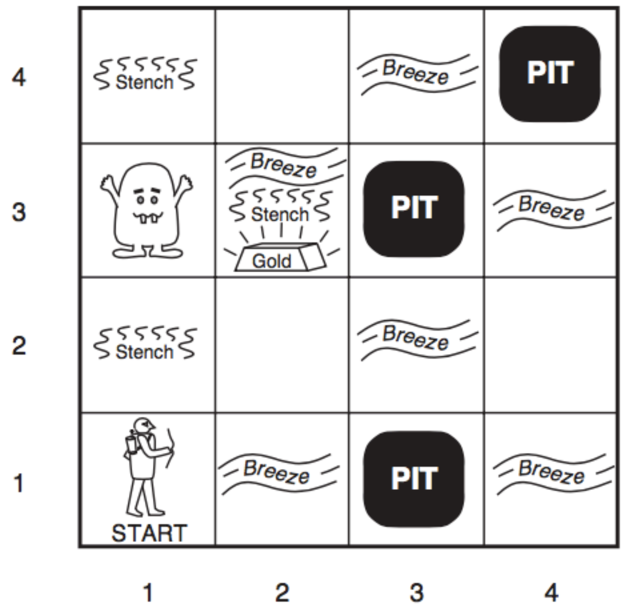
Each row requires one number $p$ for $X_i = true$
(the number for $X_i = false$ is just $1 - p$)

If each variable has no more than $k$ parents,
the complete network requires $O(n \cdot 2^k)$ numbers

I.e., grows linearly with $n$, vs. $O(2^n)$ for the full joint distribution

For burglary net, $1 + 1 + 4 + 2 + 2 = 10$ numbers (vs. $2^5 - 1 = 31$)




- Easier to utilize independence and conditional dependence relations to define the joint distribution.

# How to construct a CPT for BN?

1. Choose an ordering of variables $X_1, \ldots, X_n$
2. For $i = 1$ to $n$
   add $X_i$ to the network
   select parents from $X_1, \ldots, X_{i-1}$ such that
   $$\mathbf{P}(X_i | Parents(X_i)) = \mathbf{P}(X_i | X_1, \ldots, X_{i-1})$$

This choice of parents guarantees the global semantics:

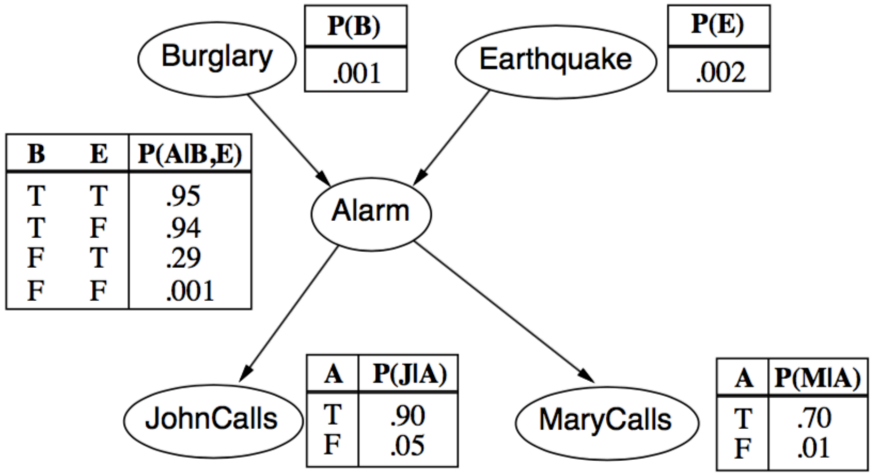
$$\begin{aligned} \mathbf{P}(X_1, \ldots, X_n) &= \Pi_{i=1}^{n} \mathbf{P}(X_i | X_1, \ldots, X_{i-1}) \quad \text{(chain rule)} \\ &= \Pi_{i=1}^{n} \mathbf{P}(X_i | Parents(X_i)) \quad \text{(by construction)} \end{aligned}$$

# IV. Exact Inference with Bayesian Networks

# Inference with BN

- Given a Bayesian Network, and an (or some) observed events, which specifies the value for evidence variables, we want to know the probability distribution of one (or several) query variables X, P (X | events).

- E. g. : $\mathbf{P}(Burglary \mid JohnCalls = true, MaryCalls = true)$

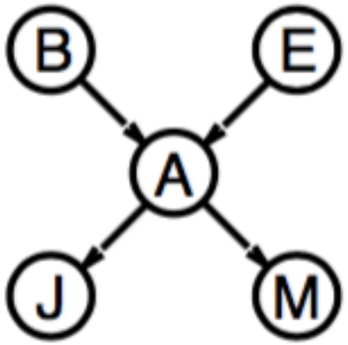


# Enumeration

Simple query on the burglary network:

$\mathbf{P}(B|j,m)$
$= \mathbf{P}(B,j,m)/P(j,m)$
$= \alpha\mathbf{P}(B,j,m)$
$= \alpha \sum_e \sum_a \mathbf{P}(B,e,a,j,m)$

Need to consider all values of "hidden variables", e.g., *alarm*=true, *alarm*=false



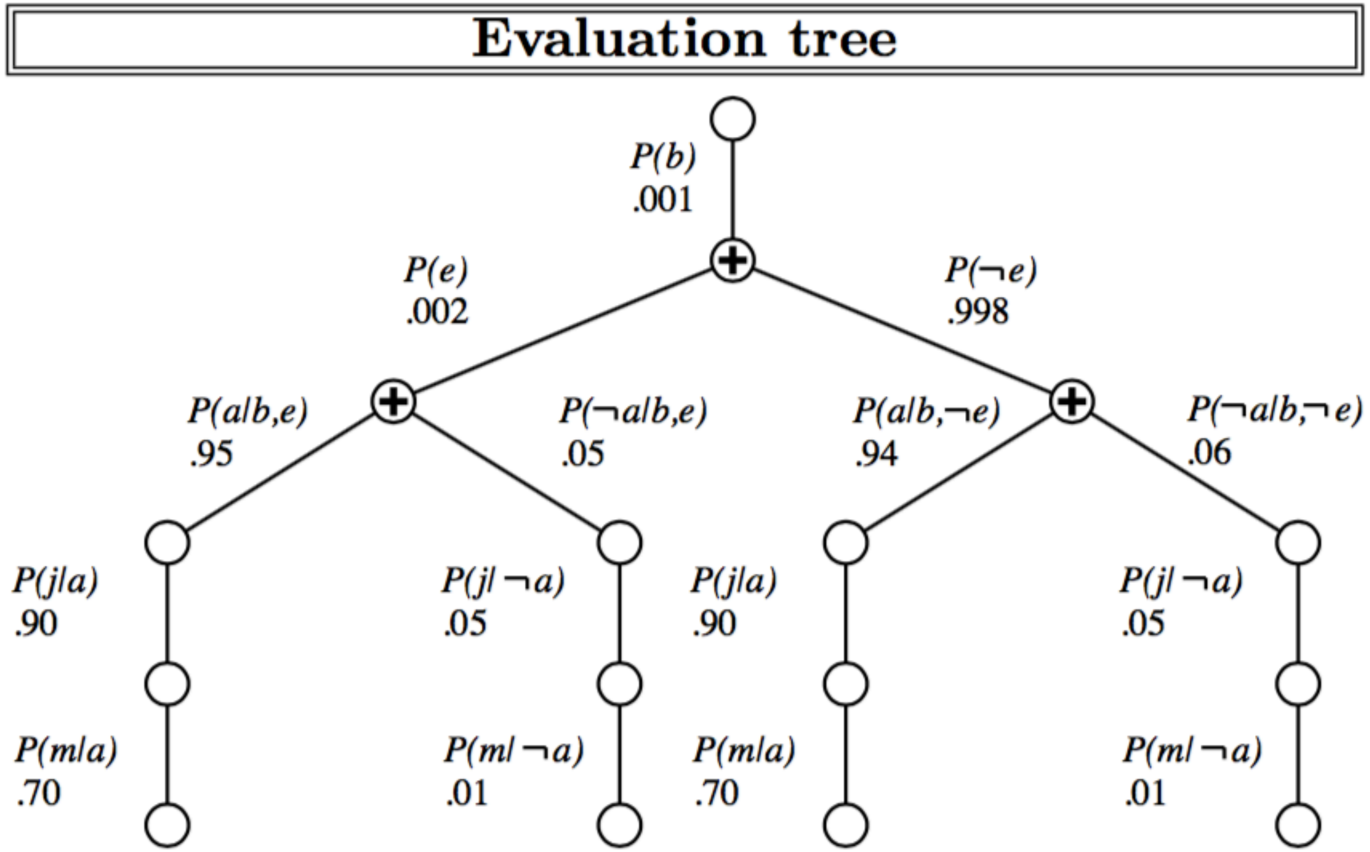

Rewrite full joint entries using product of CPT entries:

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

Recursive depth-first enumeration: $O(n)$ space, $O(d^n)$ time

d=2 for Boolean variables

# Enumeration




Enumeration is inefficient: repeated computation

e.g., computes $P(j|a)P(m|a)$ for each value of $e$

# Enumeration by Variable Elimination

Variable elimination: carry out summations right-to-left, storing intermediate results (factors) to avoid recomputation

$$
\begin{aligned}
&\mathbf{P}(B|j,m) \\
&\quad = \alpha \underbrace{\mathbf{P}(B)}_{B} \Sigma_e \underbrace{P(e)}_{E} \Sigma_a \underbrace{\mathbf{P}(a|B,e)}_{A} \underbrace{P(j|a)}_{J} \underbrace{P(m|a)}_{M} \\
&\quad = \alpha \mathbf{P}(B) \Sigma_e P(e) \Sigma_a \mathbf{P}(a|B,e) P(j|a) f_M(a) \\
&\quad = \alpha \mathbf{P}(B) \Sigma_e P(e) \Sigma_a \mathbf{P}(a|B,e) f_J(a) f_M(a) \\
&\quad = \alpha \mathbf{P}(B) \Sigma_e P(e) \Sigma_a f_A(a,b,e) f_J(a) f_M(a) \\
&\quad = \alpha \mathbf{P}(B) \Sigma_e P(e) f_{\bar{A}JM}(b,e) \text{ (sum out } A\text{)} \\
&\quad = \alpha \mathbf{P}(B) f_{\bar{E}\bar{A}JM}(b) \text{ (sum out } E\text{)} \\
&\quad = \alpha f_B(b) \times f_{\bar{E}\bar{A}JM}(b)
\end{aligned}
$$

# V. Approximate Inference with Bayesian Networks

# Basic Idea

- **Sampling/Monte Carlo/Stochastic Simulation…**

Basic idea:

1) Draw $N$ samples from a sampling distribution $S$
2) Compute an approximate posterior probability $\hat{P}$
3) Show this converges to the true probability $P$


Outline:

- Sampling from an empty network
- Rejection sampling: reject samples disagreeing with evidence
- Likelihood weighting: use evidence to weight samples
- Markov chain Monte Carlo (MCMC): sample from a stochastic process whose stationary distribution is the true posterior

# Sampling from an empty network

**function** PRIOR-SAMPLE($bn$) **returns** an event sampled from $bn$
  **inputs**: $bn$, a belief network specifying joint distribution $\mathbf{P}(X_1, \ldots, X_n)$

  $\mathbf{x} \leftarrow$ an event with $n$ elements
  **for** $i = 1$ **to** $n$ **do**
    $x_i \leftarrow$ a random sample from $\mathbf{P}(X_i \mid parents(X_i))$
      given the values of $Parents(X_i)$ in $\mathbf{x}$
  **return** $\mathbf{x}$

Assume the joint distribution could be easily sampled

# Rejection Sampling

$\hat{\mathbf{P}}(X|\mathbf{e})$ estimated from samples agreeing with $\mathbf{e}$

```
function REJECTION-SAMPLING(X, e, bn, N) returns an estimate of P(X|e)
   local variables: N, a vector of counts over X, initially zero

   for j = 1 to N do
      x ← PRIOR-SAMPLE(bn)
      if x is consistent with e then
         N[x] ← N[x]+1 where x is the value of X in x
   return NORMALIZE(N[X])
```

In case distribution of variable *e* is difficult to sample

E.g., estimate $\mathbf{P}(Rain|Sprinkler = true)$ using 100 samples
27 samples have $Sprinkler = true$
Of these, 8 have $Rain = true$ and 19 have $Rain = false$.

$\hat{\mathbf{P}}(Rain|Sprinkler = true) = \text{NORMALIZE}(\langle 8, 19 \rangle) = \langle 0.296, 0.704 \rangle$

Similar to a basic real-world empirical estimation procedure

$$\begin{aligned}
\hat{\mathbf{P}}(X|\mathbf{e}) &= \alpha \mathbf{N}_{PS}(X, \mathbf{e}) && \text{(algorithm defn.)} \\
&= \mathbf{N}_{PS}(X, \mathbf{e})/N_{PS}(\mathbf{e}) && \text{(normalized by } N_{PS}(\mathbf{e})\text{)} \\
&\approx \mathbf{P}(X, \mathbf{e})/P(\mathbf{e}) && \text{(property of PRIORSAMPLE)} \\
&= \mathbf{P}(X|\mathbf{e}) && \text{(defn. of conditional probability)}
\end{aligned}$$

Hence rejection sampling returns consistent posterior estimates

Problem: hopelessly expensive if $P(\mathbf{e})$ is small

$P(\mathbf{e})$ drops off exponentially with number of evidence variables!

# Likelihood Weighting

Fix the evidence variables to reduce the sampling space

Idea: fix evidence variables, sample only nonevidence variables, and weight each sample by the likelihood it accords the evidence

**function** LIKELIHOOD-WEIGHTING($X$, $\mathbf{e}$, $bn$, $N$) **returns** an estimate of $P(X|\mathbf{e})$
  **local variables**: $\mathbf{W}$, a vector of weighted counts over $X$, initially zero

  **for** $j = 1$ to $N$ **do**
    $\mathbf{x}, w \leftarrow$ WEIGHTED-SAMPLE($bn$)
    $\mathbf{W}[x] \leftarrow \mathbf{W}[x] + w$ where $x$ is the value of $X$ in $\mathbf{x}$
  **return** NORMALIZE($\mathbf{W}[X]$)

---

**function** WEIGHTED-SAMPLE($bn$, $\mathbf{e}$) **returns** an event and a weight

  $\mathbf{x} \leftarrow$ an event with $n$ elements; $w \leftarrow 1$
  **for** $i = 1$ **to** $n$ **do**
    **if** $X_i$ has a value $x_i$ in $\mathbf{e}$
      **then** $w \leftarrow w \times P(X_i = x_i \mid parents(X_i))$
      **else** $x_i \leftarrow$ a random sample from $\mathbf{P}(X_i \mid parents(X_i))$
  **return** $\mathbf{x}$, $w$

# Markov Chain Monte Carlo (MCMC)

"State" of network = current assignment to all variables.

Generate next state by sampling one variable given Markov blanket
Sample each variable in turn, keeping evidence fixed

**function** MCMC-ASK($X$, $\mathbf{e}$, $bn$, $N$) **returns** an estimate of $P(X|\mathbf{e})$
  **local variables**: $\mathbf{N}[X]$, a vector of counts over $X$, initially zero
    $\mathbf{Z}$, the nonevidence variables in $bn$
    $\mathbf{x}$, the current state of the network, initially copied from $\mathbf{e}$

  initialize $\mathbf{x}$ with random values for the variables in $\mathbf{Y}$
  **for** $j = 1$ to $N$ **do**
    **for each** $Z_i$ in $\mathbf{Z}$ **do**
      sample the value of $Z_i$ in $\mathbf{x}$ from $\mathbf{P}(Z_i|mb(Z_i))$
        given the values of $MB(Z_i)$ in $\mathbf{x}$
      $\mathbf{N}[x] \leftarrow \mathbf{N}[x] + 1$ where $x$ is the value of $X$ in $\mathbf{x}$
  **return** NORMALIZE($\mathbf{N}[X]$)

Markov Blanket

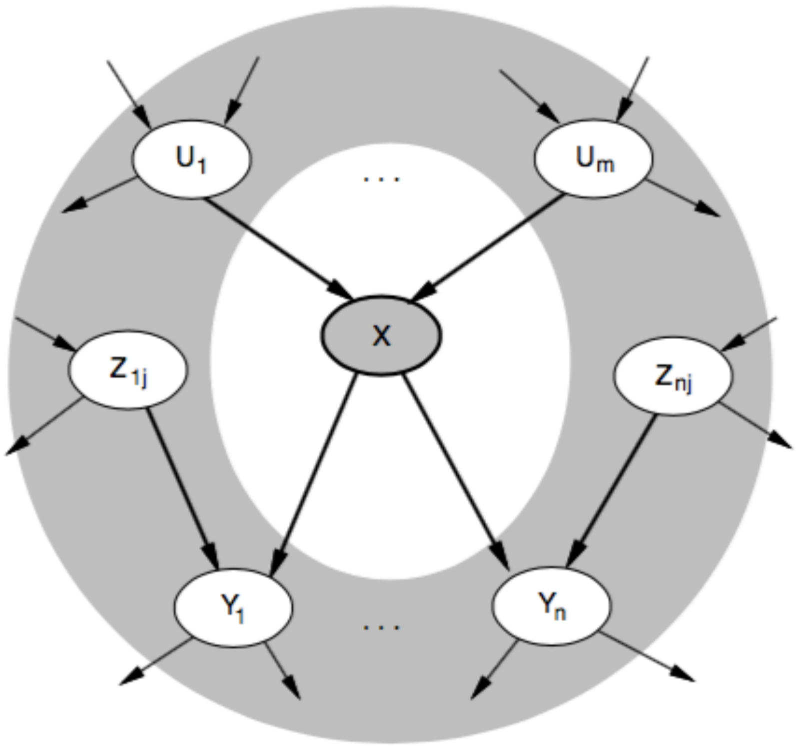
Further reduce the sampling space by only considering variables in Markov blanket

Can also choose a variable to sample at random each time

# Markov Chain Monte Carlo (MCMC)

## Markov blanket

Each node is conditionally independent of all others given its Markov blanket: parents + children + children's parents



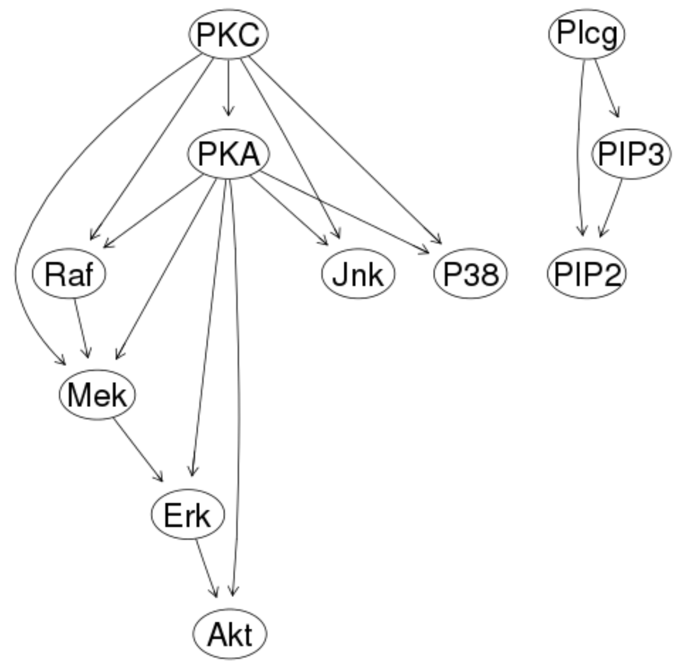
# VI. How to construct a BN (or KB in general)

# The challenge brought by KB

- Manual implement a KB is tedious (sometimes unaffordable)
- Can we obtain it **automatically** from raw data?

| 1 | Raf | Mek | Plcg | PIP2 | PIP3 | Erk | Akt | PKA | PKC | P38 | Jnk |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 2 | 26.4 | 13.2 | 8.82 | 18.3 | 58.8 | 6.61 | 17 | 414 | 17 | 44.9 | 40 |
| 3 | 35.9 | 16.5 | 12.3 | 16.8 | 8.13 | 18.6 | 32.5 | 352 | 3.37 | 16.5 | 61.5 |
| 4 | 59.4 | 44.1 | 14.6 | 10.2 | 13 | 14.9 | 32.5 | 403 | 11.4 | 31.9 | 19.5 |
| 5 | 73 | 82.8 | 23.1 | 13.5 | 1.29 | 5.83 | 11.8 | 528 | 13.7 | 28.6 | 23.1 |
| 6 | 33.7 | 19.8 | 5.19 | 9.73 | 24.8 | 21.1 | 46.1 | 305 | 4.66 | 25.7 | 81.3 |
| 7 | 18.8 | 3.75 | 17.6 | 22.1 | 10.9 | 11.9 | 25.7 | 610 | 13.7 | 49.1 | 57.8 |
| 8 | 44.9 | 36.5 | 10.4 | 132 | 16.3 | 8.66 | 17.9 | 835 | 15 | 35.9 | 18.1 |
| 9 | 47.4 | 15 | 14.6 | 30.5 | 17.5 | 20.2 | 45.3 | 466 | 6.44 | 24.4 | 20 |
| 10 | 104 | 61.5 | 10.6 | 21.1 | 41.8 | 11.5 | 23.5 | 445 | 29.2 | 61 | 25.3 |
| 11 | 21.1 | 21.5 | 1.88 | 205 | 43.7 | 13.2 | 135 | 213 | 14.6 | 26.7 | 101 |
| 12 | 16.4 | 16.4 | 14.5 | 17 | 11.2 | 21.9 | 34.6 | 449 | 20.4 | 44.9 | 24.1 |
| 13 | 74.3 | 22.9 | 7.5 | 15.5 | 26.2 | 20.9 | 36.5 | 389 | 31.9 | 71 | 35.5 |
| 14 | 85.1 | 39.6 | 8.9 | 64.9 | 11.7 | 6.67 | 12.2 | 528 | 17.9 | 44.1 | 118 |
| 15 | 36.8 | 29.2 | 5 | 9.06 | 15.5 | 17.9 | 17.9 | 400 | 14.6 | 41.4 | 151 |
| 16 | 29.2 | 24.1 | 10.2 | 16 | 46.6 | 8.82 | 7.23 | 500 | 9.73 | 19.1 | 7.91 |
| 17 | 50 | 13.8 | 11.9 | 13.2 | 11.3 | 18.1 | 27.9 | 392 | 56.2 | 77 | 1 |
| 18 | 26.2 | 26.7 | 21.3 | 10.9 | 14.7 | 9.06 | 37.9 | 89 | 40 | 65.5 | 1.42 |
| 19 | 39.6 | 38.2 | 10.5 | 92.2 | 22.7 | 27.6 | 31.3 | 223 | 16.5 | 30.8 | 7.64 |
| 20 | 30.5 | 19.8 | 7.5 | 133 | 15.7 | 19.1 | 36.2 | 319 | 24.1 | 37.2 | 17.2 |
| 21 | 49.1 | 16.5 | 13.3 | 141 | 25.7 | 31.1 | 62.1 | 710 | 14.2 | 27.4 | 22.1 |

Raw data

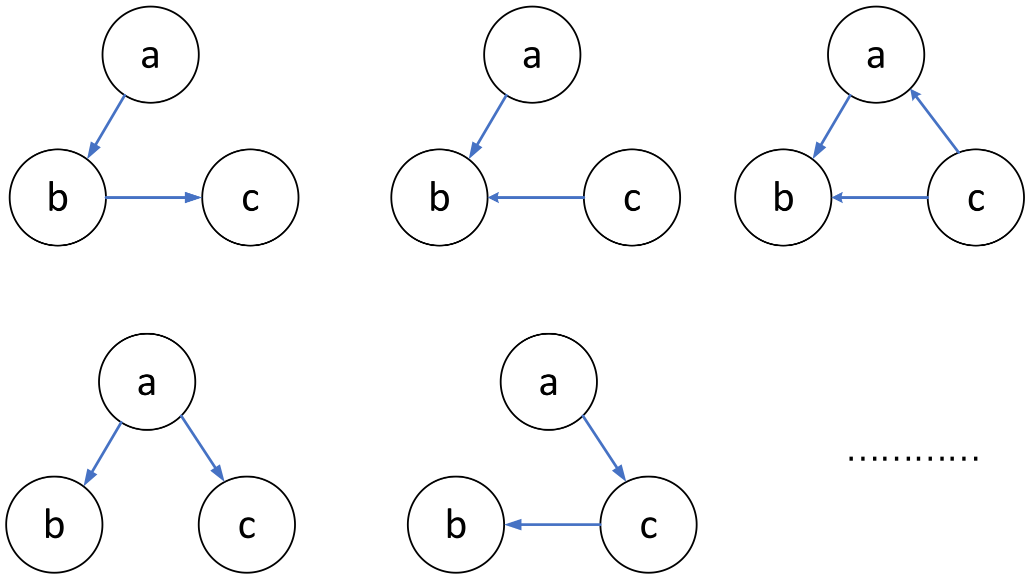



BN

# Constructing a BN from data

- Structural Learning
- Parameter Estimation

| 1 | a | b | c |
|---|---|---|---|
| 2 | 0.366329 | 0.458928 | 0.937441 |
| 3 | 0.492245 | 0.829342 | 0.657999 |
| 4 | 0.521917 | 0.119465 | 0.956325 |
| 5 | 0.741545 | 0.490508 | 0.063304 |
| 6 | 0.60749 | 0.785813 | 0.62433 |
| 7 | 0.935561 | 0.542337 | 0.464547 |
| 8 | 0.771979 | 0.501432 | 0.25861 |
| 9 | 0.505258 | 0.398753 | 0.8417 |
| 10 | 0.147234 | 0.44209 | 0.986153 |
| 11 | 0.250739 | 0.646287 | 0.167685 |
| 12 | 0.149013 | 0.907165 | 0.162593 |
| 13 | 0.127727 | 0.451099 | 0.49503 |
| 14 | 0.382873 | 0.629639 | 0.016354 |




# Similar to Neural Networks

- Structural Learning: Identify the network structure
- Parameter Estimation: find VALUEs for parameters associated with an edge
    - Depending on how you define the relationship between events/nodes
        - values in a CPT
        - parameters of a probability density function

- A machine learning or search problem again.

To be continued