Access Code: **784952**

CMP9794M
Advanced Artificial Intelligence

[Heriberto Cuayahuitl]

UNIVERSITY OF
LINCOLN

School of Engineering and Physical Sciences

Thinking humanly	Thinking rationally
Acting humanly	Acting rationally

Last Week

- Main approaches to AI
- Agents & environments

Fully-observable vs. partially observable
Single-agent vs. multi-agent
Deterministic vs. stochastic
Episodic vs. sequential
Static vs. dynamic
Discrete vs. continuous
Known vs. unknown

- History and developments
- Probability theory

P(A | B) = P(A ∧ B) / P(B)

P(A ∧ B) = P(A | B) * P(B)

P(A | B) + P(¬A | B) = 1

P(B) = $\sum_a$ P(A=a, B) = $\sum_a$ P(A=a | B) * P(B)

P(A)+P(¬A)=1, therefore P(¬A)=1-P(A)

P(B)+P(¬B)=1, therefore P(B)=1-P(¬B)

P(A ∧ B) = P(B ∧ A)

P(A | B) ≠ P(B | A)

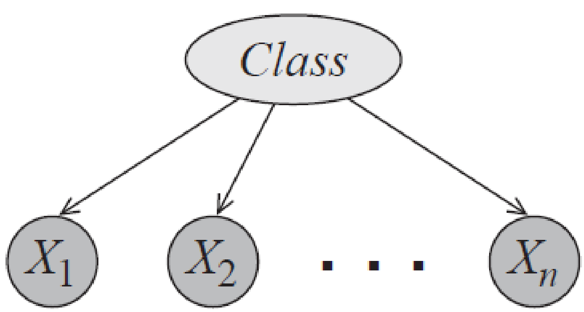
P(A | B) = (P(B | A) * P(A)) / P(B)

- Naïve Bayes classifier

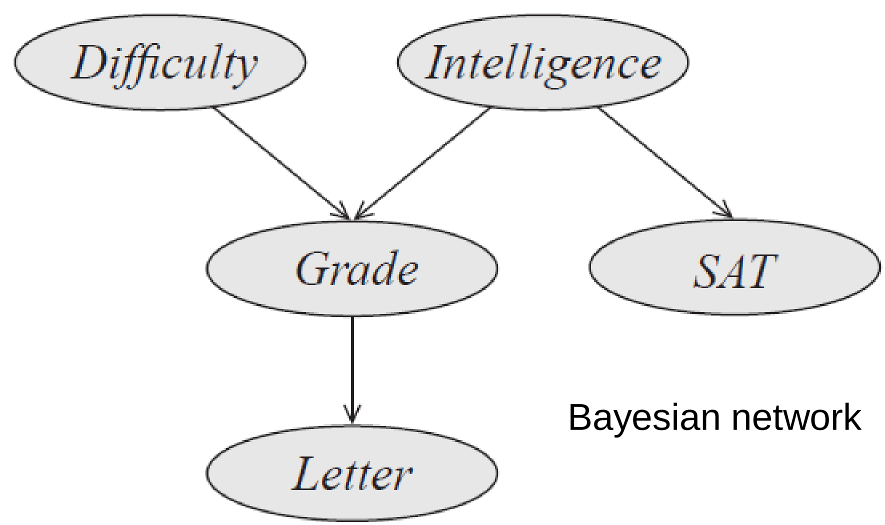
$$Y = \underset{y_k}{\operatorname{argmax}} P(Y = y_k) \prod_i P(X_i | Y = y_k)$$

From Naïve Bayes to Bayesian Nets

Naïve bayes is a simple Bayesian Network (BN) with a strong independence assumption, which is relaxed in BNs via not so simple structures.



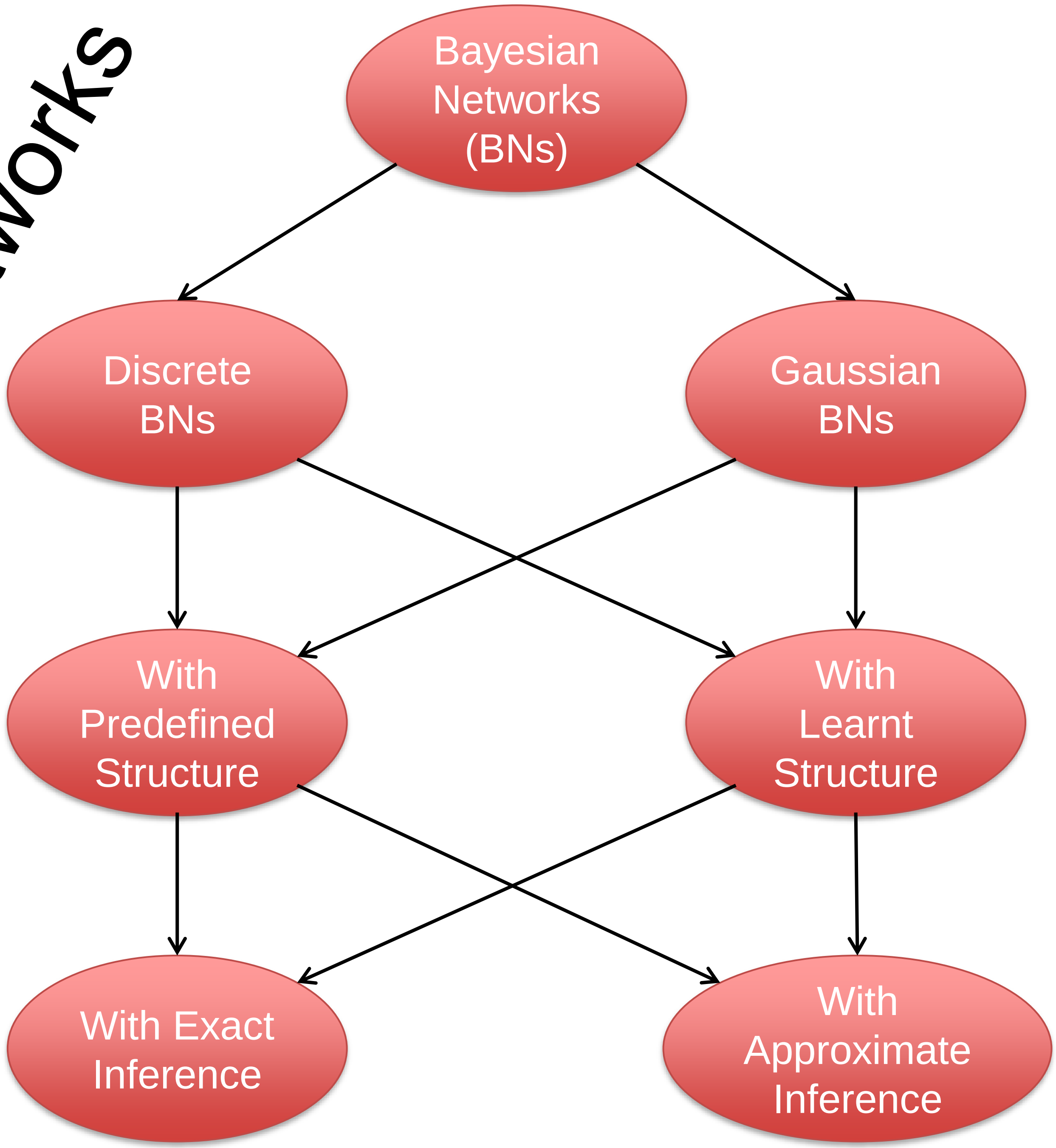
Naïve Bayes graphical model



Bayesian network

Images from Koller & Friedman. Probabilistic Graphical Models, 2009. Chapter 3.

Taxonomy of Bayesian Networks



Today

- **Introduction to Discrete Bayesian networks**
  - Graphical representation
  - Probabilistic representation
  - Parameter learning

- Algorithms for exact inference
  - Inference by enumeration
  - Inference by variable elimination

Bayesian Networks

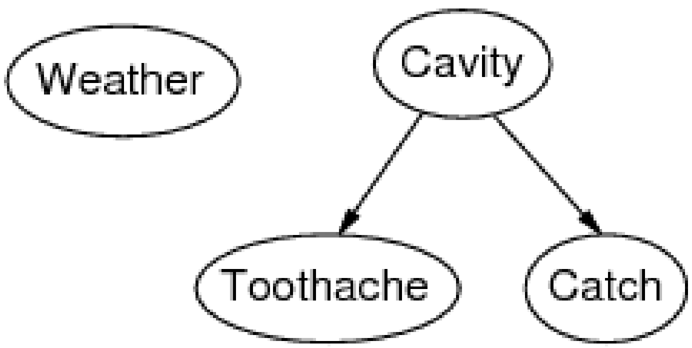
- Bayesian Networks (Bayes Nets or Belief Nets) can represent any full joint probability distribution—and they can do so very concisely!

- Syntax:
  - a set of nodes, one per random variable
  - a directed acyclic graph (link=“directly influences")
  - a conditional distribution for each node given its parents: $P(X_i|parents(X_i))$

Bayesian Networks (BNs)

- Each node of a BN is represented by a conditional probability table (CPT)—a probability distribution over X_i for each combination of parent values.

- The topology of a network encodes conditional independence assertions:
  - *Weather* is independent of the other variables
  - *Toothache* and *Catch* are conditionally independent given *Cavity*



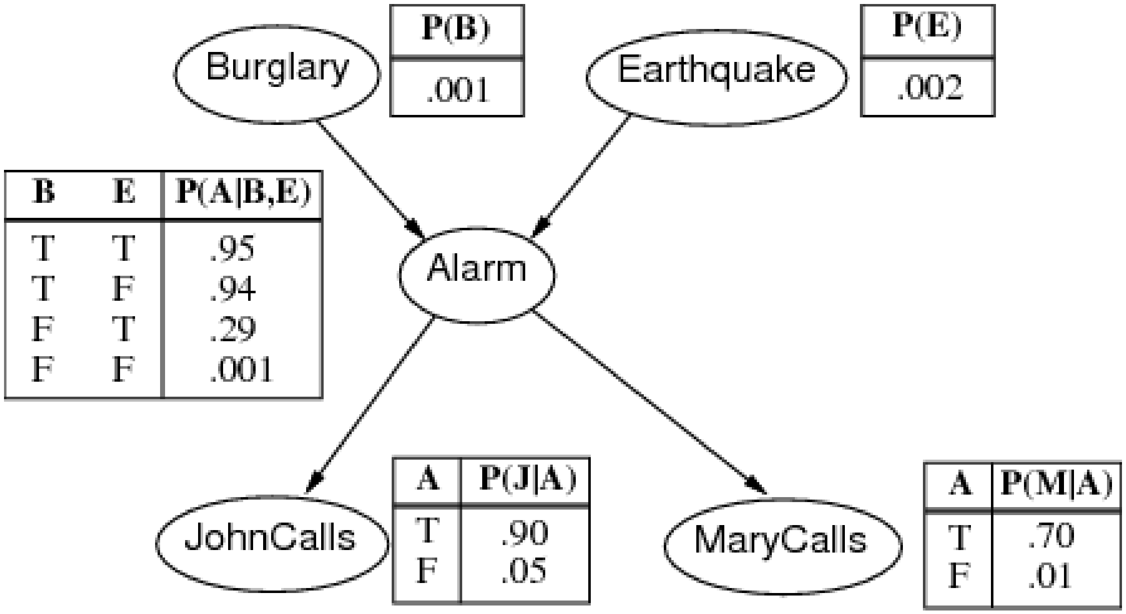
Example Scenario

- Excerpt from Russell and Norvig (2016) *“I am at work, my neighbour John calls to say my alarm is ringing, and my neighbour Mary doesn't call. Sometimes the alarm is set off by minor earthquakes. Is there a burglar?”*

- Random variables (binary):
  - B=Burglar
  - E=Earthquake
  - A=Alarm
  - J=JohnCalls
  - M=MaryCalls

Example Scenario

- The network topology reflects “causal” knowledge:
  - A burglar can set the alarm on
  - An earthquake can set the alarm on
  - The alarm can cause Mary to call
  - The alarm can cause John to call



P(B)
.001

P(E)
.002

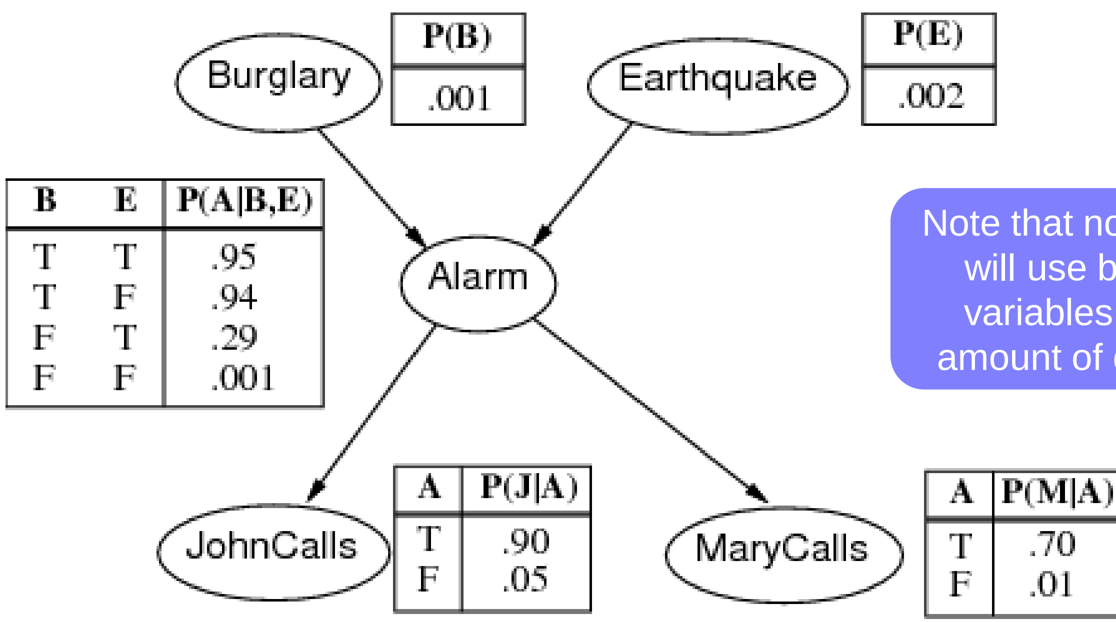
B	E	P(A|B,E)
T	T	.95
T	F	.94
F	T	.29
F	F	.001

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

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

Example Scenario

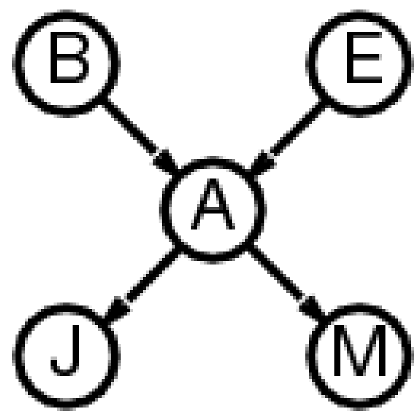
- The network topology reflects “causal” knowledge:
  - A burglar can set the alarm on
  - An earthquake can set the alarm on
  - The alarm can cause Mary to call
  - The alarm can cause John to call



Note that not all Bayes nets will use binary random variables but a varying amount of domain values.

Compactness

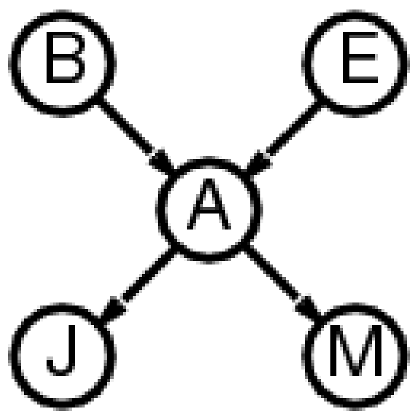
- A CPT for binary variable X_i with k binary parents has 2^k rows for the combinations of parent values



- Each row requires one real number p for $X_i = true$, and one for $X_i = false$ (i.e., $\neg p = 1 - p$). For the burglary net, $2^0 + 2^0 + 2^2 + 2^1 + 2^1 = 10$ numbers.

Compactness

- A CPT for binary variable X_i with k binary parents has 2^k rows for the combinations of parent values



- Each row requires one real number p for $X_i = true$, and one for $X_i = false$ (i.e., $\neg p = 1 - p$). For the burglary net, $2^0 + 2^0 + 2^2 + 2^1 + 2^1 = 10$ numbers. Full enumeration requires $2^1 + 2^1 + 2^3 + 2^2 + 2^2$=20

Compactness

- If each variable has no more than k parents, the complete network requires $n * 2^k$ numbers.

- [**Question**] What is the number of probabilities in a Bayesian Network with 30 random variables, each with 5 parents – using compact enumeration?

- [**Question**] What is the number of probabilities in the full joint distribution – using full enumeration)?

Compactness

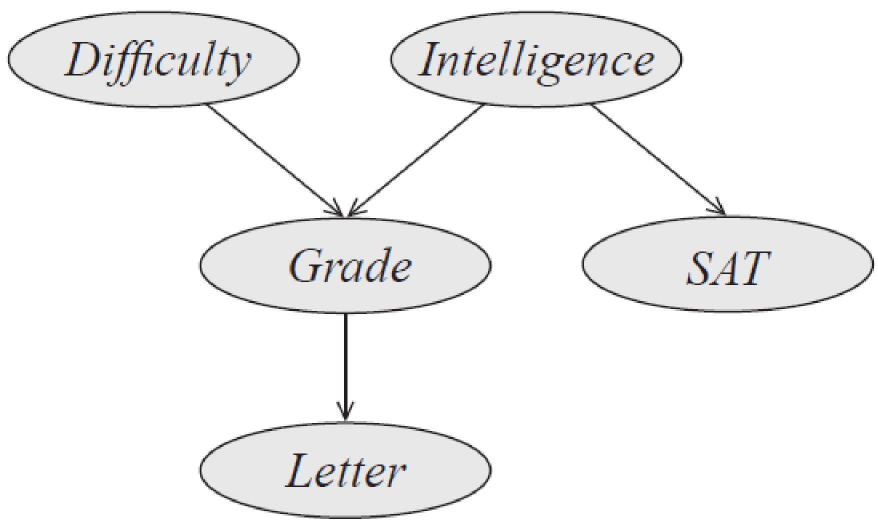
- If each variable has no more than k parents, the complete network requires $n * 2^k$ numbers.

- [**Question**] What is the number of probabilities in a Bayesian Network with 30 random variables, each with 5 parents – using compact enumeration?
$n * 2^k = 30 * 2^5 = 960$

- [**Question**] What is the number of probabilities in the full joint distribution – using full enumeration)?
$2^n = 2^{30} = 1{,}073{,}741{,}824$

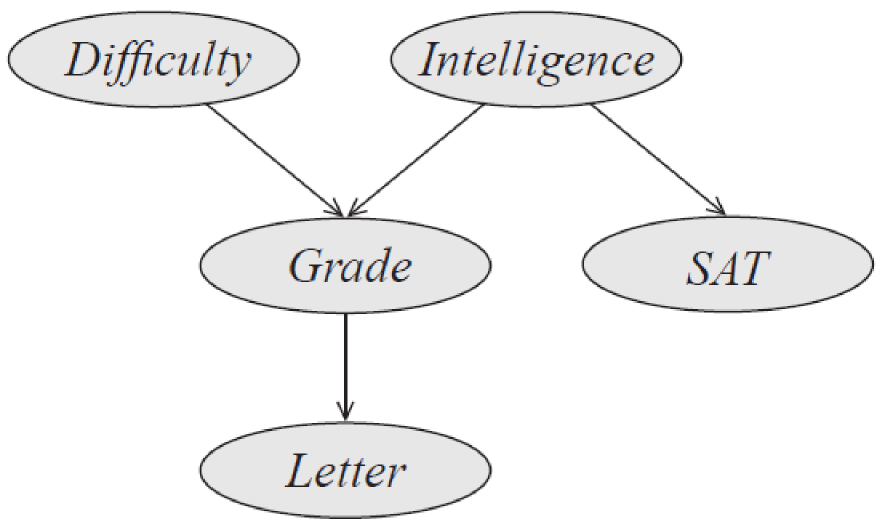
Number of Probabilities in Bayes Nets

[**Question**] How many probabilities are required by all CPTs of the Bayesian Network below considering that all variables except G are binary—G's domain size is 3?



Number of Probabilities in Bayes Nets

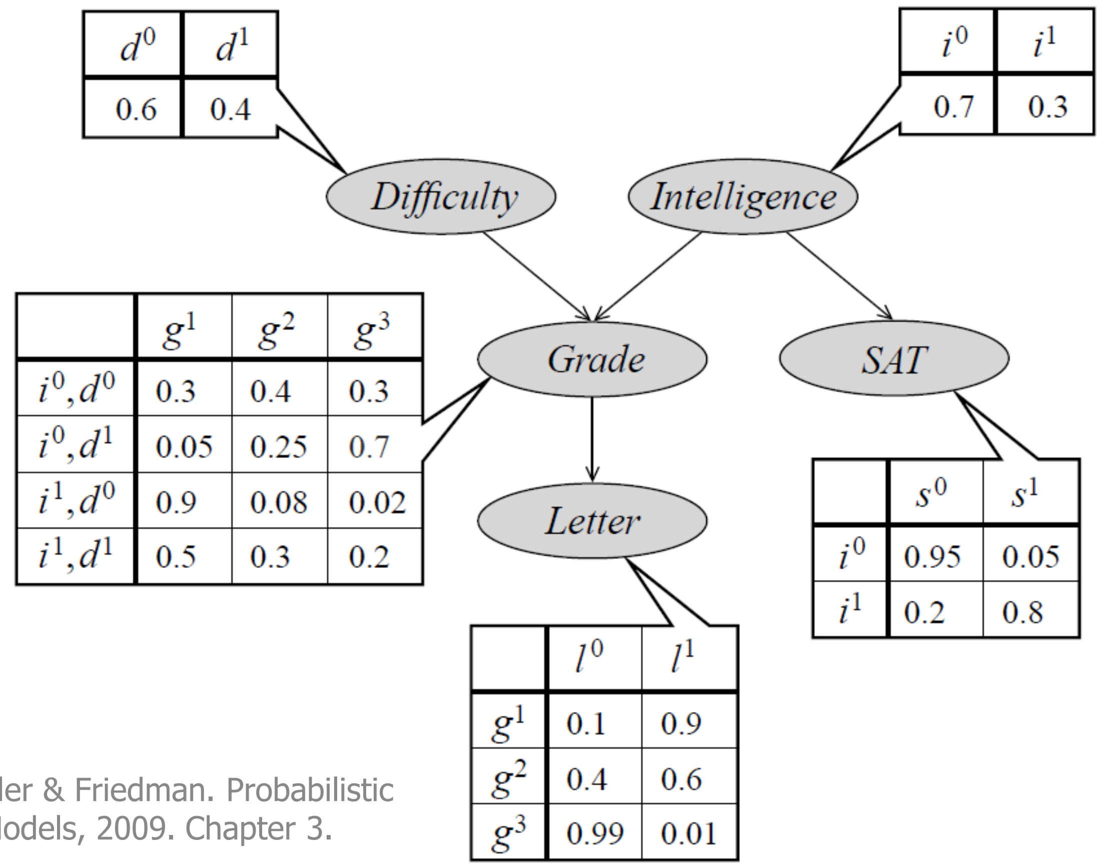
[**Question**] How many probabilities are required by all CPTs of the Bayesian Network below considering that all variables except G are binary—G's domain size is 3?



The answer is 2+2+12+4+6=26 due to $|D| = 2,\ \ |I| = 2,$
$|G| = 3 * 2 * 2 = 12,\ \ |SAT| = 2 * 2 = 4,\ \ |L| = 3 * 2 = 6.$

Number of Probabilities in Bayes Nets

The diagram below should confirm the calculations in the previous slide.



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

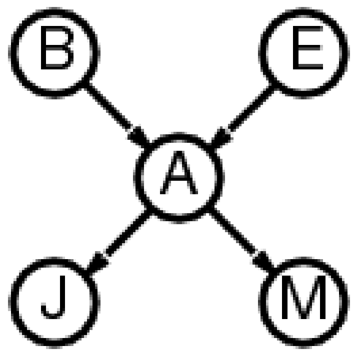
	s^0	s^1
i^0	0.95	0.05
i^1	0.2	0.8

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

Image from Koller & Friedman. Probabilistic Graphical Models, 2009. Chapter 3.

Global Semantics

- “Global” semantics refers to the full joint distribution as the product of local conditional distributions:



- $P(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i | parents(X_i))$

- Example: $P(j \wedge m \wedge a \wedge \neg b \wedge \neg e) = P(j|a)P(m|a)P(a|\neg b, \neg e)P(\neg b)P(\neg e) = 0.9 \times 0.7 \times 0.001 \times 0.999 \times 0.998 \approx 0.00063$

Parameter Learning via MLE (Maximum Likelihood Estimation)

For Conditional Probability Tables (CPTs) with one variable we use $P(X = x) = \frac{count(x)+1}{count(X)+|X|}$, where $|X|$=domain size of variable X

play	P(play)
yes	(9+1)/(14+2)=0.625
no	(5+1)/(14+2)=0.375

For CPTs with two variables we use $P(x|y) = \frac{count(x|y)+1}{count(y)+|X|}$

outlook	play	P(outlook)
sunny	yes	(2+1)/(9+3)=0.25
overcast	yes	(4+1)/(9+3)=0.417
rainy	yes	(3+1)/(9+3)=0.333
sunny	no	(3+1)/(5+3)=0.5
overcast	no	(0+1)/(5+3)=0.125
rainy	no	(2+1)/(5+3)=0.375

For CPTs with 3 vars. we use $P(x|y,z) = \frac{count(x|y,z)+1}{count(y,z)+|X|}$, and so on

Techniques for Parameter Learning avoiding Zero Probabilities

1. Laplace smoothing

$P(x) = \frac{count(x)+1}{N+J}$, where N is the total number of data points and J is the total number of possible outcomes (domain size).

2. Additive smooting

$P(x) = \frac{count(x)+l}{N+l*J}$, where $0 < l < 1$.

3. Dirichlet priors

A Dirichlet prior is a probability distribution over the parameters of a discrete distribution. The prior ensures that all events have non-zero probabilities by distributing probability mass across all possible events.

Techniques for Parameter Learning avoiding Zero Probabilities

1. Laplace smoothing

$P(x) = \frac{count(x)+1}{N+J}$, where N is the total number of data points and J is the total number of possible outcomes (domain size).

2. Additive smooting

$P(x) = \frac{count(x)+l}{N+l*J}$, where $0 < l < 1$.

Look for an implementation of MLE with Laplace/Additive smoothing during this week's workshop: **CPT_Generator.py**

3. Dirichlet priors

A Dirichlet prior is a probability distribution over the parameters of a discrete distribution. The prior ensures that all events have non-zero probabilities by distributing probability mass across all possible events.

Dirichlet Priors via Moment Matching

- Compute empirical probabilities and variances

$\hat{p}_i = \frac{count(x_i)}{N}$, where N=the total number of data points of interest.

$\hat{\sigma}_i^2 = \frac{\hat{p}_i(1-\hat{p}_i)}{N}$, which is the empirical variance of probabity $\hat{p}_i$.

- Match the moments

Mean: $E[P(X = x_i)] = \frac{\alpha_i}{\sum_j \alpha_j}$

Variance: $Var\big(P(X = x_i)\big) = \frac{\alpha_i(\alpha_0 - \alpha_i)}{\alpha_0^2(\alpha_0 + 1)}$, where $\alpha_0 = \sum_j \alpha_j$

Estimation of Dirichlet Parameters: PlayTennis and Outlook Data (1/4)

Compute empirical probabilities:

$$\hat{p}_{yes,sunny} = \frac{count(yes, sunny)}{count(yes)} = \frac{2}{9} = 0.222$$

$$\hat{p}_{yes,overcast} = \frac{count(yes, overcast)}{count(yes)} = \frac{4}{9} = 0.444$$

$$\hat{p}_{yes,rain} = \frac{count(yes, rain)}{count(yes)} = \frac{3}{9} = 0.333$$

$$\hat{p}_{no,sunny} = \frac{count(no, sunny)}{count(no)} = \frac{3}{5} = 0.6$$

$$\hat{p}_{no,overcast} = \frac{count(no, overcast)}{count(no)} = \frac{0}{5} = 0$$

$$\hat{p}_{no,rain} = \frac{count(no, rain)}{count(no)} = \frac{2}{5} = 0.4$$

Estimation of Dirichlet Parameters: PlayTennis and Outlook Data (2/4)

Compute empirical variances:

$$\hat{\sigma}^2_{yes,sunny} = \frac{\hat{p}_{yes,sunny}\left(1 - \hat{p}_{yes,sunny}\right)}{count(yes)} = \frac{0.222 * 0.778}{9} = 0.0192$$

$$\hat{\sigma}^2_{yes,overcast} = \frac{\hat{p}_{yes,overcast}\left(1 - \hat{p}_{yes,overcast}\right)}{count(yes)} = \frac{0.444 * 0.556}{9} = 0.0274$$

$$\hat{\sigma}^2_{yes,rain} = \frac{\hat{p}_{yes,rain}\left(1 - \hat{p}_{yes,rain}\right)}{count(yes)} = \frac{0.333 * 0.667}{9} = 0.0247$$

$$\hat{\sigma}^2_{no,sunny} = \frac{\hat{p}_{no,sunny}\left(1 - \hat{p}_{no,sunny}\right)}{count(no)} = \frac{0.6 * 0.4}{5} = 0.048$$

$$\hat{\sigma}^2_{no,overcast} = \frac{\hat{p}_{no,overcast}\left(1 - \hat{p}_{no,overcast}\right)}{count(no)} = \frac{0 * 1}{5} = 0$$

$$\hat{\sigma}^2_{no,rain} = \frac{\hat{p}_{no,rain}\left(1 - \hat{p}_{no,rain}\right)}{count(no)} = \frac{0.4 * 0.6}{5} = 0.048$$

Estimation of Dirichlet Parameters: PlayTennis and Outlook Data (3/4)

- From moment matching we know that

$\hat{p}_i = \frac{\alpha_i}{\alpha_0}$ and that $\hat{\sigma}_i^2 = \frac{\hat{p}_i(1-\hat{p}_i)}{\alpha_0+1}$

- Estimating α_0 for $PlayTennis = yes$:

$$\alpha_0 = \frac{\hat{p}_{yes,sunny}\left(1 - \hat{p}_{yes,sunny}\right)}{\hat{\sigma}^2_{yes,sunny}} - 1 = \frac{0.222 * 0.778}{0.0192} - 1 = \ 8$$

- Estimating α_0 for $PlayTennis = no$:

$$\alpha_0 = \frac{\hat{p}_{no,sunny}\left(1 - \hat{p}_{no,sunny}\right)}{\hat{\sigma}^2_{no,sunny}} - 1 = \frac{0.4 * 0.6}{0.048} - 1 = 4$$

Estimation of Dirichlet Parameters: PlayTennis and Outlook Data (4/4)

- Dirichlet parameters α_i for $PlayTennis = yes$:

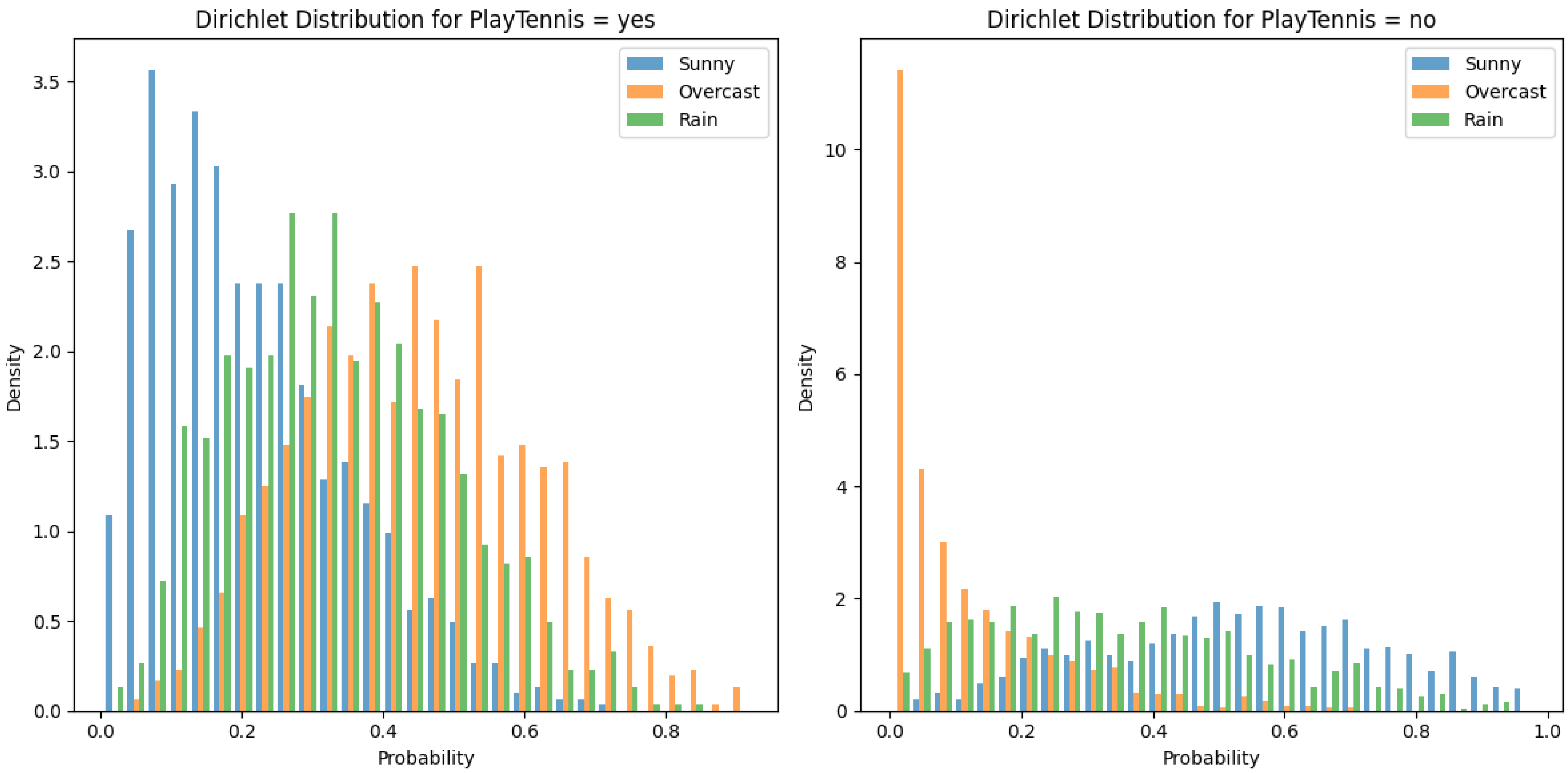
$$\alpha_{yes,sunny} = \hat{p}_{yes,sunny} * \alpha_0 = 0.222 * 8 = 1.78$$
$$\alpha_{yes,overcast} = \hat{p}_{yes,overcast} * \alpha_0 = 0.444 * 8 = 3.55$$
$$\alpha_{yes,rain} = \hat{p}_{yes,rain} * \alpha_0 = 0.333 * 8 = 2.66$$

- Dirichlet parameters α_i for $PlayTennis = no$:

$$\alpha_{no,sunny} = \hat{p}_{no,sunny} * \alpha_0 = 0.6 * 4 = 2.4$$
$$\alpha_{no,overcast} = \hat{p}_{no,overcast} * \alpha_0 = 0 * 4 + \epsilon = 0.5$$
$$\alpha_{no,rain} = \hat{p}_{no,rain} * \alpha_0 = 0.4 * 4 = 1.6$$

where $\epsilon = 0.5$ is used to avoid zero values. While higher values of α mean more confidence in the estimated probabilities, lower values suggest less confidence or more uncertainty in the probabilities.

Dirichlet Distributions for PlayTennis and Outlook Example (1K samples)




```
samples_yes = np.random.dirichlet([1.78, 3.55, 2.66], 1000) # 30 bins
samples_no = np.random.dirichlet([2.4, 0.5, 1.6], 1000) # 30 bins
```

MLE with Dirichlet Parameters

$$P(sunny|yes) = \frac{count(yes, sunny) + \alpha_{yes,sunny}}{count(yes) + \alpha_0(yes)} = \frac{2 + 1.78}{9 + 8} = 0.2223$$

$$P(overcast|yes) = \frac{count(yes, overcast) + \alpha_{yes,overcast}}{count(yes) + \alpha_0(yes)} = \frac{4 + 3.55}{9 + 8} = 0.4441$$

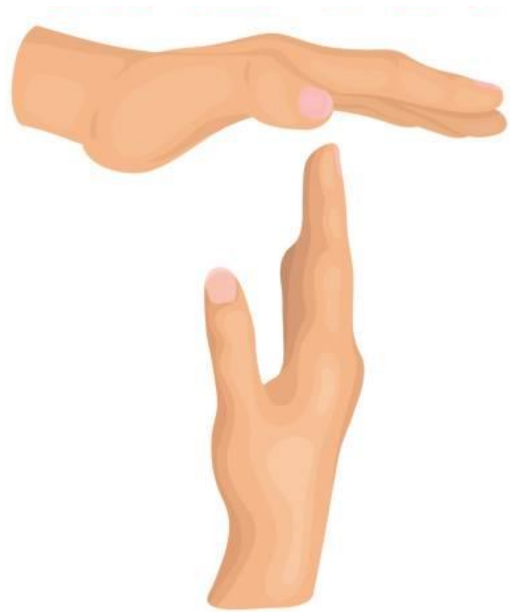
$$P(rain|yes) = \frac{count(yes, rain) + \alpha_{yes,rain}}{count(yes) + \alpha_0(yes)} = \frac{3 + 2.66}{9 + 8} = 0.3329$$

$$P(sunny|no) = \frac{count(no, sunny) + \alpha_{no,sunny}}{count(no) + \alpha_0(no)} = \frac{3 + 2.4}{5 + 4.5} = 0.5684$$

$$P(overcast|no) = \frac{count(no, overcast) + \alpha_{no,overcast}}{count(no) + \alpha_0(no)} = \frac{0 + 0.5}{5 + 4.5} = 0.0526$$

$$P(rain|no) = \frac{count(no, rain) + \alpha_{no,rain}}{count(no) + \alpha_0(no)} = \frac{2 + 1.6}{5 + 4.5} = 0.3789$$

Break



Today

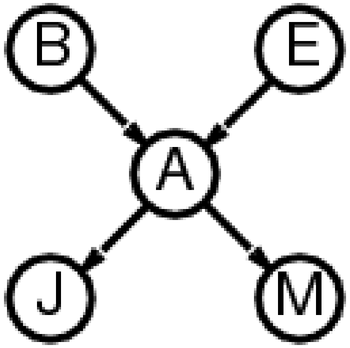
- Introduction to Bayesian networks
  - Graphical representation
  - Probabilistic representation
  - Parameter learning

- **Algorithms for exact inference**
  - Inference by enumeration
  - Inference by variable elimination

Inference by Enumeration

- Sums out variables from the joint without actually constructing its explicit representation.

- Simple query on the burglary network:

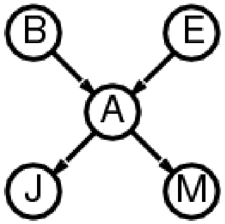


- $P(B|j,m) = \frac{P(B,j,m)}{P(j,m)}$
- $P(B|j,m) = \alpha P(B,j,m)$
- $P(B|j,m) = \alpha \sum_a \sum_e P(B,e,a,j,m)$

Normalisation constant

Inference by Enumeration

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



Rewriting joint entries using product of CPT entries:

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

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

$$= \alpha < P(b|j,m), P(\neg b|j,m) >$$

Inference by Enumeration: $P(b|j,m)$

$$P(b|j,m) = \alpha \sum_a \sum_e P(b)P(e)P(a|b,e)P(j|a)P(m|a)$$
$$= \alpha P(b) \sum_e P(e) \sum_a P(a|b,e)P(j|a)P(m|a)$$

$$= \alpha P(b) \sum_e P(e) [P(a|b,e)P(j|a)P(m|a) + P(\neg a|b,e)P(j|\neg a)P(m|\neg a)]$$

$$= \alpha P(b)[P(e)[P(a|b,e)P(j|a)P(m|a) + P(\neg a|b,e)P(j|\neg a)P(m|\neg a)] + P(\neg e)[P(a|b,\neg e)P(j|a)P(m|a) + P(\neg a|b,\neg e)P(j|\neg a)P(m|\neg a)]]$$

Inference by Enumeration: $P(b|j,m)$

$$P(b|j,m) = \alpha P(b)[P(e)[P(a|b,e)P(j|a)P(m|a) + P(\neg a|b,e)P(j|\neg a)P(m|\neg a)] + P(\neg e)[P(a|b,\neg e)P(j|a)P(m|a) + P(\neg a|b,\neg e)P(j|\neg a)P(m|\neg a)]]$$

$$= \alpha\, [\, 0.001 \times [\, 0.002 \times [0.95 \times 0.9 \times 0.7 + 0.05 \times 0.05 \times 0.01] \;+\; [0.998 \times [0.94 \times 0.9 \times 0.7 + 0.06 \times 0.05 \times 0.01]]]$$

$$= \alpha\, [0.001 \times [\, 0.002 \times [0.5985 \;+\; 0.000025] \;+\; 0.998 \times [0.5922 \;+\; 0.00003]]]$$

$$= \alpha\, [0.001 \times [\, 0.0001197 \;+ 0.591045]]$$

$$= \alpha\, 0.000592243$$

Inference by Enumeration: $P(\neg b|j,m)$

$$P(\neg b|j,m)$$

$$= \alpha \sum_a \sum_e P(\neg b)P(e)P(a|\neg b,e)P(j|a)P(m|a)$$

$$= \alpha P(\neg b) \sum_e P(e) \sum_a P(a|\neg b,e)P(j|a)P(m|a)$$

$$= \alpha P(\neg b) \sum_e P(e) [P(a|\neg b,e)P(j|a)P(m|a) + P(\neg a|\neg b,e)P(j|\neg a)P(m|\neg a)]$$

$$= \alpha P(\neg b)[P(e)[P(a|\neg b,e)P(j|a)P(m|a) + P(\neg a|\neg b,e)P(j|\neg a)P(m|\neg a)] + P(\neg e)[P(a|\neg b,\neg e)P(j|a)P(m|a) + P(\neg a|\neg b,\neg e)P(j|\neg a)P(m|\neg a)]]$$

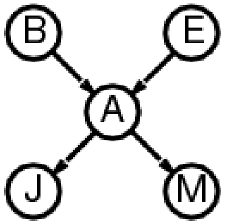
Inference by Enumeration: $P(\neg b|j,m)$

$P(\neg b|j,m) = \alpha P(\neg b)[P(e)[P(a|\neg b,e)P(j|a)P(m|a) +$
$P(\neg a|\neg b,e)P(j|\neg a)P(m|\neg a)] +$
$P(\neg e)[P(a|\neg b,\neg e)P(j|a)P(m|a) +$
$P(\neg a|\neg b,\neg e)P(j|\neg a)P(m|\neg a)]]$

$= \alpha\, [\, 0.999 \times [\, 0.002 \times [0.29 \times 0.9 \times 0.7 + 0.71 \times 0.05$
$\times 0.01] \;+\; [0.998 \times [0.001 \times 0.9 \times 0.7 + 0.999 \times 0.05$
$\times 0.01]]]$
$= \alpha\, [0.999 \times [\, 0.002 \times [0.1827 \;+\; 0.000355] \;+\; 0.998$
$\times [0.00063 \;+ 0.0004995]]]$
$= \alpha\, [0.999 \times [0.00036611 + 0.00112724]]$
$= \alpha\, 0.001491858$

Inference by Enumeration: $P(B|j,m)$

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



Rewriting joint entries using product of CPT entries:

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

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

$$= \alpha < P(b|j,m),\ P(\neg b|j,m) >$$

$$= \alpha < 0.000592243, 0.001491858 >$$

$$=< 0.2842, 0.7158 >$$

$$\propto = \frac{1}{0.000592243 + 0.001491858} = 479.82$$

Inference by Enumeration: *Algorithm*

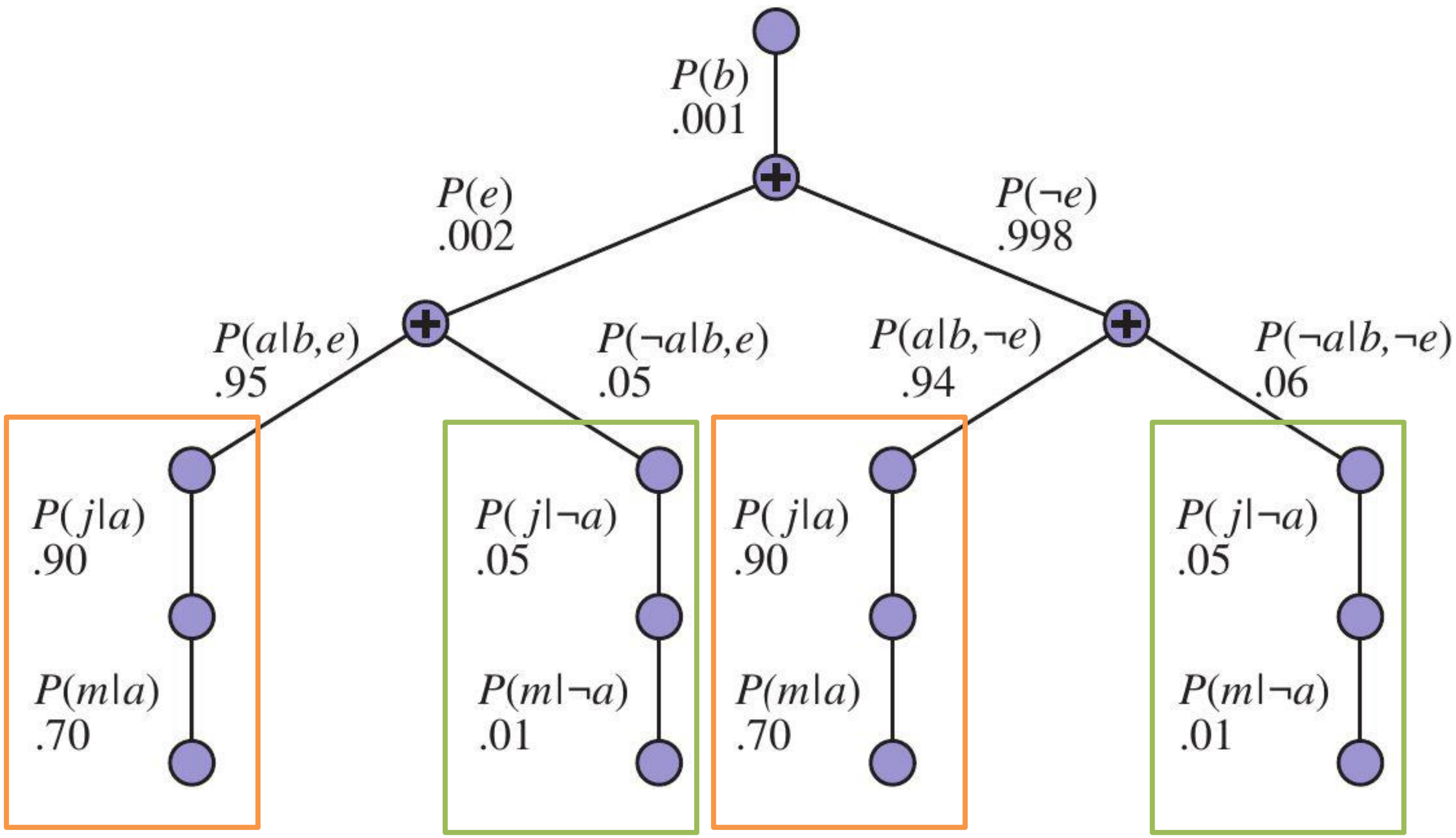
function ENUMERATION-ASK(X, $\mathbf{e}$, bn) **returns** a distribution over X
  inputs: X, the query variable
    $\mathbf{e}$, observed values for variables $\mathbf{E}$
    bn, a Bayesian network with variables $\{X\} \cup \mathbf{E} \cup \mathbf{Y}$

  $\mathbf{Q}(X) \leftarrow$ a distribution over X, initially empty
  for each value x_i of X **do**
    extend $\mathbf{e}$ with value x_i for X
    $\mathbf{Q}(x_i) \leftarrow$ ENUMERATE-ALL(VARS[bn], $\mathbf{e}$)
  return NORMALIZE($\mathbf{Q}(X)$)

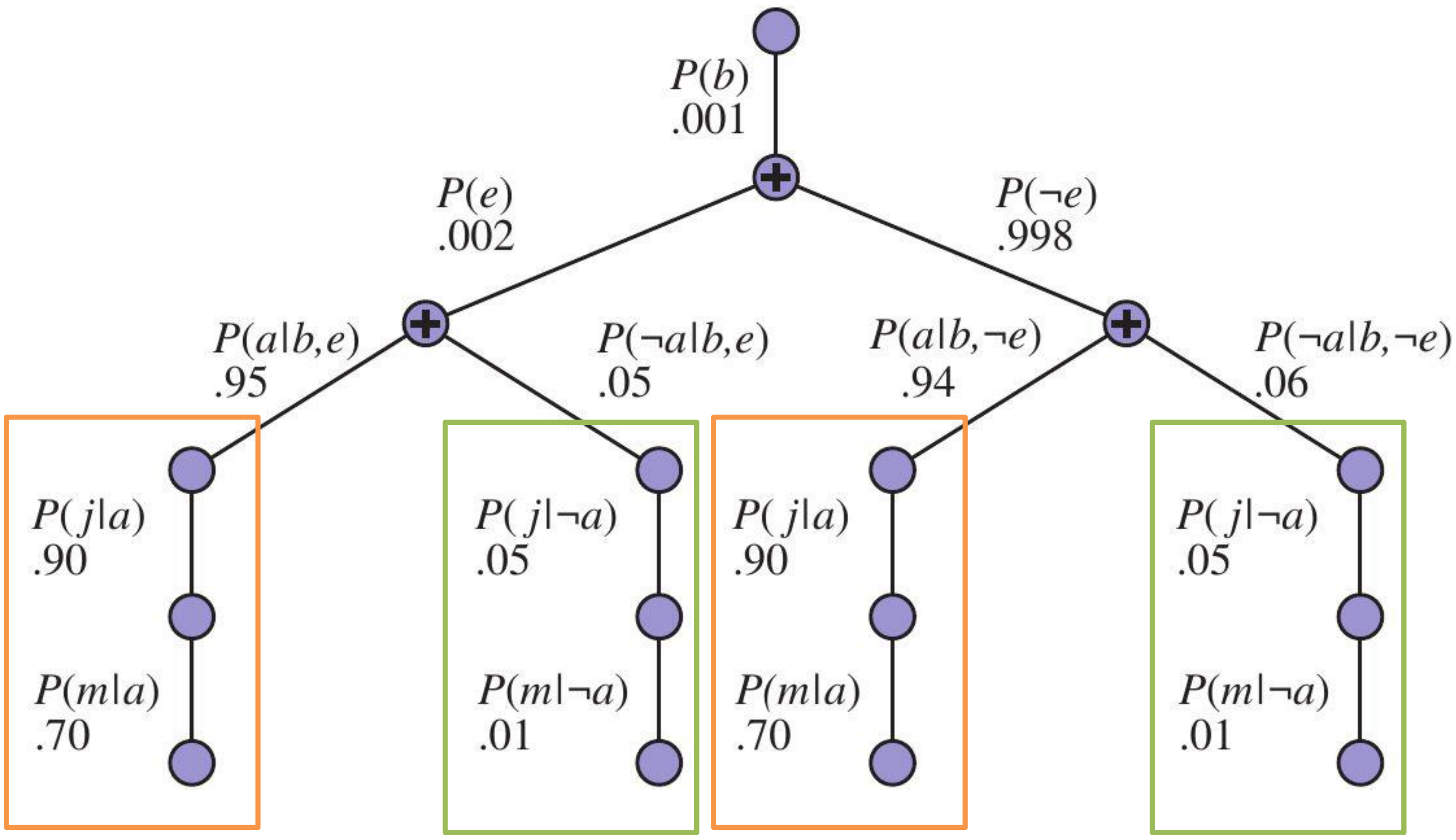
---

function ENUMERATE-ALL($vars$, $\mathbf{e}$) **returns** a real number
  if EMPTY?($vars$) **then return** 1.0
  $Y \leftarrow$ FIRST($vars$)
  if Y has value y in $\mathbf{e}$
    then return $P(y \mid Pa(Y)) \times$ ENUMERATE-ALL(REST($vars$), $\mathbf{e}$)
    else return $\Sigma_y \; P(y \mid Pa(Y)) \times$ ENUMERATE-ALL(REST($vars$), $\mathbf{e}_y$)
      where $\mathbf{e}_y$ is $\mathbf{e}$ extended with $Y = y$

Inference by Enumeration: Evaluation Tree



Inference by Enumeration: Evaluation Tree



Enumeration is inefficient due to repeated computation

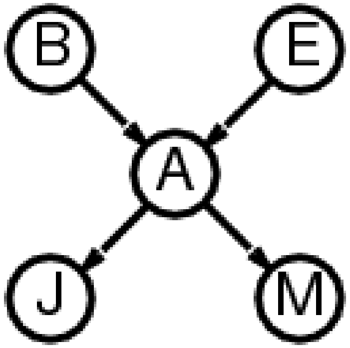
Today

- Introduction to Bayesian networks
  - Graphical representation
  - Probabilistic representation
  - Parameter learning

- Algorithms for exact inference
  - Inference by enumeration
  - **Inference by variable elimination**

Inference by Variable Elimination

- Idea:
  - do the calculation once &
  - save the results for later use



- Variable elimination evaluates expressions in right-to-left order, and uses factors f_i (matrices) as follows:

$$P(B|j,m) = \alpha \underbrace{P(B)}_{f_1(B)} \sum_e \underbrace{P(e)}_{f_2(E)} \sum_a \underbrace{P(a|b,e)}_{f_3(A,B,E)} \underbrace{P(j|a)}_{f_4(A)} \underbrace{P(m|a)}_{f_5(A)}$$

Inference by Variable Elimination

$$f_4(A) = < P(j|a), P(j|\neg a) = < 0.90, 0.05 >$$
$$f_5(A) = < P(m|a), P(m|\neg a) = < 0.70, 0.01 >$$

Therefore, $P(B|j,m) =$

$$\alpha f_1(B) \times \sum_e f_2(E) \times \underbrace{\sum_a f_3(A,B,E) \times f_4(A) \times f_5(A)}_{f_6(B,E)},$$

where $\times$ denotes a pointwise product operation.

$$f_6(B,E) = \sum_a f_3(A,B,E) \times f_4(A) \times f_5(A)$$
$$= [f_3(a,B,E) \times f_4(a) \times f_5(a)] + [f_3(\neg a,B,E) \times f_4(\neg a) \times f_5(\neg a)]$$

Inference by Variable Elimination

Therefore, $P(B|j,m) = \alpha f_1(B) \times \underbrace{\sum_e f_2(E) \times f_6(B,E)}_{f_7(B)}$

Summing out E we get:

$$f_7(B) = \sum_e f_2(E) \times f_6(B,E)$$
$$= [f_2(e) \times f_6(b,e)] + [f_2(\neg e) \times f_6(b,\neg e)]$$

Thus, $P(B|j,m) = \alpha f_1(B) \times f_7(B)$

We only need to know how to do operations with factors!

Pointwise Product with Factors

A	B	$\mathbf{f}_1(A,B)$	B	C	$\mathbf{f}_2(B,C)$	A	B	C	$\mathbf{f}_3(A,B,C)$
T	T	.3	T	T	.2	T	T	T	$.3 \times .2 = .06$
T	F	.7	T	F	.8	T	T	F	$.3 \times .8 = .24$
F	T	.9	F	T	.6	T	F	T	$.7 \times .6 = .42$
F	F	.1	F	F	.4	T	F	F	$.7 \times .4 = .28$
						F	T	T	$.9 \times .2 = .18$
						F	T	F	$.9 \times .8 = .72$
						F	F	T	$.1 \times .6 = .06$
						F	F	F	$.1 \times .4 = .04$

Figure 14.10 Illustrating pointwise multiplication: $\mathbf{f}_1(A,B) \times \mathbf{f}_2(B,C) = \mathbf{f}_3(A,B,C)$.

Operations on Factors

A	B	$\mathbf{f}_1(A,B)$	B	C	$\mathbf{f}_2(B,C)$	A	B	C	$\mathbf{f}_3(A,B,C)$
T	T	.3	T	T	.2	T	T	T	$.3 \times .2 = .06$
T	F	.7	T	F	.8	T	T	F	$.3 \times .8 = .24$
F	T	.9	F	T	.6	T	F	T	$.7 \times .6 = .42$
F	F	.1	F	F	.4	T	F	F	$.7 \times .4 = .28$
						F	T	T	$.9 \times .2 = .18$
						F	T	F	$.9 \times .8 = .72$
						F	F	T	$.1 \times .6 = .06$
						F	F	F	$.1 \times .4 = .04$

Figure 14.10 Illustrating pointwise multiplication: $\mathbf{f}_1(A,B) \times \mathbf{f}_2(B,C) = \mathbf{f}_3(A,B,C)$.

$$f(Y,Z) = \sum_x f(X,Y,Z) = f(x,Y,Z) + f(\neg x,Y,Z)$$
$$= \begin{pmatrix} 0.06 & 0.24 \\ 0.42 & 0.28 \end{pmatrix} + \begin{pmatrix} 0.18 & 0.72 \\ 0.06 & 0.04 \end{pmatrix} = \begin{pmatrix} 0.24 & 0.96 \\ 0.48 & 0.32 \end{pmatrix}$$

Inference by Variable Elimination: Full Example

$$P(B|j,m) = \alpha \underbrace{P(B)}_{f_1(B)} \sum_e \underbrace{P(e)}_{f_2(E)} \sum_a \underbrace{P(a|b,e)}_{f_3(A,B,E)} \underbrace{P(j|a)}_{f_4(A)} \underbrace{P(m|a)}_{f_5(A)}$$

$$= \alpha f_1(B) \times \sum_e f_2(E) \times \sum_a \underbrace{f_3(A,B,E) \times f_4(A) \times f_5(A)}_{f_6(B,E)}$$

$$f_6(B,E) =$$
$$= [f_3(a,B,E) \times f_4(a) \times f_5(a)] + [f_3(\neg a,B,E) \times f_4(\neg a) \times f_5(\neg a)]$$

$$= \begin{pmatrix} B & E & f_3 \\ t & t & 0.95 \\ t & f & 0.94 \\ f & t & 0.29 \\ f & f & 0.001 \end{pmatrix} \times 0.63 + \begin{pmatrix} B & E & f_4 \\ t & t & 0.05 \\ t & f & 0.06 \\ f & t & 0.71 \\ f & f & 0.94 \end{pmatrix} \times 0.0005 = \begin{pmatrix} B & E & f_6 \\ t & t & 0.59852 \\ t & f & 0.59222 \\ f & t & 0.18305 \\ f & f & 0.00110 \end{pmatrix}$$

Inference by Variable Elimination: Full Example

$$f_7(B) = [f_2(e) f_6(B,e)] + [f_2(\neg e) f_6(B,\neg e)]$$

$$= 0.002 \times \begin{pmatrix} B & f_6 \\ t & 0.59852 \\ f & 0.18305 \end{pmatrix} + 0.998 \begin{pmatrix} B & f_6 \\ t & 0.59222 \\ f & 0.00110 \end{pmatrix} = \begin{pmatrix} B & f_7 \\ t & 0.59223 \\ f & 0.00146 \end{pmatrix}$$

$$P(B|j,m) = \alpha f_1(B) \times f_7(B)$$

$$= \alpha \begin{pmatrix} B & f_1 \\ t & 0.001 \\ f & 0.999 \end{pmatrix} \times \begin{pmatrix} B & f_7 \\ t & 0.59223 \\ f & 0.00146 \end{pmatrix} = \alpha \begin{pmatrix} & P(B|j,m) \\ t & 0.000592 \\ f & 0.001458 \end{pmatrix}$$

$$= < 0.289, 0.711 >$$

$$\alpha = \frac{1}{0.000592 + 0.001458}$$

Variable Elimination: $Algorithm$

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

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

Homework (recommended)

1. Calculate $P(E|j,m)$ using inference by enumeration with pen and paper

2. Calculate $P(E|j,m)$ using variable elimination with pen and paper

Ideas for your Assignment (optional)

1. Implement Dirichlet priors with moment matching in the code of today’s workshop (*program that does parameter learning*)

2. Implement the Variable Elimination algorithm in the code of today’s workshop (*program that does probabilistic inference*)

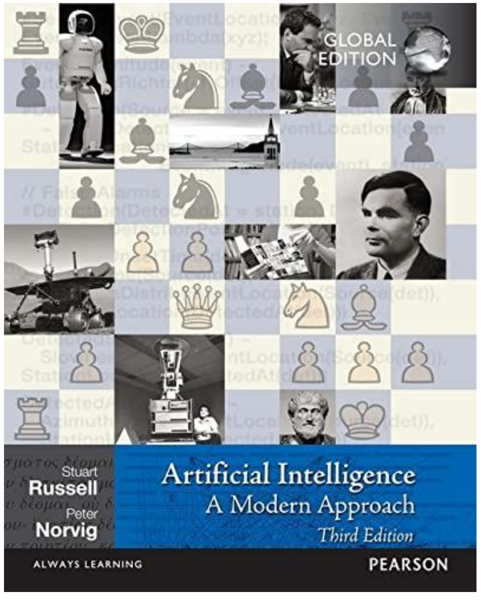
Today

- Introduction to Bayesian networks
- Parameter learning via Max. Likelihood Est.
- Inference by enumeration
- Inference by variable elimination

Readings:

Russell & Norvig 2016. [Chapters 14-14.4]

Koller & Friedman 2009. [Section 17.3.2]



This and Next Week

Workshop (today):

Exercises using Bayesian networks

Python program for exact inference

Lecture (next week):

Structure Learning for Bayesian Networks

Reading: [Kitson et al. A survey on Bayesian Network structure learning, 2023]

Questions?