Artificial Intelligence

Lecture 7：Bayesian Network I

Xiaojin Gong

2022-04-02

Credits: AI Course in Berkeley

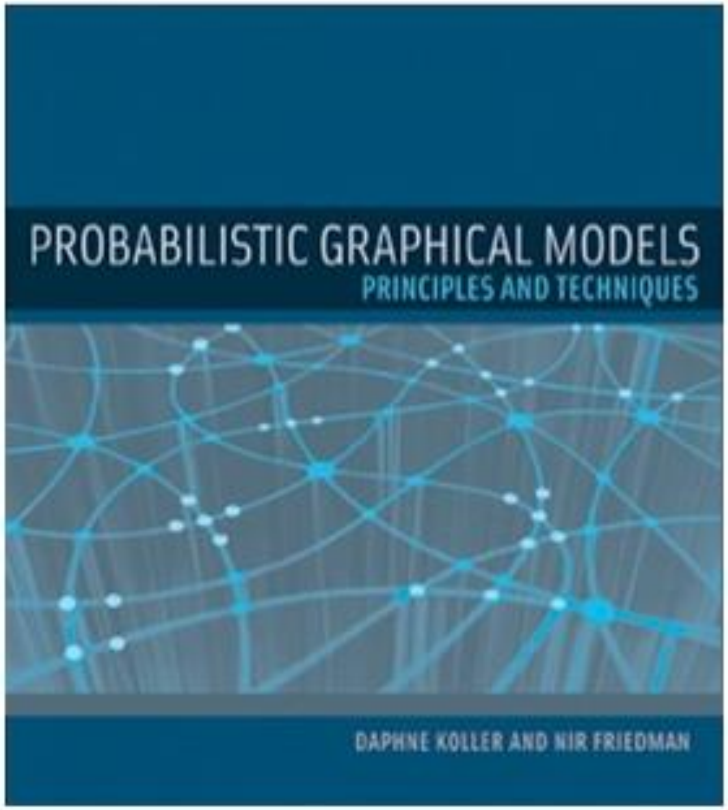
Review

- Probability
    - Random variables
    - Joint and marginal distributions
    - Conditional distribution
    - Product rule, chain rule, Bayes' rule
    - Inference
    - Independence and conditional independence

Outline

- Bayesian network
    - Representation
        - Joint probability
        - Conditional independence
    - Inference
        - Exact inference
            - Enumeration
            - Variable elimination
        - Approximate inference
            - Sampling

Reference Book

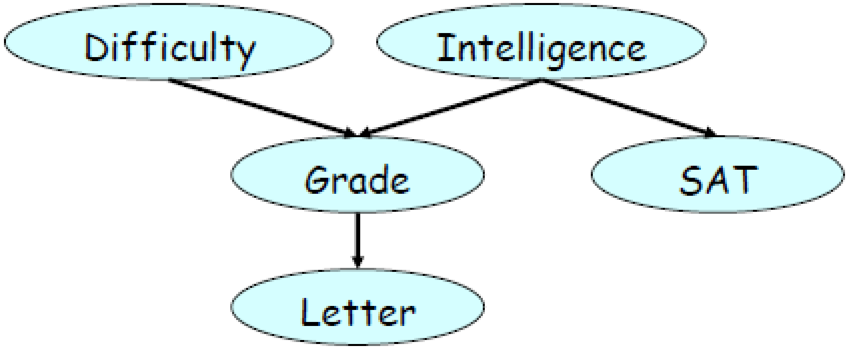

PROBABILISTIC GRAPHICAL MODELS
PRINCIPLES AND TECHNIQUES
DAPHNE KOLLER AND NIR FRIEDMAN

Daphne Koller
Computer Science Dept.
Stanford University

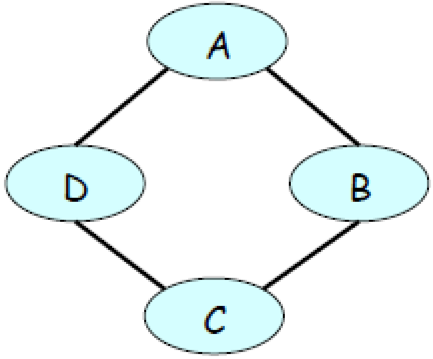
Nir Friedman
School of Computer Science &
Engineering
Hebrew University

Probabilistic Graphical Models

- Bayesian network



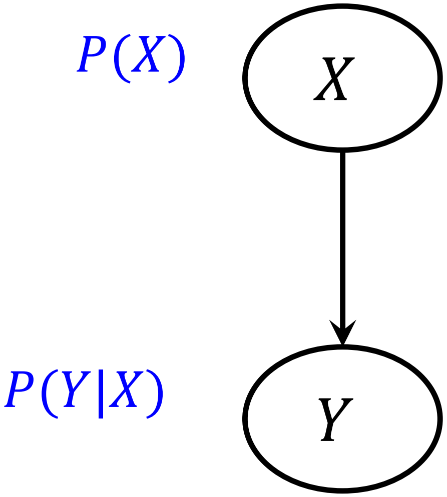
- Markov network



Bayesian Network

- A Bayesian network is
  - A directed acyclic graph (DAG)
  - Each node corresponds to a random variable X_i
  - Each node X_i has a conditional probability distribution (CPD) $P(X_i|Parents(X_i))$

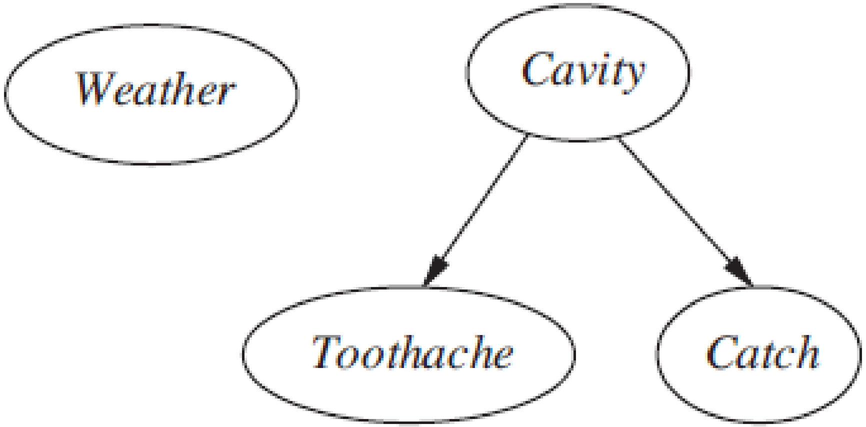
A Bayes net = Topology (graph) + Local Conditional Probabilities



- X is a parent of Y
- X has ad direct influence on Y
- X: cause, Y:effect

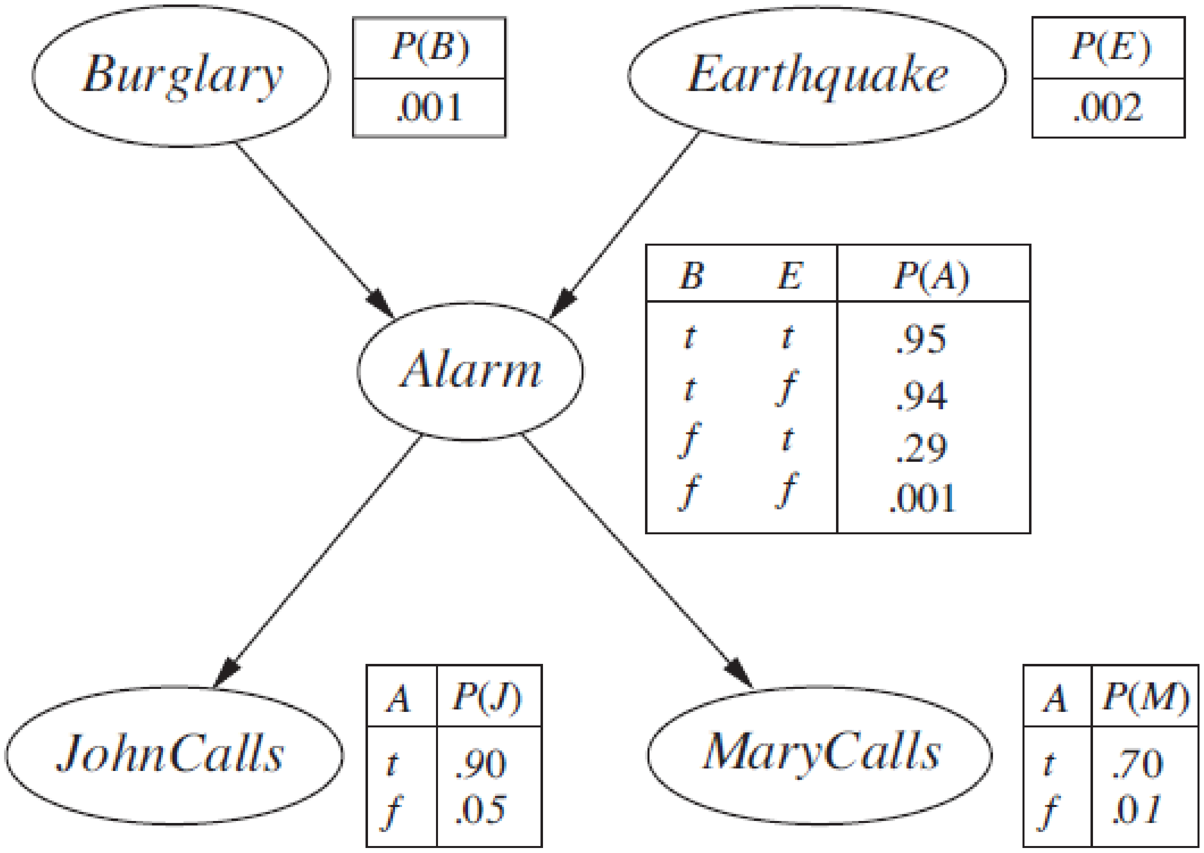
Bayesian Network

- Example 1: Diagnosing a dental patient’s toothache



Bayesian Network

- Example 2:
  - Conditional probability table (CPT)



P(B)
.001

P(E)
.002

B	E	P(A)
t	t	.95
t	f	.94
f	t	.29
f	f	.001

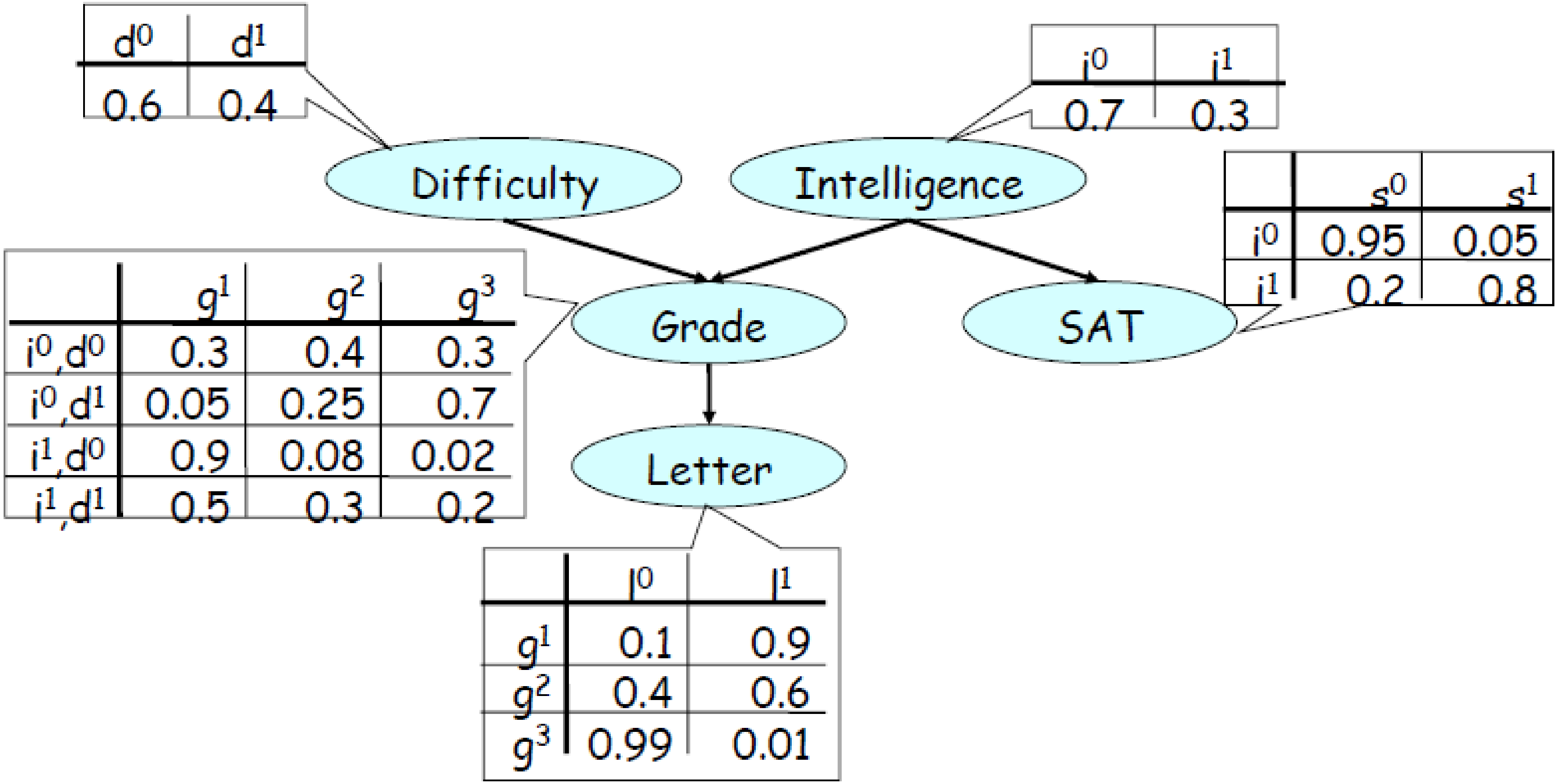
A	P(J)
t	.90
f	.05

A	P(M)
t	.70
f	.01

Bayesian Network

- Example 3:

- Conditional probability table (CPT)



d^0	d^1
0.6	0.4

i^0	i^1
0.7	0.3

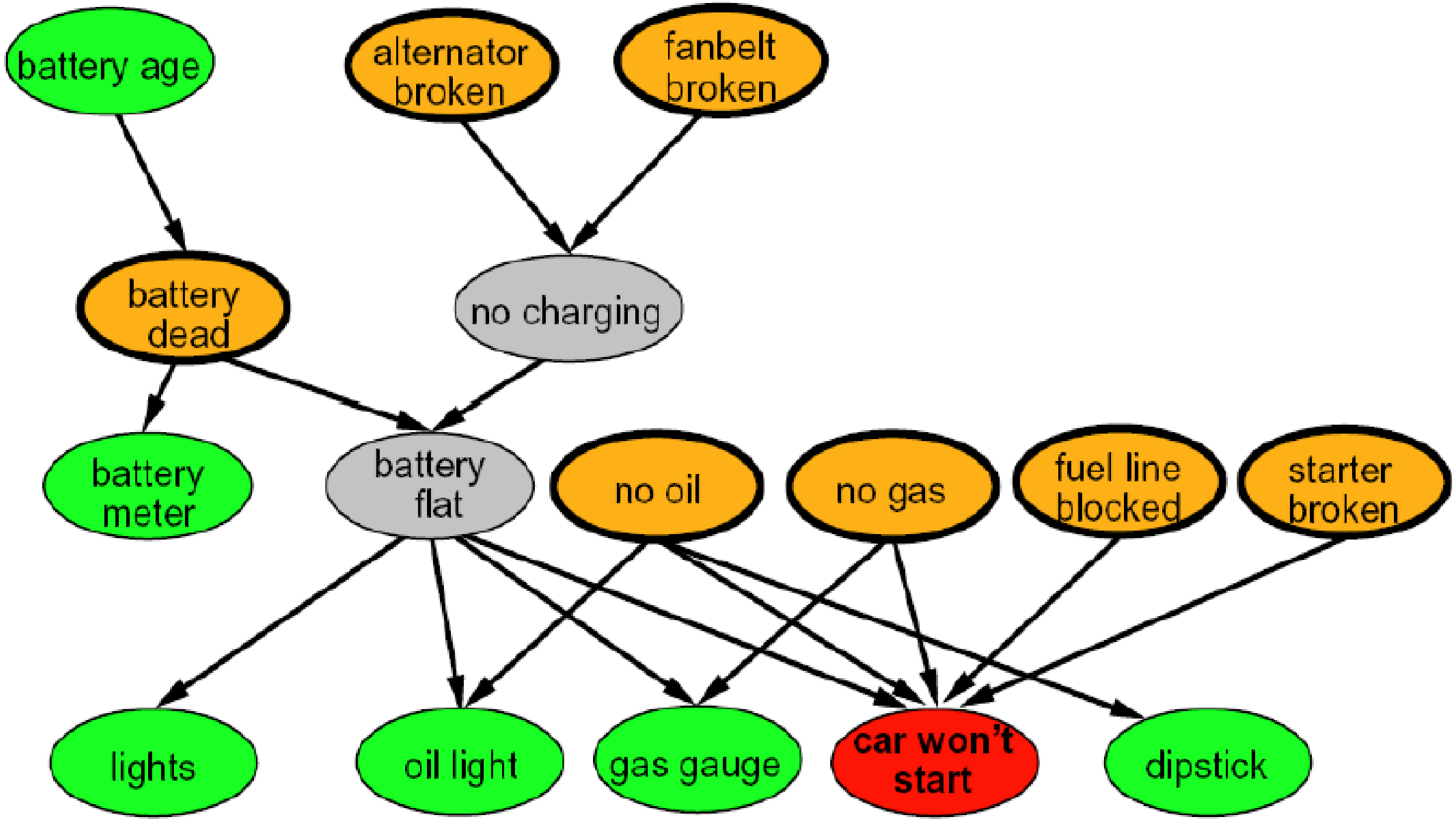
	s^0	s^1
i^0	0.95	0.05
i^1	0.2	0.8

	g^1	g^2	g^3
i^0,d^0	0.3	0.4	0.3
i^0,d^1	0.05	0.25	0.7
i^1,d^0	0.9	0.08	0.02
i^1,d^1	0.5	0.3	0.2

	l^0	l^1
g^1	0.1	0.9
g^2	0.4	0.6
g^3	0.99	0.01

Bayesian Network

- Example 4:
  - Initial evidence: car won’t start
  - Testable variables (green), “broken, so fix it” variables (orange)
  - Hidden variables (gray) ensure sparse structure, reduce parameters



Bayesian Network

- Syntax:
  - A BN is a directed acyclic graph with some numeric parameters attached to each node.
- Semantics:
  - A BN is a representation of the joint probability distribution.
  - A BN is an encoding of a collection of conditional independence statements.

Joint Distribution of BN

- The BN represents a joint distribution via the chain rule for Bayesian networks

$$P(X_1,\ X_2, \cdots, X_n) = \prod_{\mathrm{i}} P(X_i | Parents(X_i))$$

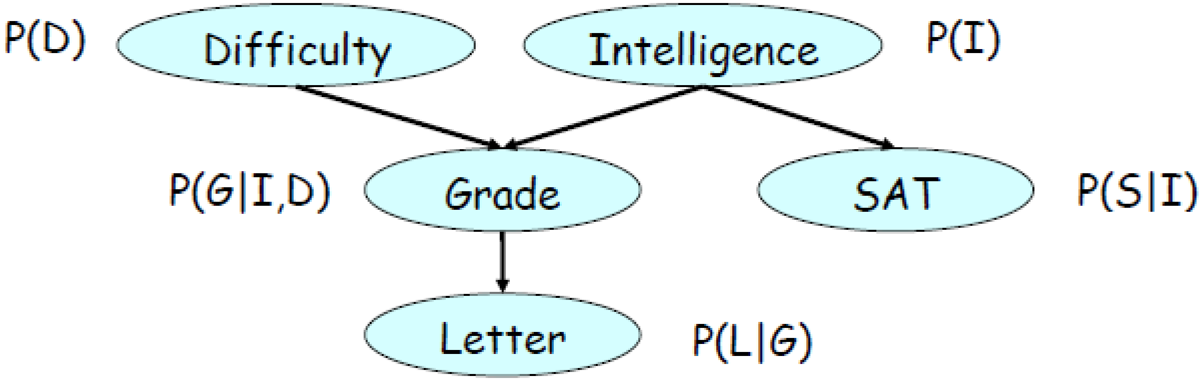
  - Assume conditional independences

$$P(X_i |\ X_1, \cdots, X_{i-1}) = P(X_i | Parents(X_i))$$

- BN is a legal distribution:
  - $P(X_1,\ X_2, \cdots, X_n) \geq 0$
  - $\sum_{X_1,\ X_2, \cdots, X_n} P(X_1,\ X_2, \cdots, X_n) = 1$

Joint Distribution of BN

- $P(D, I, G, S, L)$?

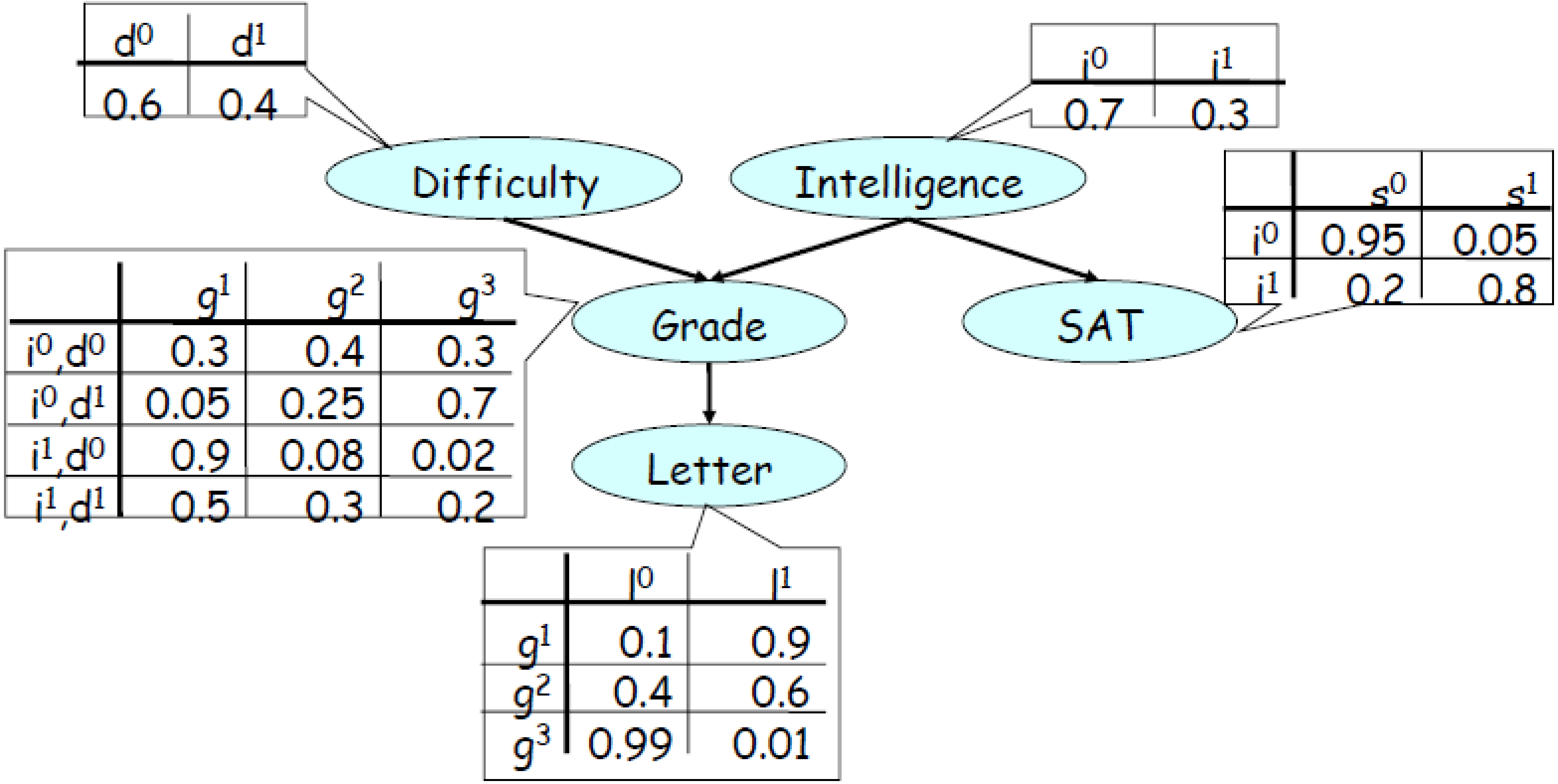


$$P(D, I, G, S, L) = P(D)P(I|D)P(G|D, I)P(S|D, I, G)P(L|S, D, I, G)$$

$$= P(D)P(I)P(G|D, I)P(S|I)P(L|G)$$

Joint Distribution of BN

- $P(d^0, i^1, g^3, s^1, l^1)$?



d^0	d^1
0.6	0.4

i^0	i^1
0.7	0.3

	g^1	g^2	g^3
i^0,d^0	0.3	0.4	0.3
i^0,d^1	0.05	0.25	0.7
i^1,d^0	0.9	0.08	0.02
i^1,d^1	0.5	0.3	0.2

	s^0	s^1
i^0	0.95	0.05
i^1	0.2	0.8

	l^0	l^1
g^1	0.1	0.9
g^2	0.4	0.6
g^3	0.99	0.01

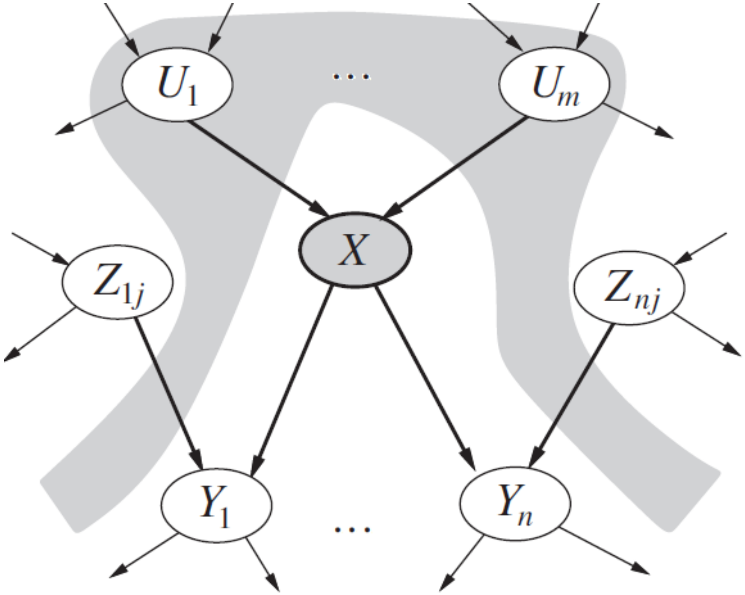
Conditional Independence in BN

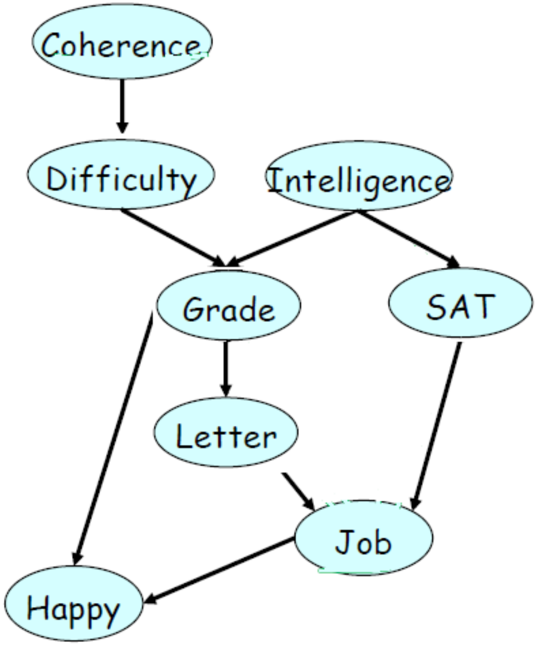
- Two variables are independent ($X \perp Y$) in a joint distribution if:
  - $P(X,Y) = P(X)P(Y)$
  - $\forall x,y \;\; P(x,y) = P(x)P(y)$
  - $P(x|y) = P(x)$
  - $P(y|x) = P(y)$

- Conditional independence: For (sets of) random variables X, Y, Z, ($X \perp Y|Z$) iff:
  - $P(X,Y|Z) = P(X|Z)P(Y|Z)$
  - $P(X|Y,Z) = P(X|Z)$
  - $P(Y|X,Z) = P(Y|Z)$
  - $\forall x,y,z \qquad P(x,y|z) = P(x|z)P(y|z)$

Conditional Independence in BN

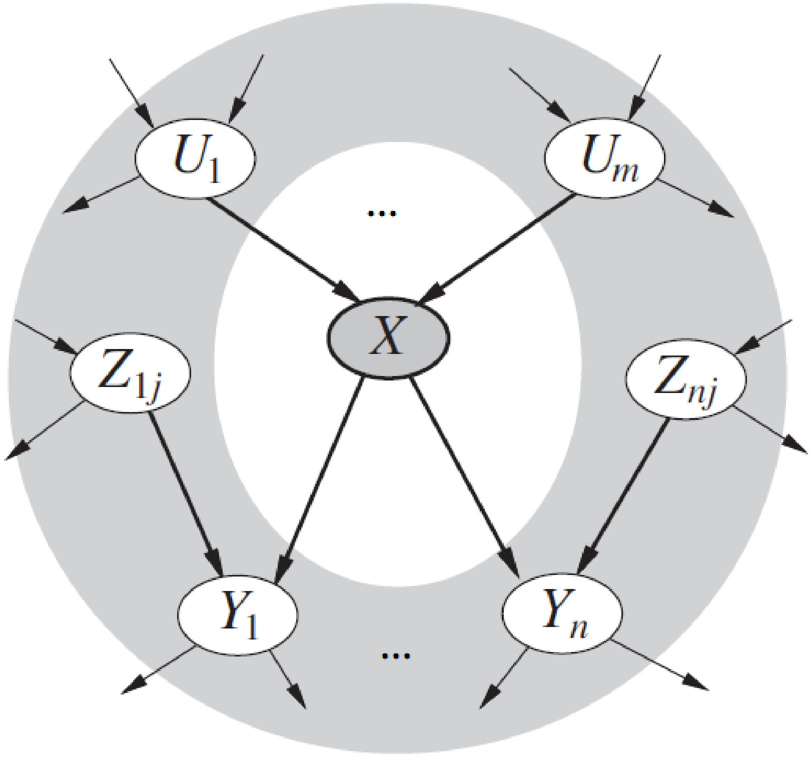
- Numerical semantics:
  - Each node is conditionally independent of its other predecessors, given its parents.
- Topological semantics:
  - Each node is conditionally independent of its non-descendants, given its parents.





Conditional Independence in BN

- Markov blanket:
  - A node is conditionally independent of all other nodes in the network, given its parents, children and children’s parents

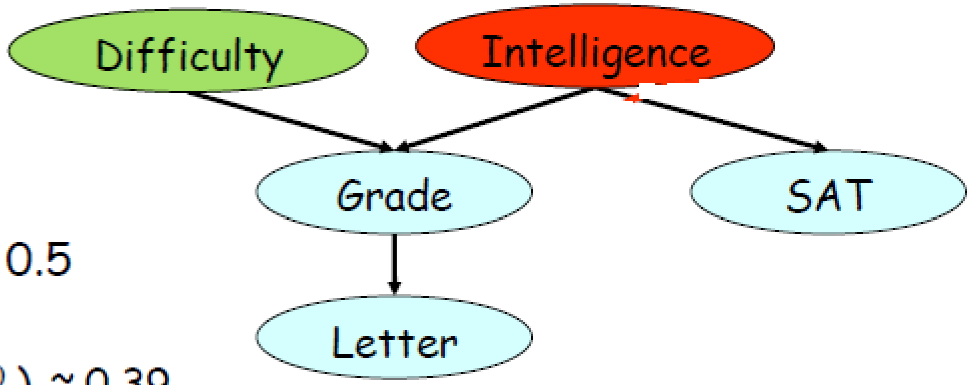


Conditional Independence in BN

- Reasoning Patterns：
  - Important questions about a BN:
    - Are two nodes independent?
    - Are two nodes independent given certain evidence?

Conditional Independence in BN

- Reasoning Patterns：
  - Causal reasoning

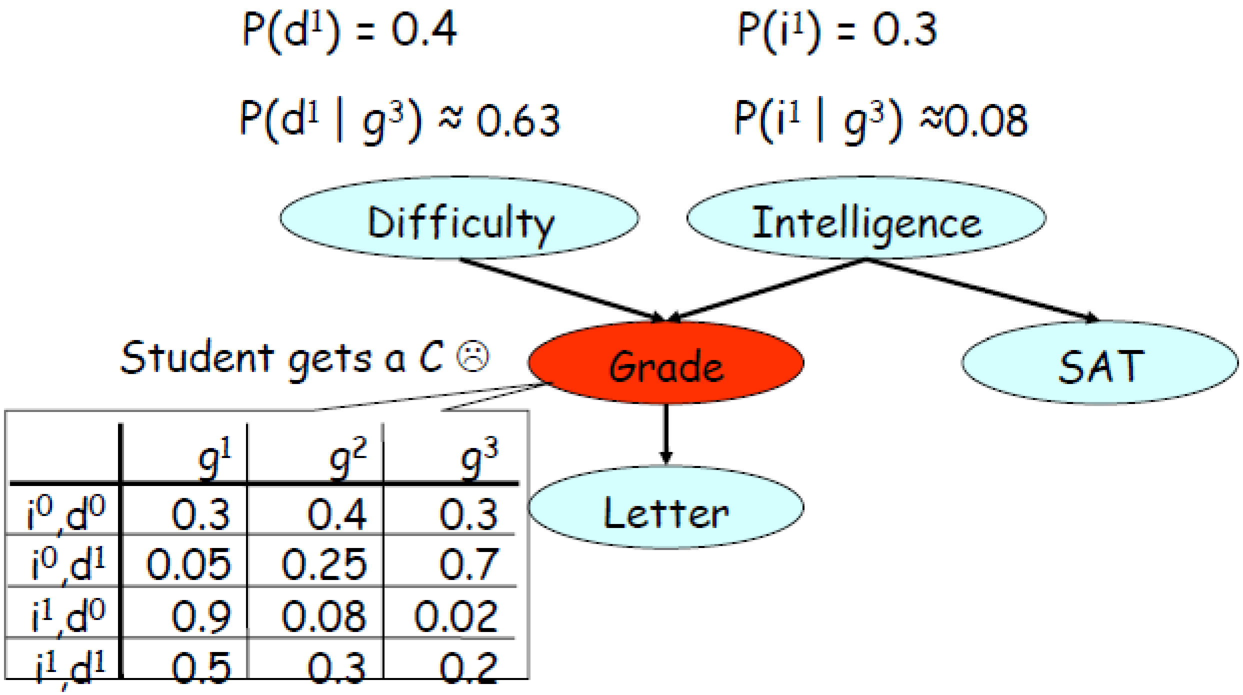


$P(l^1) \approx 0.5$

$P(l^1 \mid i^0) \approx 0.39$

Conditional Independence in BN

- Reasoning Patterns：
  - Evidence reasoning



$P(d^1) = 0.4$

$P(d^1 \mid g^3) \approx 0.63$

$P(i^1) = 0.3$

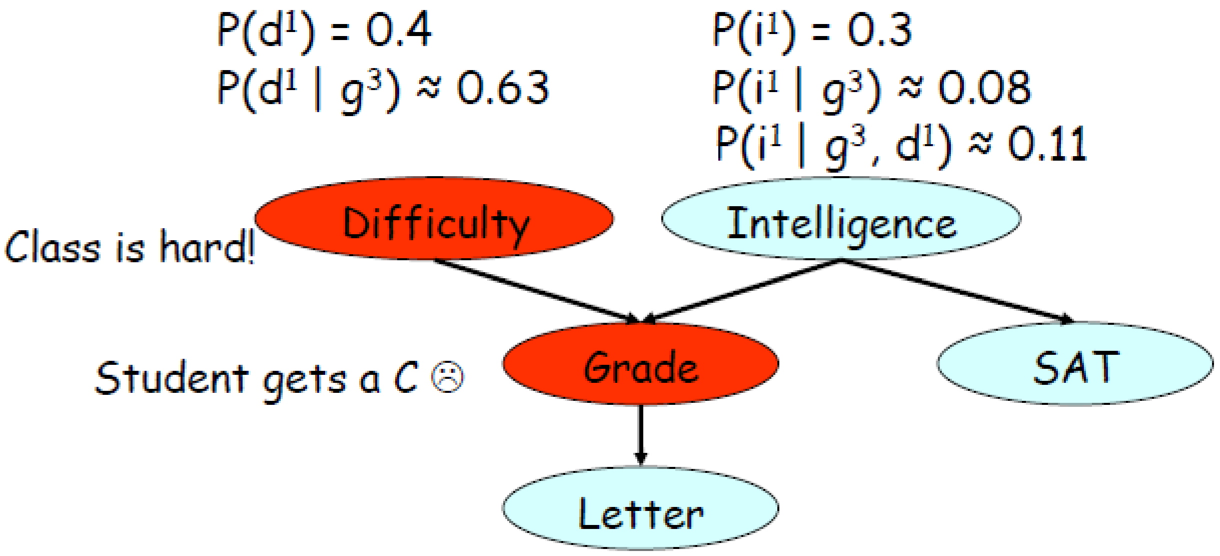
$P(i^1 \mid g^3) \approx 0.08$

Student gets a C ☹

	g^1	g^2	g^3
i^0,d^0	0.3	0.4	0.3
i^0,d^1	0.05	0.25	0.7
i^1,d^0	0.9	0.08	0.02
i^1,d^1	0.5	0.3	0.2

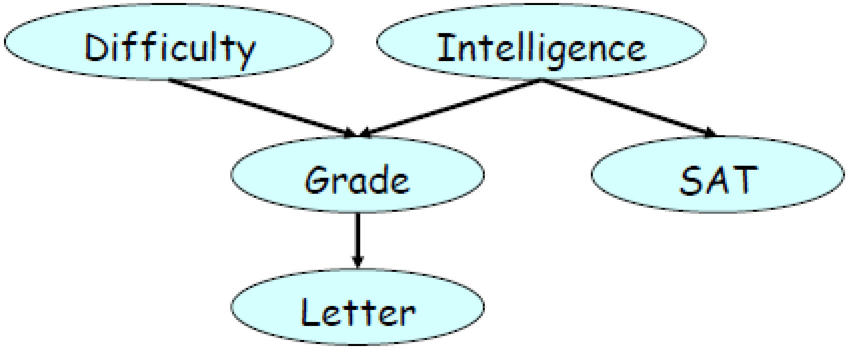
Conditional Independence in BN

- Reasoning Patterns：
  - Intercausal reasoning



Conditional Independence in BN

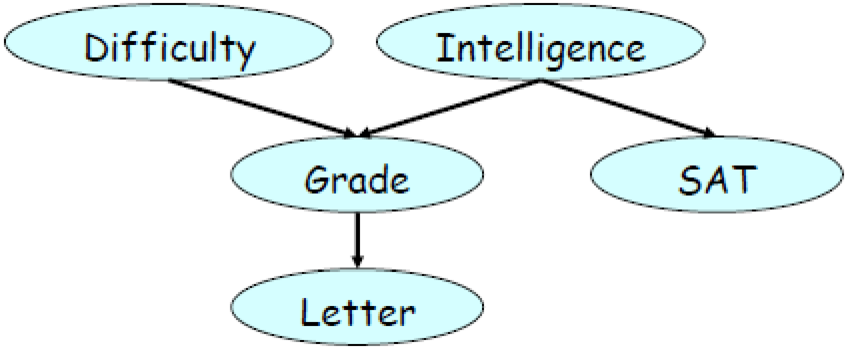
- Flow of Probabilistic Influence
  - When can X influence Y?
    - $X \rightarrow Y$ √
    - $X \leftarrow Y$ √
    - $X \rightarrow Z \rightarrow Y$ √
    - $X \leftarrow Z \leftarrow Y$ √
    - $X \leftarrow Z \rightarrow Y$ √
    - $X \rightarrow Z \leftarrow Y$ ×



  - Active trails
    - A trail $X_1 - X_2 - \cdots - X_n$ is active if it has no v-structures $X_{i-1} \rightarrow X_i \leftarrow X_{i+1}$

Conditional Independence in BN

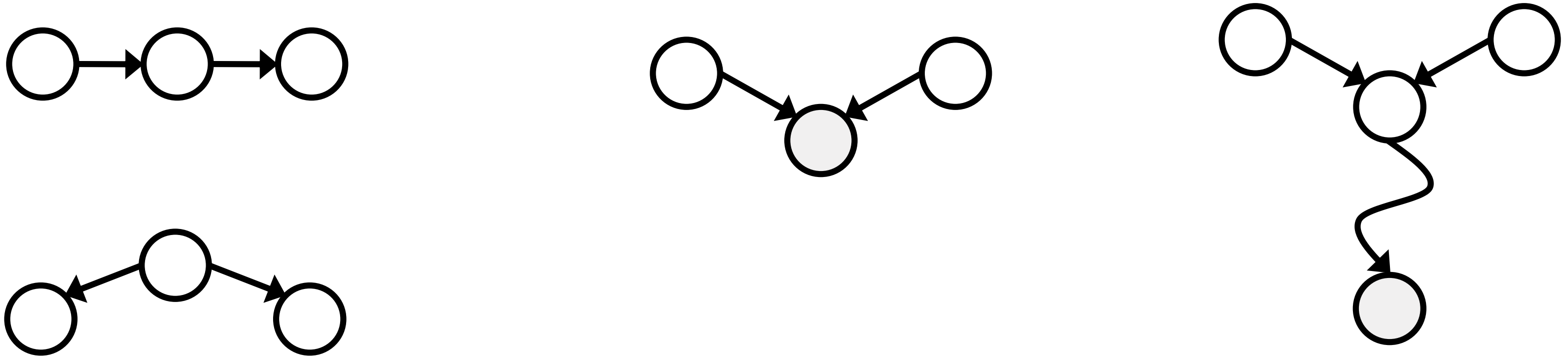
- Flow of Probabilistic Influence
  - When can X influence Y given Z?
    - $X \rightarrow Y$
    - $X \leftarrow Y$
    - $X \rightarrow Z \rightarrow Y$ ×
    - $X \leftarrow Z \leftarrow Y$ ×
    - $X \leftarrow Z \rightarrow Y$ ×
    - $X \rightarrow Z \leftarrow Y$ √



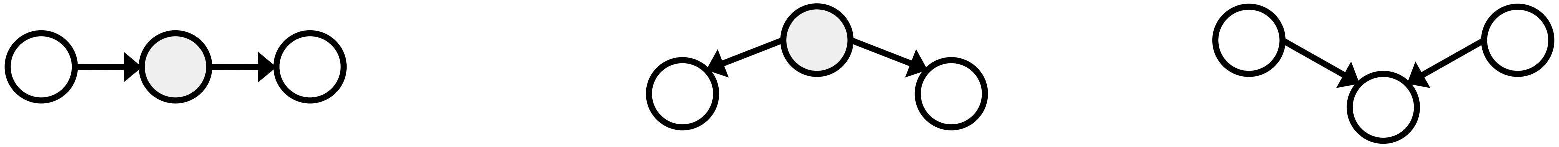
- Active trails
  - A trail $X_1 - X_2 - \cdots - X_n$ is active given Z if
    - for any v-structures $X_{i-1} \rightarrow X_i \leftarrow X_{i+1}$ we have that X_i or one of its descendants $\in Z$
    - No other $X_i \in Z$

Conditional Independence in BN

- Active trails

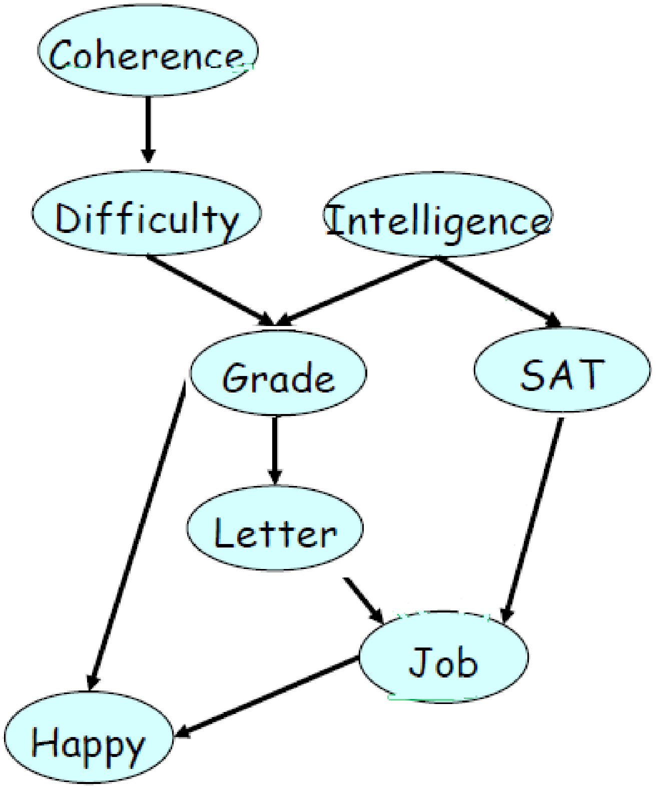


- Inactive trails



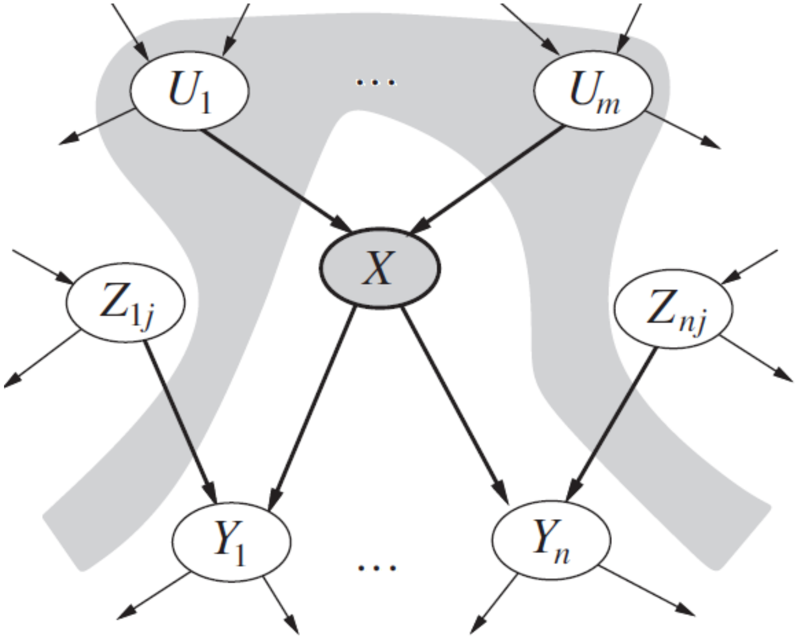
Conditional Independence in BN

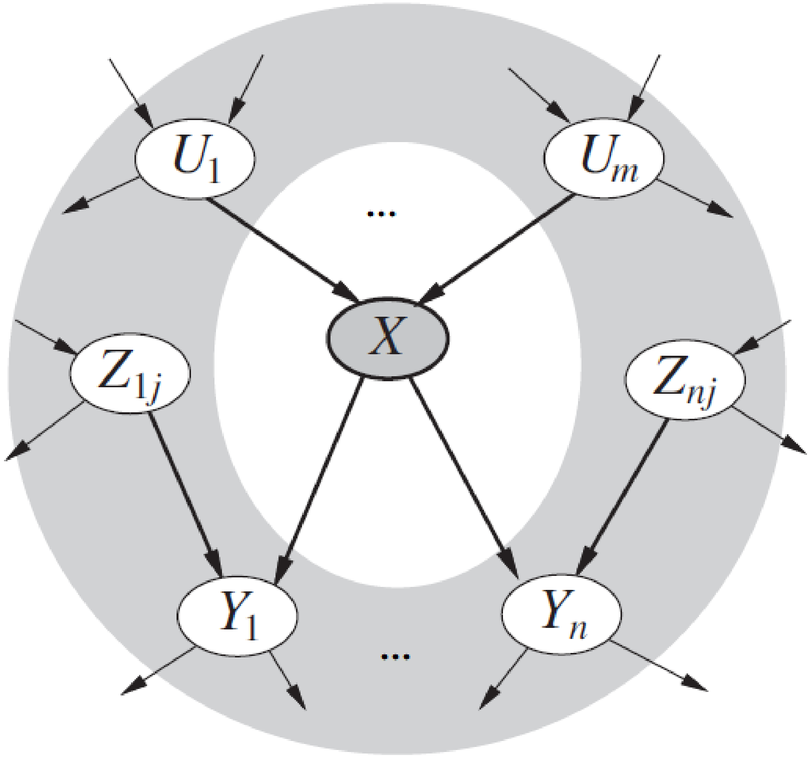
- D-separation
  - $d - sep_G(X_i, Xj|Z)$: X_i and X_j are d-separated in G given Z if there is no active trail in G between X and Y given Z.
  - Any node is d-separated from its non-descendants given its parents



Conditional Independence in BN

- Conditional independence in BN





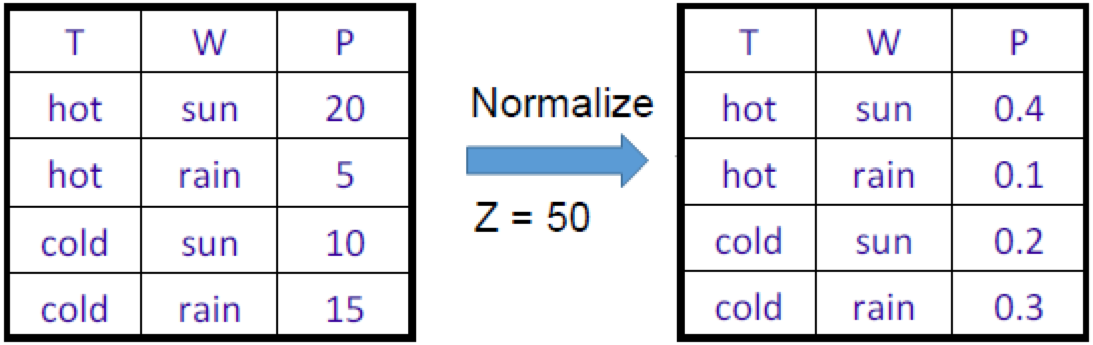
Probabilistic Inference

- Inference:
  - Calculating some useful quantity from a joint probability distribution
- Examples:
  - Posterior probability: $P(Q|E_1 = e_1, \ldots E_k = e_k)$
  - Most likely explanation: $\text{argmax}_q \; P(Q = q|E_1 = e_1 \ldots)$
- Methods:
  - Exact inference
    - Enumeration (exponential complexity)
    - Variable elimination (worst-case exponential complexity, often better)
  - Approximate inference
    - Sampling

Factors

- A factor Φ: $(X_1, X_2, \cdots, X_n) \rightarrow Val(X_1, X_2, \cdots, X_n) \in R$
- Scope = $\{X_1, X_2, \cdots, X_n\}$

$\Phi(T, W)$ $P(T, W)$



T	W	P
hot	sun	20
hot	rain	5
cold	sun	10
cold	rain	15

Normalize

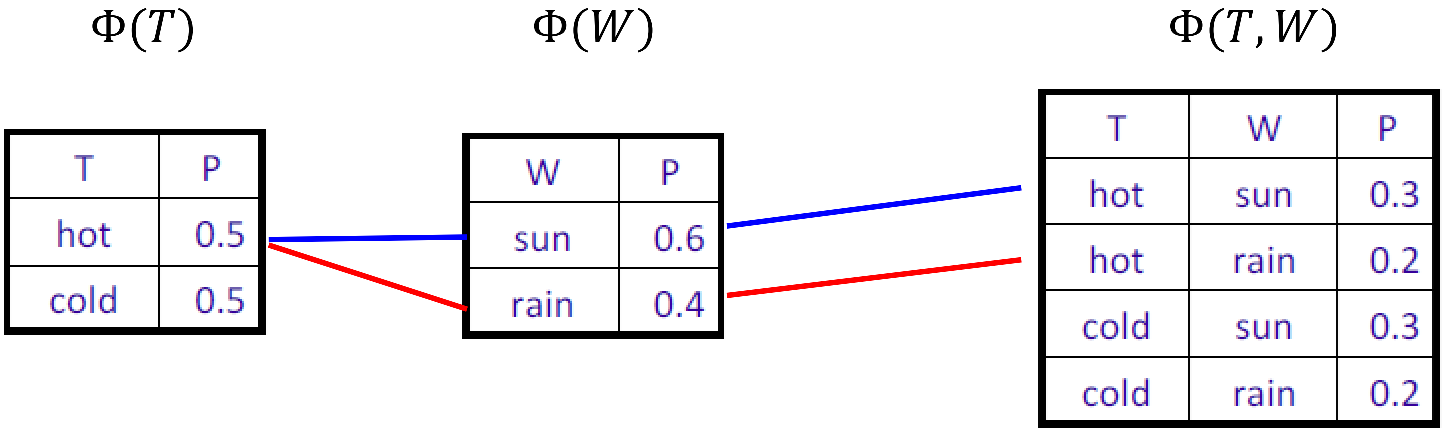
Z = 50

T	W	P
hot	sun	0.4
hot	rain	0.1
cold	sun	0.2
cold	rain	0.3

Factor Product

- $\Phi(X_1, X_2) = \Phi(X_1)\ \Phi(X_2)$

- $\Phi(X_1, X_2, X_3) = \Phi(X_1, X_2)\ \Phi(X_2, X_3)$



$\Phi(T)$

T	P
hot	0.5
cold	0.5

$\Phi(W)$

W	P
sun	0.6
rain	0.4

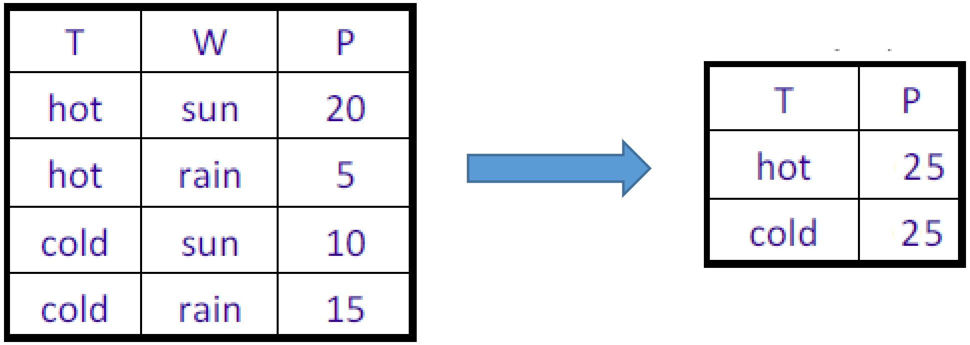
$\Phi(T, W)$

T	W	P
hot	sun	0.3
hot	rain	0.2
cold	sun	0.3
cold	rain	0.2

Factor Marginalization

- $\Phi(X_1) \ = \sum_{X_2} \Phi(X_1, X_2)$

$\Phi(T, W)$ $\Phi(T)$



T	W	P
hot	sun	20
hot	rain	5
cold	sun	10
cold	rain	15

T	P
hot	25
cold	25

Enumeration

- Given
  - Evidence variables: $E_1 \ldots E_k = e_1 \ldots e_k$
  - Query variable: Q
  - Hidden variables: $H_1 \ldots H_r$

$$\left.\begin{array}{l} E_1 \ldots E_k = e_1 \ldots e_k \\ Q \\ H_1 \ldots H_r \end{array}\right\} X_1, X_2, \ldots X_n$$

All variables

- Goal: $P(Q|e_1 \ldots e_k)$
- Inference by enumeration:
  - Step 1: Select the entries consistent with the evidence
  - Step 2: Sum out H to get joint of Query and evidence

$$P(Q, e_1 \ldots e_k) = \sum_{h_1 \ldots h_r} P\underbrace{(Q, h_1 \ldots h_r, e_1 \ldots e_k)}_{X_1, X_2, \ldots X_n}$$

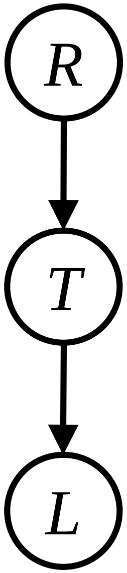
  - Step 3: Normalization

$$P(Q|e_1 \cdots e_k) = \frac{1}{Z} P(Q, e_1 \cdots e_k) \qquad Z = \sum_q P(Q, e_1 \cdots e_k)$$

Enumeration: Example

- Random Variables
  - R: Raining
  - T: Traffic
  - L: Late for class
- Goal:

$$P(L) = ?$$



$P(R)$

+r	0.1
-r	0.9

$P(T|R)$

+r	+t	0.8
+r	-t	0.2
-r	+t	0.1
-r	-t	0.9

$P(L|T)$

+t	+l	0.3
+t	-l	0.7
-t	+l	0.1
-t	-l	0.9

Enumeration: Example

- Inference: Step 1: initialize factors & select entries
  - Initial factors are local CPTs (one per node)

$\Phi(R)$

$P(R)$

+r	0.1
-r	0.9

$\Phi(T,R)$

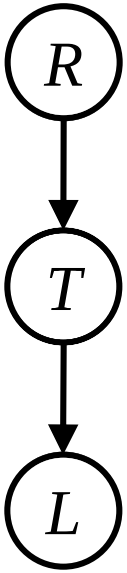
$P(T|R)$

+r	+t	0.8
+r	-t	0.2
-r	+t	0.1
-r	-t	0.9

$\Phi(L,T)$

$P(L|T)$

+t	+l	0.3
+t	-l	0.7
-t	+l	0.1
-t	-l	0.9



  - Any known values are selected
    - E.g. if we know $R = +r$, the initial factors are

$P(+r)$

+r	0.1

$P(T| + r)$

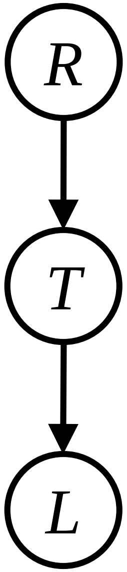
+r	+t	0.8
+r	-t	0.2

$P(L|T)$

+t	+l	0.3
+t	-l	0.7
-t	+l	0.1
-t	-l	0.9

Enumeration: Example

- Inference: Step 2: join factors



$\Phi_1(R)$
$P(R)$

+r	0.1
-r	0.9

$\Phi_2(T,R)$
$P(T|R)$

+r	+t	0.8
+r	-t	0.2
-r	+t	0.1
-r	-t	0.9

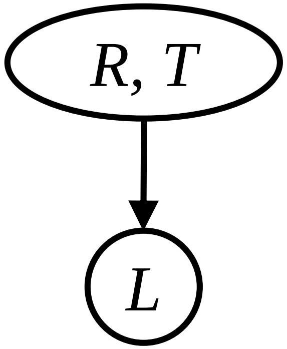
$\Phi_3(L,T)$
$P(L|T)$

+t	+l	0.3
+t	-l	0.7
-t	+l	0.1
-t	-l	0.9

Join R

$\Phi_4(T,R)$
$= \Phi_1(R)\Phi_2(T,R)$
$P(R,T)$

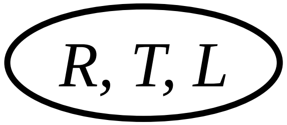
+r	+t	0.08
+r	-t	0.02
-r	+t	0.09
-r	-t	0.81



Join T

$\Phi_5(T,R,L)$
$= \Phi_3(L,T)\Phi_4(T,R)$
$P(R,T,L)$

+r	+t	+l	0.024
+r	+t	-l	0.056
+r	-t	+l	0.002
+r	-t	-l	0.018
-r	+t	+l	0.027
-r	+t	-l	0.063
-r	-t	+l	0.081
-r	-t	-l	0.729



Enumeration: Example

- Inference: Step 3: marginalization

$\Phi_5(T,R,L)$

$P(R,T,L)$

+r	+t	+l	0.024
+r	+t	-l	0.056
+r	-t	+l	0.002
+r	-t	-l	0.018
-r	+t	+l	0.027
-r	+t	-l	0.063
-r	-t	+l	0.081
-r	-t	-l	0.729

Sum out R

$f(T,L)=\sum_R \Phi_5(T,R,L)$

$$= \sum_R \Phi_1(R)\Phi_2(T,R)\Phi_3(L,T)$$

$P(T,L)$

+t	+l	0.051
+t	-l	0.119
-t	+l	0.083
-t	-l	0.747

Sum out T

$$\sum_T f(T,L)$$

$P(L)$

+l	0.134
-l	0.886

R, T, L

T, L

L

Enumeration

- Join up the whole joint distribution before you sum out the hidden variables
- Computational complexity: O(d^n)
- Very slow!

Variable Elimination

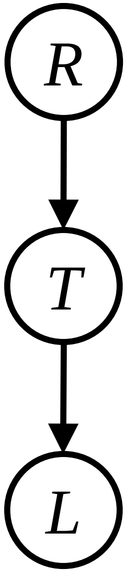
- Variable Elimination
  - Idea: interleave joining and marginalizing!
  - Still NP-hard, but usually much faster than inference by enumeration

- Inference by Enumeration

$$P(L) = \underbrace{\sum_t \underbrace{\sum_r \underbrace{P(L|t)\underbrace{P(r)P(t|r)}_{\text{Join on r}}}_{\text{Join on t}}}_{\text{Eliminate r}}}_{\text{Eliminate t}}$$

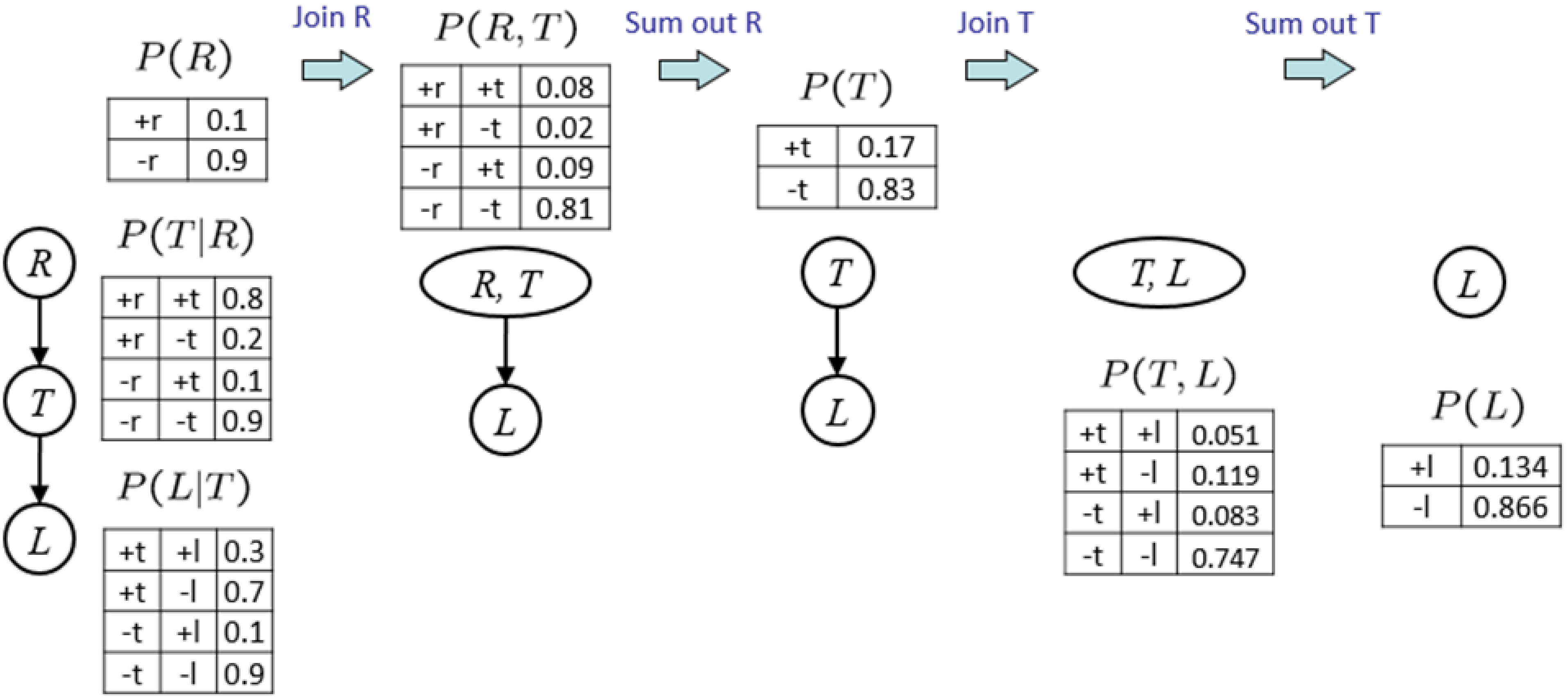
- Variable Elimination

$$= \underbrace{\sum_t \underbrace{P(L|t)\underbrace{\sum_r \underbrace{P(r)P(t|r)}_{\text{Join on r}}}_{\text{Eliminate r}}}_{\text{Join on t}}}_{\text{Eliminate t}}$$



Variable Elimination

- $P(L) = ?$



Join R

Sum out R

Join T

Sum out T

$P(R)$

+r	0.1
-r	0.9

$P(T|R)$

+r	+t	0.8
+r	-t	0.2
-r	+t	0.1
-r	-t	0.9

$P(L|T)$

+t	+l	0.3
+t	-l	0.7
-t	+l	0.1
-t	-l	0.9

$P(R, T)$

+r	+t	0.08
+r	-t	0.02
-r	+t	0.09
-r	-t	0.81

$P(T)$

+t	0.17
-t	0.83

$P(T, L)$

+t	+l	0.051
+t	-l	0.119
-t	+l	0.083
-t	-l	0.747

$P(L)$

+l	0.134
-l	0.866

Variable Elimination with Evidence

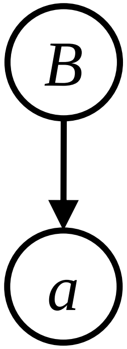
- $P(B|+a) = ?$

Initial / Select

$\Phi_1(B)$

$P(B)$

B	P
+b	0.1
¬b	0.9

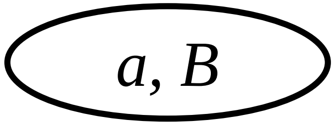


$\Phi_2(B)$

$P(A|B) \rightarrow P(a|B)$

B	A	P
+b	+a	0.8
~~+b~~	~~¬a~~	~~0.2~~
¬b	+a	0.1
~~¬b~~	~~¬a~~	~~0.9~~

Join on B



$\Phi_3(B) = \Phi_1(B)\Phi_2(B)$

$P(a, B)$

A	B	P
+a	+b	0.08
+a	¬b	0.09

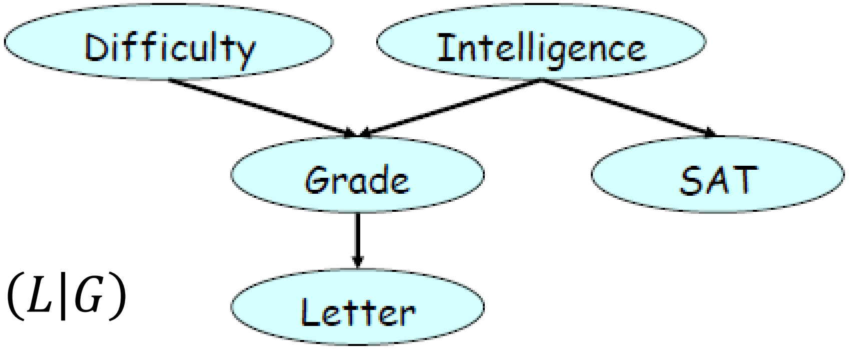
Normalize

$P(B|a)$

A	B	P
+a	+b	8/17
+a	¬b	9/17

Variable Elimination

- Goal: $P(L)$= ?
- Eliminate: D, I, G, S



- $P(D,I,G,S,L) = P(D)P(I)P(G|D,I)P(S|I)P(L|G)$

- $P(L) = \sum_{D,I,G,S} P(D,I,G,S,L)$

$$= \sum_{D,I,G,S} \Phi_D(D)\Phi_I(I)\Phi_G(G,D,I)\Phi_S(S,I)\Phi_L(L,G)$$

$$= \sum_{I,G,S} \Phi_I(I)\,\Phi_S(S,I)\Phi_L(L,G) \sum_D \Phi_D(D)\Phi_G(G,D,I)$$

$$= \sum_{I,G,S} \Phi_I(I)\Phi_S(S,I)\Phi_L(L,G)\, f_1(G,I)$$

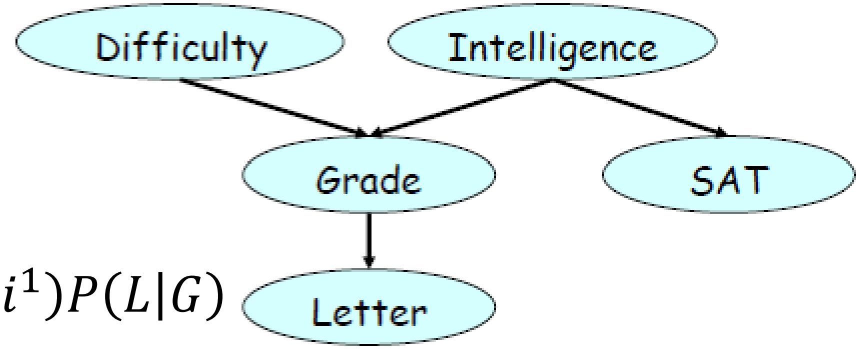
$$= \sum_{G,S} \Phi_L(L,G) \sum_I \Phi_I(I)\Phi_S(S,I) f_1(G,I)$$

$$= \sum_{G,S} \Phi_L(L,G)\, f_2(G,S)$$

$$= \sum_G \sum_S \Phi_L(L,G)\, f_2(G,S)$$

Variable Elimination with Evidence

- Goal: $P(L, i^1)$= ?
- Eliminate: D, G, S

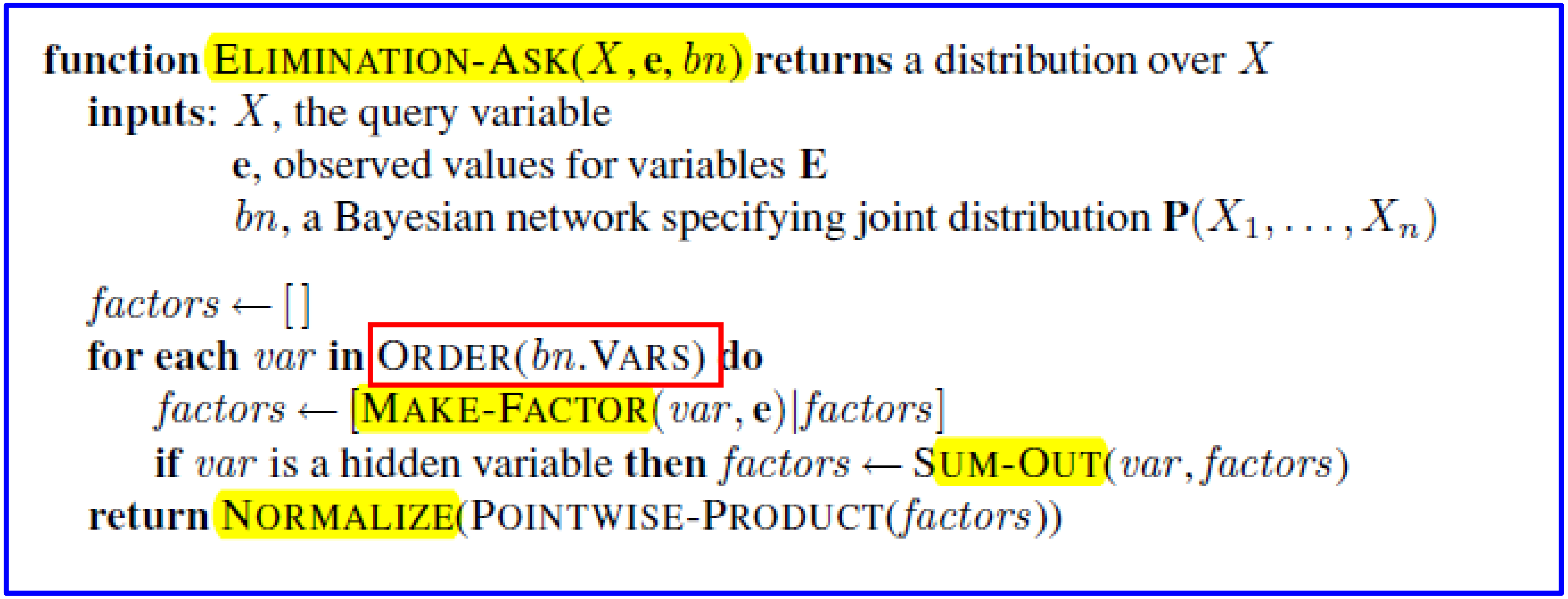


- $P(D, i^1, G, S, L) = P(D)P(i^1)P(G|D, i^1)P(S|i^1)P(L|G)$

- $P(L, i^1) = \sum_{D,G,S} P(D, i^1, G, S, L)$

$$= \sum_{D,G,S} \Phi_D(D)\, \Phi_G(G, D) \Phi_S(S) \Phi_L(L, G)$$

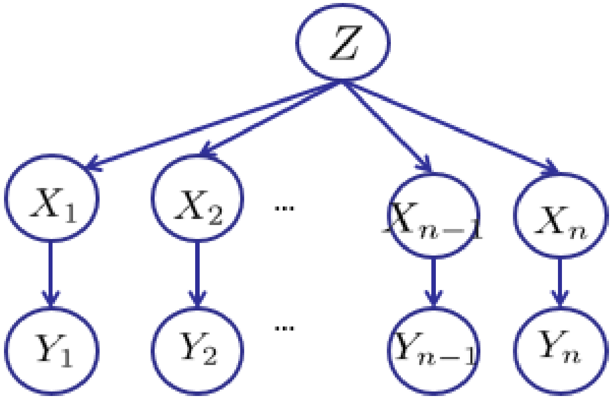
Variable Elimination Algorithm



```
function ELIMINATION-ASK($X$, $\mathbf{e}$, $bn$) returns a distribution over $X$
    inputs: $X$, the query variable
            $\mathbf{e}$, observed values for variables $\mathbf{E}$
            $bn$, a Bayesian network specifying joint distribution $\mathbf{P}(X_1, \ldots, X_n)$

    $factors \leftarrow [\,]$
    for each $var$ in ORDER($bn$.VARS) do
        $factors \leftarrow$ [MAKE-FACTOR($var$, $\mathbf{e}$)|$factors$]
        if $var$ is a hidden variable then $factors \leftarrow$ SUM-OUT($var$, $factors$)
    return NORMALIZE(POINTWISE-PRODUCT($factors$))
```

Variable Elimination Ordering

- Query $P(X_n | y_1, \dots, y_n)$
- Ordering:
  - $Z, X_1, \dots, X_{n-1}$ and $X_1, \dots, X_{n-1}, Z$
  - What is the size of the maximum factor generated for each of the orderings?



  - Answer: 2^{n+1} versus 2^2 (assuming binary)
  - In general: the ordering can greatly affect efficiency.

Variable Elimination Ordering

- Computational and Space Complexity:
  - The computational and space complexity of variable elimination is determined by the largest factor
  - The elimination ordering can greatly affect the size of the largest factor.
  - Does there always exist an ordering that only results in small factors?
    - Min-neighbors
    - Min-weights

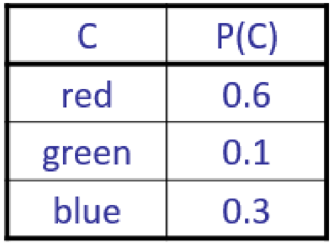
Sampling

- Basic idea
  - Draw N samples from a sampling distribution S
  - Compute an approximate posterior probability
  - Show this converges to the true probability P

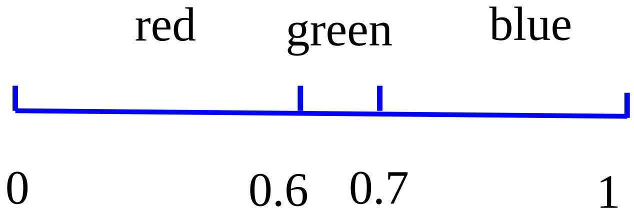
- Why sampling?
  - Approximate inference: getting a sample is faster than computing the right answer (e.g. with variable elimination)

Sampling from Given Distribution

- Sampling
  - Step 1: Get sample u from uniform distribution over [0, 1)
  - Step 2: Convert this sample u into an outcome for the given distribution
- Example:
  - Discrete:



C	P(C)
red	0.6
green	0.1
blue	0.3

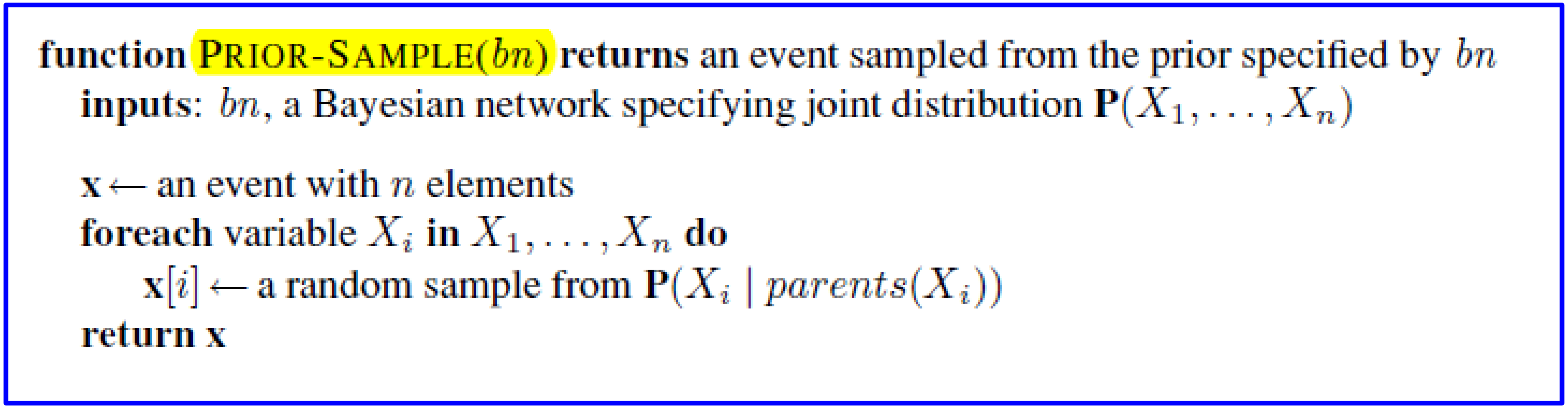


Sampling in Bayesian Network

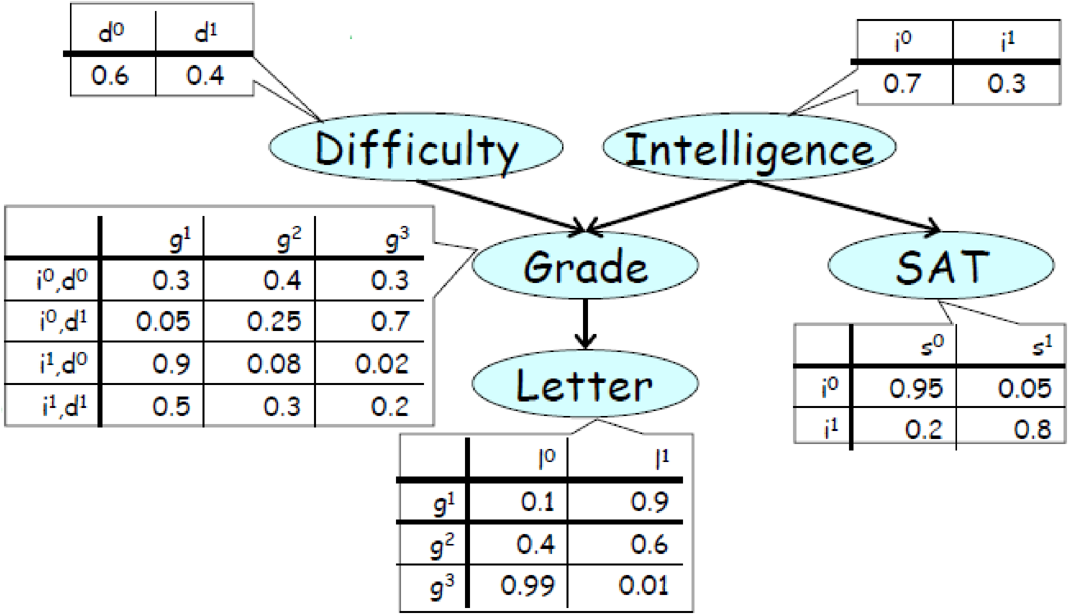
- Direct sampling methods
  - Prior / Froward sampling
  - Rejection sampling
  - Likelihood weighting
- Markov chain sampling methods
  - Gibbs sampling
  - Collapsed Gibbs sampling
  - Markov chain Monte Carlo

Prior Sampling

- Goal: Estimate *$P(X=x)$*



```
function PRIOR-SAMPLE($bn$) returns an event sampled from the prior specified by $bn$
    inputs: $bn$, a Bayesian network specifying joint distribution $\mathbf{P}(X_1, \ldots, X_n)$

    $\mathbf{x} \leftarrow$ an event with $n$ elements
    foreach variable $X_i$ in $X_1, \ldots, X_n$ do
        $\mathbf{x}[i] \leftarrow$ a random sample from $\mathbf{P}(X_i \mid parents(X_i))$
    return $\mathbf{x}$
```



d^0	d^1
0.6	0.4

i^0	i^1
0.7	0.3

	g^1	g^2	g^3
i^0,d^0	0.3	0.4	0.3
i^0,d^1	0.05	0.25	0.7
i^1,d^0	0.9	0.08	0.02
i^1,d^1	0.5	0.3	0.2

	s^0	s^1
i^0	0.95	0.05
i^1	0.2	0.8

	l^0	l^1
g^1	0.1	0.9
g^2	0.4	0.6
g^3	0.99	0.01

Prior Sampling

- This process generates samples with probability:

$$S_{PS}(x_1, \dots, x_n) = \prod_{i=1}^{n} P(x_i | Par(X_i)) = P(x_1, \dots, x_n)$$

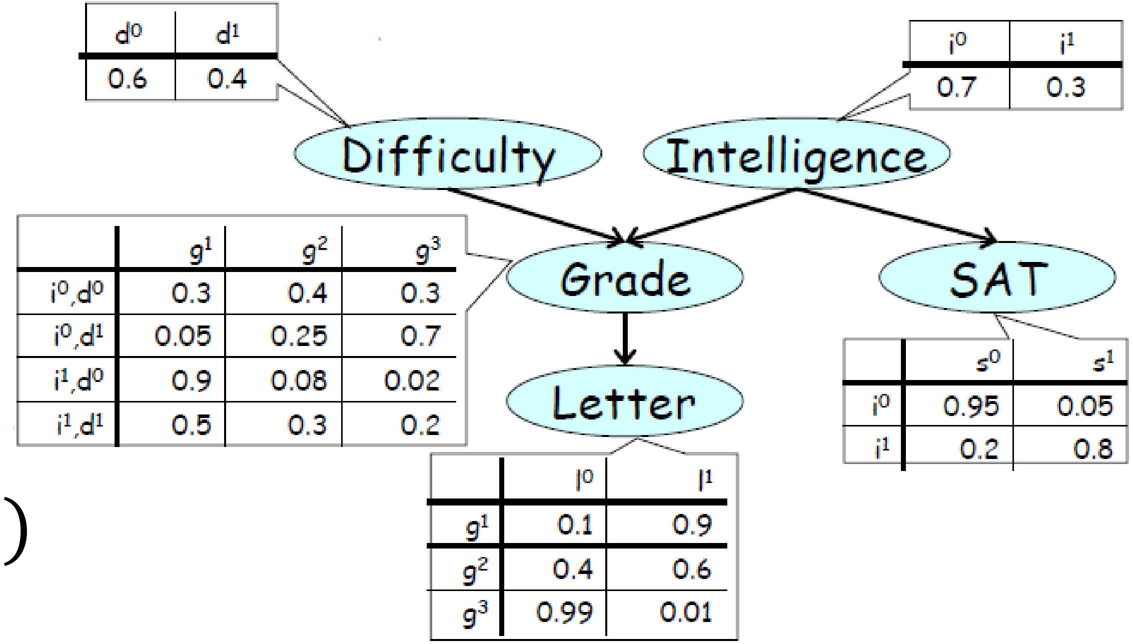
- Let the number of samples of an event be $N_{PS}(x_1, \dots, x_n)$
- Then

$$\lim_{N \to \infty} \hat{P}\,(x_1, \dots, x_n) = \lim_{N \to \infty} N_{PS}(x_1, \dots, x_n)\,/N$$
$$= S_{PS}\,(x_1, \dots, x_n)$$
$$= P(x_1, \dots, x_n)$$

- The sampling procedure is consistent!

Example

- We’ll get a bunch of samples from the BN:
  - d^0, i^1, g^1, s^1, l^1
  - d^1, i^0, g^3, s^0, l^0
  - d^0, i^1, g^2, s^0, l^1
  - d^0, i^0, g^2, s^0, l^1
  - d^1, i^1, g^1, s^1, l^0
  - d^0, i^1, g^3, s^1, l^0



d^0	d^1
0.6	0.4

i^0	i^1
0.7	0.3

	g^1	g^2	g^3
i^0,d^0	0.3	0.4	0.3
i^0,d^1	0.05	0.25	0.7
i^1,d^0	0.9	0.08	0.02
i^1,d^1	0.5	0.3	0.2

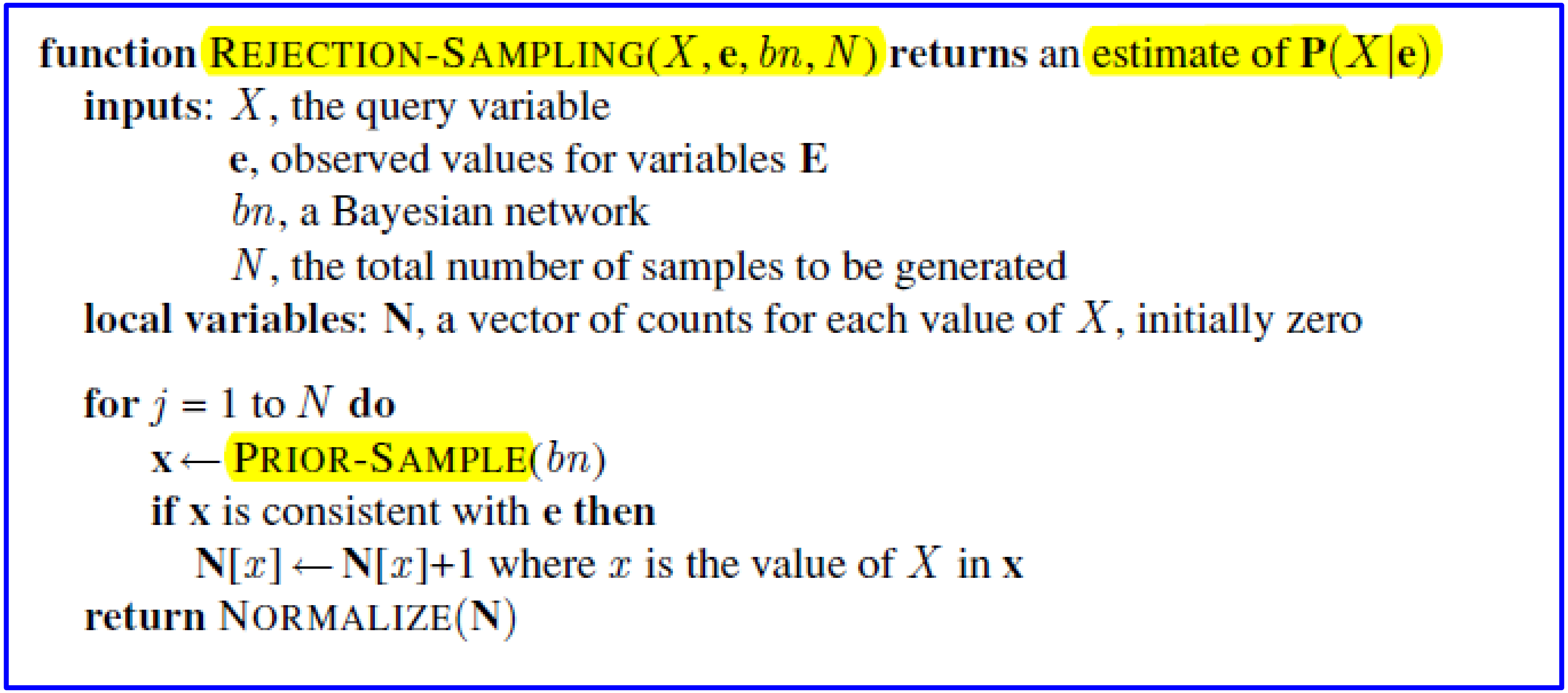
	s^0	s^1
i^0	0.95	0.05
i^1	0.2	0.8

	l^0	l^1
g^1	0.1	0.9
g^2	0.4	0.6
g^3	0.99	0.01

- If we want to know $P(L)$
  - l^0#:3, l^1#: 3
  - $\Rightarrow P(l^0)$=0.5, $P(l^1)$=0.5

Rejection Sampling

- Goal: Estimate *P(X=x|E=e)*



```
function REJECTION-SAMPLING($X$, e, $bn$, $N$) returns an estimate of $\mathbf{P}(X|\mathbf{e})$
    inputs: $X$, the query variable
            e, observed values for variables E
            $bn$, a Bayesian network
            $N$, the total number of samples to be generated
    local variables: N, a vector of counts for each value of $X$, 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)
```

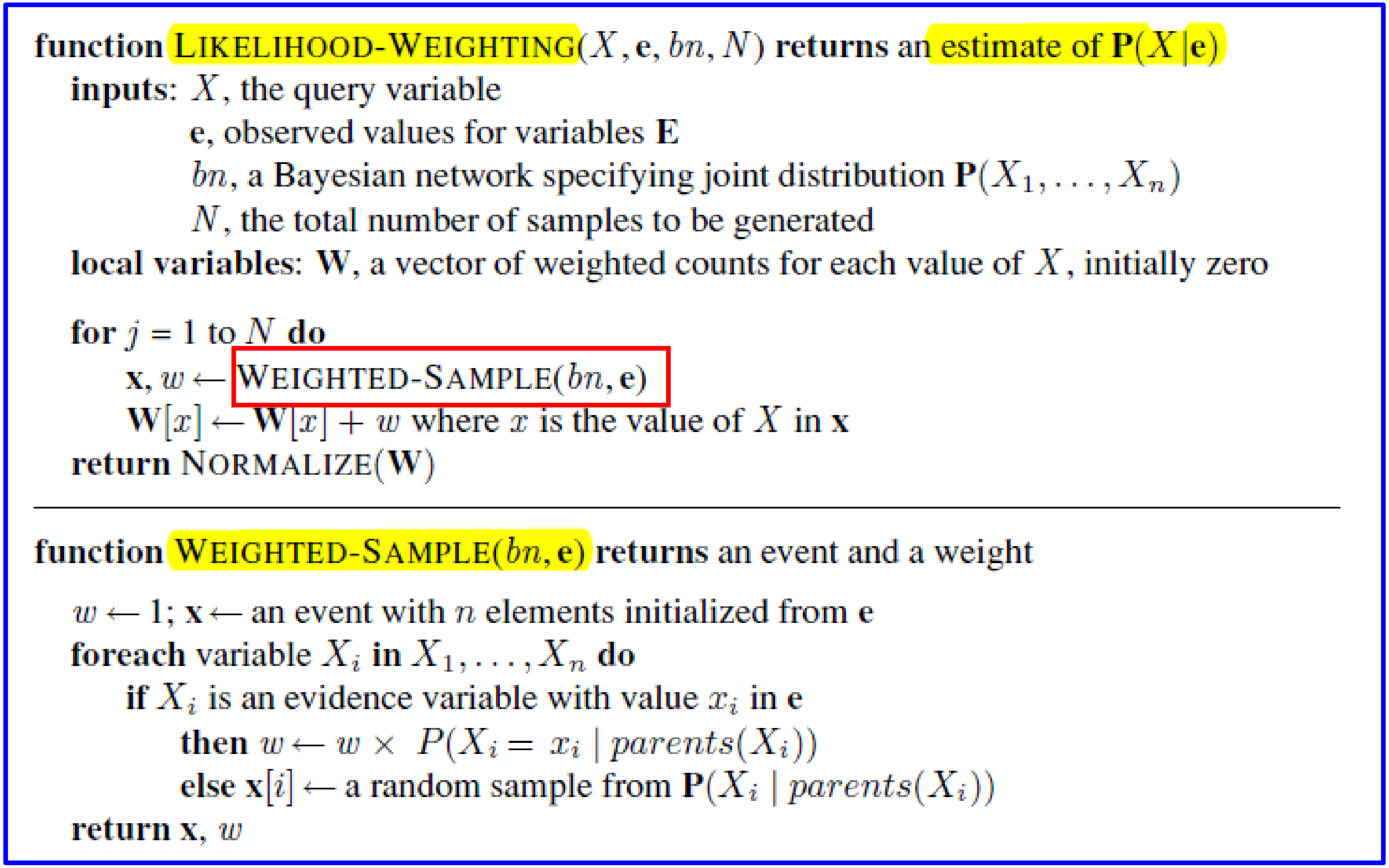
- Problem: reject lots of samples if the evidence is unlikely

Likelihood Weighting

- Goal: Estimate $P(X=x|E=e)$
- Basic idea:
  - Fix evidence variables and sample the rest
  - Problem: sample distribution not consistent!
  - Solution: weight by probability of evidence given parents

Likelihood Weighting

- Goal: Estimate P(X=x|E=e)

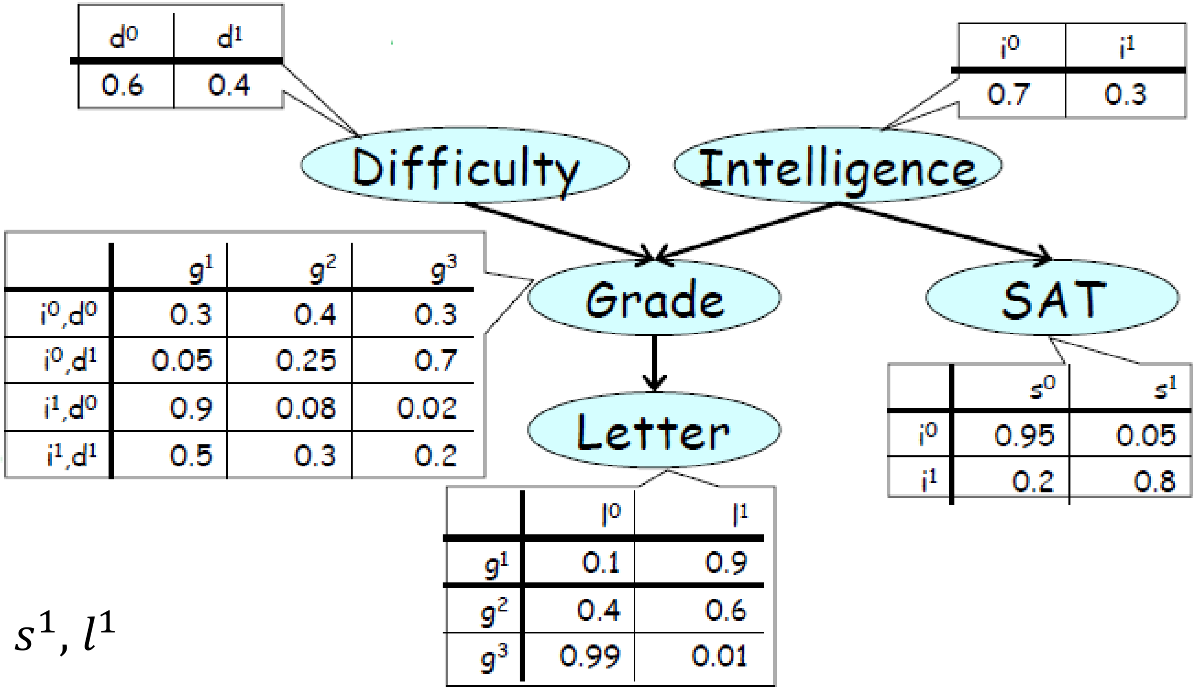


```
function LIKELIHOOD-WEIGHTING($X$, e, $bn$, $N$) returns an estimate of $\mathbf{P}(X|\mathbf{e})$
    inputs: $X$, the query variable
            e, observed values for variables E
            $bn$, a Bayesian network specifying joint distribution $\mathbf{P}(X_1, \ldots, X_n)$
            $N$, the total number of samples to be generated
    local variables: W, a vector of weighted counts for each value of $X$, initially zero

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

```
function WEIGHTED-SAMPLE($bn$, e) returns an event and a weight

    $w \leftarrow 1$; x ← an event with $n$ elements initialized from e
    foreach variable $X_i$ in $X_1, \ldots, X_n$ do
        if $X_i$ is an evidence variable with value $x_i$ in e
            then $w \leftarrow w \times P(X_i = x_i \mid parents(X_i))$
            else $\mathbf{x}[i] \leftarrow$ a random sample from $\mathbf{P}(X_i \mid parents(X_i))$
    return x, $w$
```

Likelihood Weighting

- Example: $P(L|g^1)$?



d^0	d^1
0.6	0.4

i^0	i^1
0.7	0.3

	g^1	g^2	g^3
i^0,d^0	0.3	0.4	0.3
i^0,d^1	0.05	0.25	0.7
i^1,d^0	0.9	0.08	0.02
i^1,d^1	0.5	0.3	0.2

	s^0	s^1
i^0	0.95	0.05
i^1	0.2	0.8

	l^0	l^1
g^1	0.1	0.9
g^2	0.4	0.6
g^3	0.99	0.01

d^0, i^0, g^1, s^1, l^1

$w = 1 \times 0.3$

Likelihood Weighting

- Sampling distribution if z sampled and e fixed evidence

$$S_{WS}(\mathbf{z}, \mathrm{e}) = \prod_{i=1}^{l} P(z_i|\mathsf{Parents}(Z_i))$$

- Now, samples have weights

$$w(\mathbf{z}, \mathrm{e}) = \prod_{i=1}^{m} P(e_i|\mathsf{Parents}(E_i))$$

- Together, weighted sampling distribution is consistent

$$S_{\mathrm{WS}}(z, e) \cdot w(z, e) = \prod_{i=1}^{l} P(z_i|\mathrm{Parents}(z_i)) \prod_{i=1}^{m} P(e_i|\mathrm{Parents}(e_i))$$

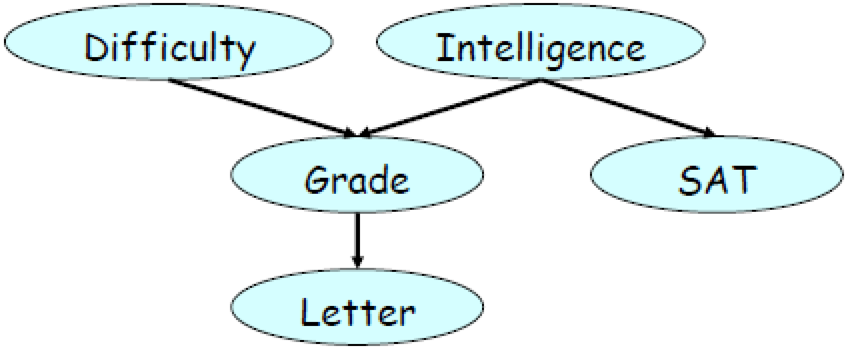
$$= P(\mathbf{z}, \mathrm{e})$$

Likelihood Weighting

- Pros:
  - We have taken evidence into account as we generate the sample
  - More of our samples will reflect the state of the world suggested by the evidence

- Cons:
  - Evidence influences the choice of downstream variables, but not upstream ones
  - Suffer a degradation in performance as the number of evidence variables increases.

$P(L|g^1)$



Markov Chain Sampling

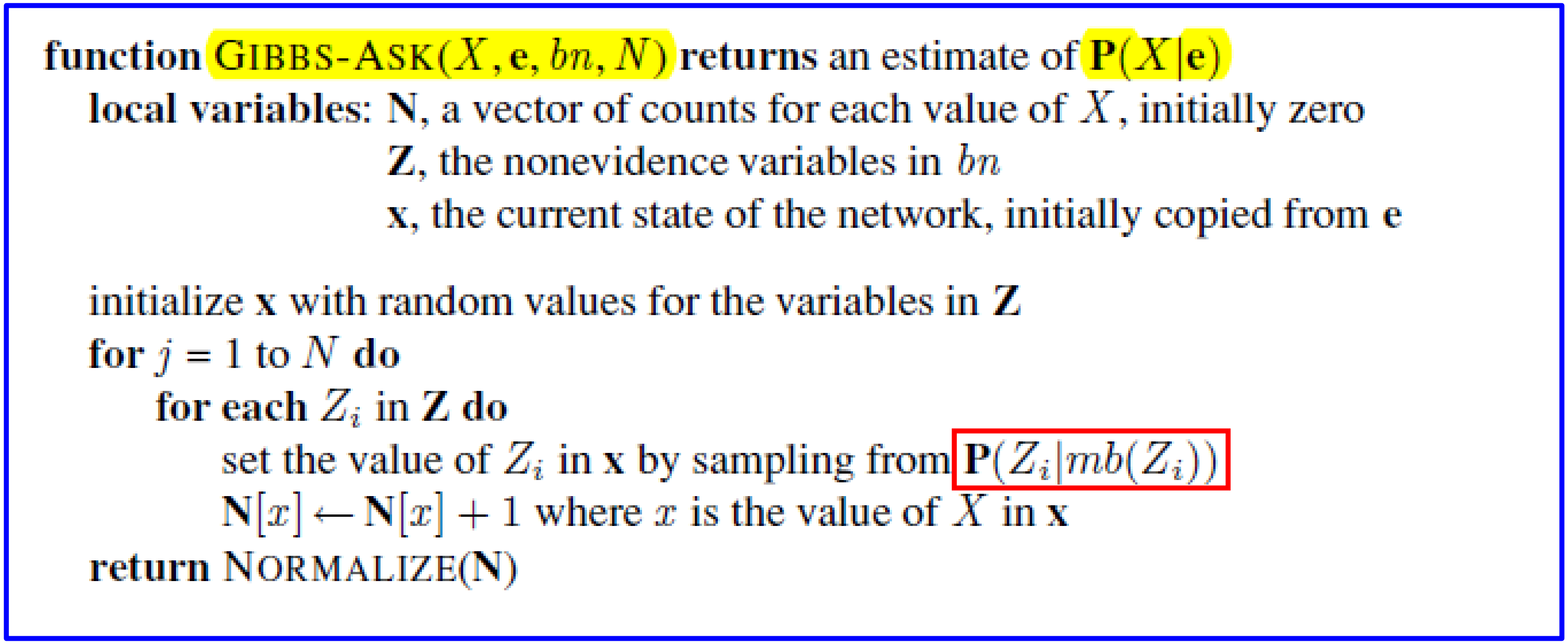
- Basic idea:
    - Generate each sample by making a random change to the preceding sample.
- Methods:
    - Gibbs sampling
    - Collapsed Gibbs sampling
    - Markov chain Monte Carlo

Gibbs Sampling

- Goal: Estimate $P(X=x|E=e)$
- Basic idea:
  - Step 1: Fix evidence
  - Step 2: Randomly initialize other variables
  - Step 3: Repeat
    - Choose a non-evidence variable X
    - Resample X from $P(X|all\ other\ variables)$

Gibbs Sampling

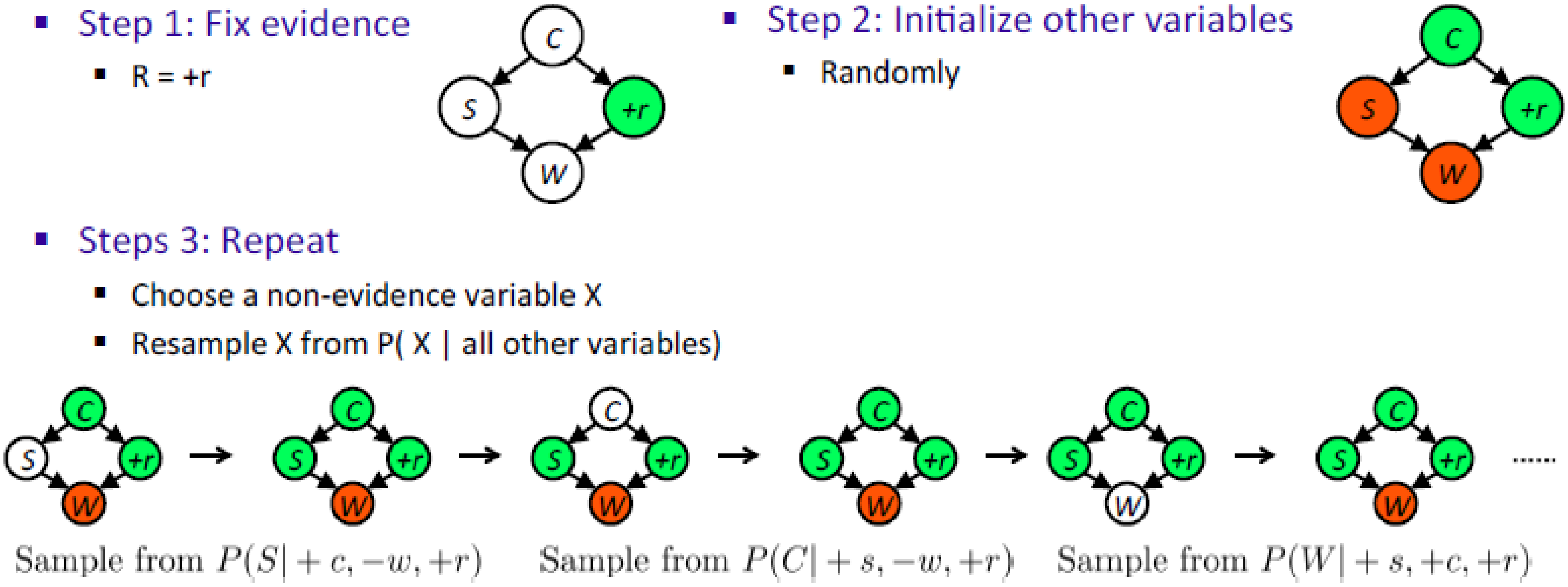
- Goal: Estimate *P(X=x|E=e)*



```
function GIBBS-ASK($X$, $\mathbf{e}$, $bn$, $N$) returns an estimate of $\mathbf{P}(X|\mathbf{e})$
    local variables: $\mathbf{N}$, a vector of counts for each value of $X$, 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 $\mathbf{Z}$
    for $j$ = 1 to $N$ do
        for each $Z_i$ in $\mathbf{Z}$ do
            set the value of $Z_i$ in $\mathbf{x}$ by sampling from $\boxed{\mathbf{P}(Z_i|mb(Z_i))}$
            $\mathbf{N}[x] \leftarrow \mathbf{N}[x] + 1$ where $x$ is the value of $X$ in $\mathbf{x}$
    return NORMALIZE($\mathbf{N}$)
```

Gibbs Sampling

- Example



- Step 1: Fix evidence
  - R = +r
- Step 2: Initialize other variables
  - Randomly
- Steps 3: Repeat
  - Choose a non-evidence variable X
  - Resample X from P(X | all other variables)

Sample from $P(S|+c,-w,+r)$ Sample from $P(C|+s,-w,+r)$ Sample from $P(W|+s,+c,+r)$

Gibbs Sampling

- Pros:
  - The simplest Markov chain for PGMs
  - Computationally efficient to sample

- Others:
  - Collapsed Gibbs Sampling
  - Markov Chain Monte Carlo

Assignments

- Reading assignment:
    - Ch. 14.1-14.5

- Homework 4