

2019 – Bayesian Networks

Basics of Probability Theory

University of Augsburg

Multimedia Computing and Computer Vision

Prof. Dr. Rainer Lienhart

Rainer.Lienhart@informatik.uni-augsburg.de

www.multimedia-computing.org

Reference

Richard E. Neapolitan. **Learning Bayesian Networks.** *Prentice Hall Series in Artificial Intelligence*, ISBN 0-13-012534-2.

Don't forget. Reading the book chapters 1 – 6 is mandatory.

Chapter on ***Basics of Probability Theory***
(chapter 1)

Figures and text are taken from that book

Introduction to BNs (1)

- Presence / absence of a disease in a human being has direct influence on whether a test for that disease turns out positive or negative

$$P(\text{Test} = \text{positive} \mid \text{Disease} = \text{present})$$
$$P(\text{Test} = \text{negative} \mid \text{Disease} = \text{absent})$$

Abbreviated form:

$$P(T = + \mid D = +)$$
$$P(T = - \mid D = -)$$

Even shorter:

$$\begin{matrix} P(T \mid D) \\ P(\neg T \mid \neg D) \end{matrix}$$

⇒ **Bayes' theorem**: compute the conditional probability of an individual having a disease when a test for the disease comes back positive.

$$P(D \mid T) = \frac{P(T \mid D)P(D)}{P(T)}$$

Introduction to BNs: Example HIV-Test (2)

- According to Robert-Koch-Institut ~80,000 infected people in Germany:

$$P(HIV) = \frac{80,000}{80\text{ Mio}} = 0.001$$

- A standard HIV test is the Elisa HIV Test

$$P(T | HIV) = \frac{997}{1000} = 0.997 = 99.7\%$$

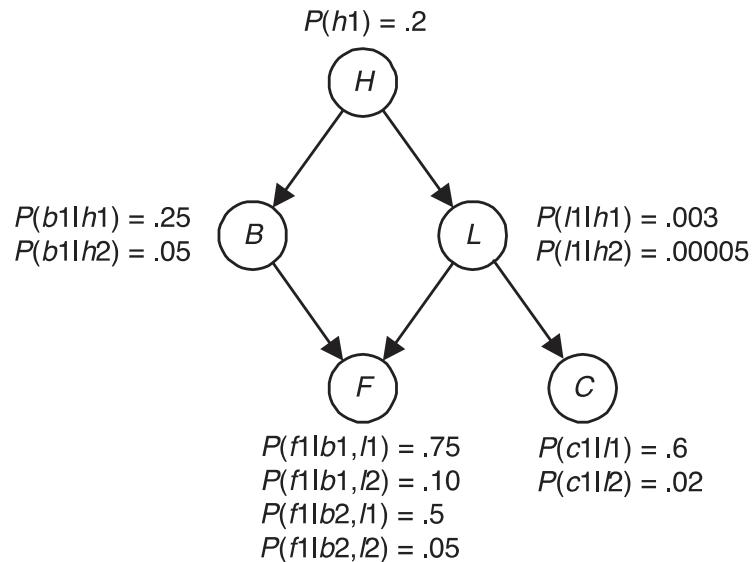
$$P(\neg T | \neg HIV) = \frac{985}{1000} = 0.985 = 98.5\%$$

⇒ [Bayes' theorem](#): Compute the conditional probability of an individual having HIV when a test for the disease comes back positive.

$$\begin{aligned} P(HIV | T) &= \frac{P(T|HIV)P(HIV)}{P(T)} \\ &= \frac{0.997 \cdot 0.001}{P(T|HIV)P(HIV) + P(T|\neg HIV)P(\neg HIV)} \\ &= \frac{0.997 \cdot 0.001}{0.997 \cdot 0.001 + 0.015 \cdot 0.999} \\ &= 0.062 \end{aligned}$$

⇒ This result is never reported back to the patient, but a second test is performed.

Introduction to BNs (3)



Feature	Value	When the Feature Takes this Value
<i>H</i>	<i>h1</i>	There is a history of smoking
	<i>h2</i>	There is no history of smoking
<i>B</i>	<i>b1</i>	Bronchitis is present
	<i>b2</i>	Bronchitis is absent
<i>L</i>	<i>l1</i>	Lung cancer is present
	<i>l2</i>	Lung cancer is absent
<i>F</i>	<i>f1</i>	Fatigue is present
	<i>f2</i>	Fatigue is absent
<i>C</i>	<i>c1</i>	Chest X-ray is positive
	<i>c2</i>	Chest X-ray is negative

- **Goal:** Do probabilistic inference involving features that are not directly related via a direct influence, e.g., $P(B | H, F, C) = ?$ or $P(L | H, F, C) = ?$
- ⇒ **Brute force:** requires full joint probability $P(H, B, L, F, C)$

$$P(B | H, F, C) = \frac{P(B, H, F, C)}{P(H, F, C)} = \frac{\sum_l P(H, B, l, F, C)}{\sum_b \sum_l P(H, b, l, F, C)}$$
 - Ordinarily not available prob. distribution
 - Exponential space and time complexity
- ⇒ **Bayesian Networks** can answer questions in less than exponential space and time complexity!

Set-Theoretic Definition of Probability (1)

Definition is based on experiments with a set of distinct **outcomes**. The collection of all outcomes is called **sample space**, and the outcomes are the elements of the set. In case of a finite sample space, every subset of the sample space is called an **event**. A subset containing exactly one element is called an **elementary event**.

Definition 1.1: Suppose we have a sample space Ω containing n distinct elements: $\Omega = \{e_1, e_2, \dots, e_n\}$. A function that assigns a real number $P(E)$ to each event $E \subseteq \Omega$ is called a **probability function** on the set of subsets of Ω if it satisfies the following conditions:

1. $0 \leq P(e_i) \leq 1$ for $1 \leq i \leq n$.
2. $P(e_1) + P(e_2) + \dots + P(e_n) = 1$.
3. For each non-elementary event $E = \{e_{i_1}, e_{i_2}, \dots, e_{i_k}\}$ we have $P(E) = P(\{e_{i_1}\}) + P(\{e_{i_2}\}) + \dots + P(\{e_{i_k}\})$

The pair (Ω, P) is called a **probability space**.

Principle of indifference: Elementary events are to be considered equiprobable if we have no reason to expect or prefer one over the other.

Set-Theoretic Definition of Probability (2)

Theorem 1.1: Let (Ω, P) be a probability space. Then

1. $P(\Omega) = 1$.
2. $0 \leq P(E) \leq 1 \quad \forall E \subseteq \Omega$
3. For $E, F \subseteq \Omega$ such that $E \cap F = \emptyset$, then $P(E \cup F) = P(E) + P(F)$.

= 1.-3. called the *axioms of probability theory*.

In general: $P(E \cup F) = P(E) + P(F) - P(E \cap F)$.

(see book example 1.2, p.5)

Conditional Prob. & Independence (1)

Definition 1.2: Let E and F be events such that $P(F) \neq 0$. Then the **conditional probability** of E given F , denoted $P(E|F)$, is given by

$$P(E|F) = \frac{P(E \cap F)}{P(F)}$$

$P(E|F)$ means the probability of E occurring given that we know F has occurred.

⇒ Fraction of items in F that are also in E

$$P(E|F) = \frac{P(E \cap F)}{P(F)} = \frac{n_{EF}/n}{n_F/n}$$

with $n = |\Omega|$, $n_F = |F|$, $n_{EF} = |E \cap F|$.

Definition 1.3: Two events E and F are **independent** if one of the following holds:

1. $P(E|F) = P(E)$ and $P(E) \neq 0, P(F) \neq 0$.
2. $P(E) = 0$ or $P(F) = 0$.

Note:

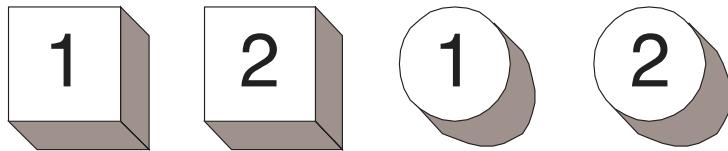
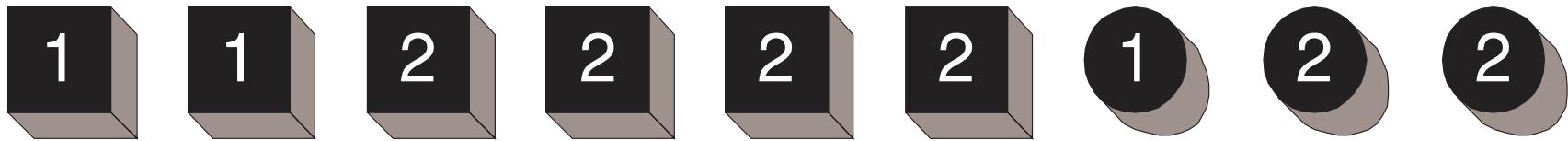
- Independence is symmetric, i.e., if $P(F) \neq 0$ and $P(E) \neq 0$, then $P(E|F) = P(E)$ iff $P(F|E) = P(F)$.
- E and F independent iff $P(E \cap F) = P(E)P(F)$

Conditional Prob. & Independence (2)

Definition 1.4: Two events E and F are *conditionally independent* given G if $P(G) \neq 0$ and one of the following holds:

1. $P(E|F \cap G) = P(E|G)$ and $P(E|G) \neq 0, P(F|G) \neq 0$.
2. $P(E|G) = 0$ or $P(F|G) = 0$.

Book Example 1.5



$$P(\text{One}) = 5/13$$

$$P(\text{One}|\text{Square}) = 3/8$$

$$P(\text{One}|\text{Black}) = 3/9 = \textcolor{red}{1/3}$$

$$P(\text{One}|\text{Square} \cap \text{Black}) = 2/6 = \textcolor{red}{1/3}$$

$$P(\text{One}|\text{White}) = 2/4 = \textcolor{blue}{1/2}$$

$$P(\text{One}|\text{Square} \cap \text{White}) = \textcolor{blue}{1/2}$$

Theorem 1.2

Law of total probability: Suppose we have n mutually exclusive and exhaustive events E_i , i.e., E_1, \dots, E_n such that $E_i \cap E_j = \emptyset \forall i \neq j$ and $E_1 \cup \dots \cup E_n = \Omega$. Then the law of total probability says that for any other event F ,

$$P(F) = \sum_{i=1}^n P(F \cap E_i)$$

If $P(E_i) \neq 0$, then $P(F \cap E_i) = P(F|E_i)P(E_i)$. Therefore, if $P(E_i) \neq 0 \forall i$, we get:

$$P(F) = \sum_{i=1}^n P(F|E_i) P(E_i)$$

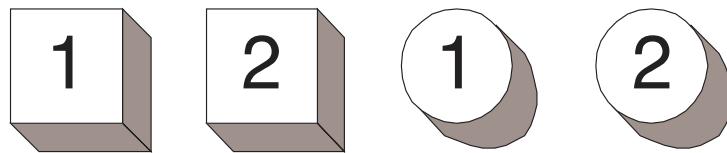
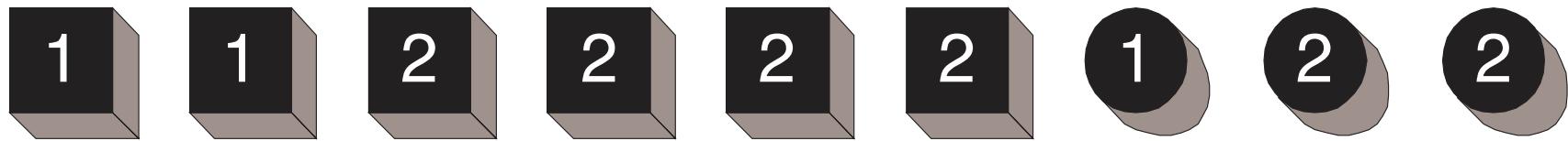
Theorem 1.2 (Bayes): Given two events E and F such that $P(E) \neq 0$ and $P(F) \neq 0$, we have

$$P(E|F) = \frac{P(F|E)P(E)}{P(F)}$$

Furthermore, given n mutually exclusive and exhaustive events E_1, \dots, E_n such that $P(E_i) \neq 0 \forall i$, we have for $1 \leq i \leq n$,

$$P(E_i|F) = \frac{P(F|E_i)P(E_i)}{P(F)} = \frac{P(F|E_i)P(E_i)}{\sum_{i=1}^n P(F|E_i)P(E_i)}$$

Example 1.6



$$\begin{aligned} P(One|Black) &= \frac{P(Black|One)P(One)}{P(Black|One)P(One) + P(Black|Two)P(Two)} \\ &= \frac{(-)(-)}{(-)(-) + (-)(-)} = \frac{1}{3} \\ &= \frac{P(Black \cap One)}{P(Black)} = \frac{/13}{/13} \end{aligned}$$

Random Variable

Definition 1.5: Given a probability space (Ω, P) , a **random variable** X is a function on Ω :

$$X: \Omega \rightarrow ?$$

Thus, a random variable assigns a unique value to each element (outcome) in the sample space. The set of values a random variable X can assume is called the **space of X** .

Example 1.7: Let $\Omega = \{(1,1), (1,2), \dots, (1,6), (2,1), \dots, (6,6)\}$ = all outcomes of a throw of a pair of six-sided dice. Let $X :=$ sum of each ordered pair. Then the space of X is $\{2,3,4,5,6,7,8,9,10,11,12\}$.

$X=x$ denotes the set of all elements $e \in \Omega$ that X maps to the value of x : $\{e \in \Omega | X(e) = x\}$.

Probability distribution of the random variable X :

$$P_X(\{x\}) = P(X = x) = P(x)$$

We also denote $P(x)$ as “the probability of x ”.

Expected Value of X: E(X)

Expected value of X:

$$E(X) = \sum_x xP(x)$$

Example 1.10: Throw of a dice. Define Ω , define P, X assigns the value of each outcome to each outcome:

$$E(X) =$$

Joint Probability, Marginals

Given two random variables X and Y , defined on the sample space Ω , we use $X = x, Y = y$ to denote the set of all elements $e \in \Omega$ that are mapped both by X to x and by Y to y .

$$X = x, Y = y \equiv \{e \in \Omega | X(e) = x\} \cap \{e \in \Omega | Y(e) = y\}$$

Joint probability distribution of X and Y : $P(X = x, Y = y) = P(x, y)$

Note: For consistency $P(\emptyset = \emptyset) = 1$ and $P(\emptyset) = 0$.

Empty set of random variables Empty set of events

⇒ With $A = \{X, Y\}$, $a = \{x, y\}$, we can also write

$$P(a) = P(A = a) = P(X = x, Y = y)$$

⇒ With $A = \{X, Y\}$, $B = \{Z, W\}$ then

$A = a, B = b$ represents $X = x, Y = y, Z = z, W = w$

Marginal probability distribution of X given $P(x, y)$:

$$P(X = x) = \sum_y P(X = x, Y = y)$$

⇒ extends to more random variables.

Example: Joint & Marginal Probability Distribution

Given some population of students of a CS class:

		Intelligence		
		low	high	
Grade	A	0.07	0.18	0.25
	B	0.28	0.09	0.37
	C	0.35	0.03	0.38
		0.7	0.3	1

(Conditional) Independence

Definition 1.6: Suppose we have a probability space (Ω, P) , and two sets A and B containing random variables defined on Ω . Then the sets A and B are said to be **independent** if, for all values of the variables in the sets a and b , the events $A = a$ and $B = b$ are independent. That is,

- Either $P(a) = 0$ or $P(b) = 0$ or
- $P(a|b) = P(a)$.

When this is the case, we write $I_P(A, B)$ where I_P stands for independent in P .

Definition 1.7: Suppose we have a probability space (Ω, P) , and three sets A , B and C containing random variables defined on Ω . Then the sets A and B are said to be **conditionally independent given set C** if, for all values of the variables in the sets a , b and c , whenever $P(c) \neq 0$, the events $A = a$ and $B = b$ are conditionally independent given the event $C = c$. That is,

- Either $P(a|c) = 0$ or $P(b|c) = 0$ or
- $P(a|b, c) = P(a|c)$.

When this is the case, we write $I_P(A, B|C)$.

Some Independence Properties

Symmetry:

$$I_P(X, Y|Z) \Rightarrow I_P(Y, X|Z)$$

Decomposition:

$$I_P(X, \{Y, W\}|Z) \Rightarrow I_P(X, Y|Z)$$

Weak Union:

$$I_P(X, \{Y, W\}|Z) \Rightarrow I_P(X, Y|\{Z, W\})$$

Contraction:

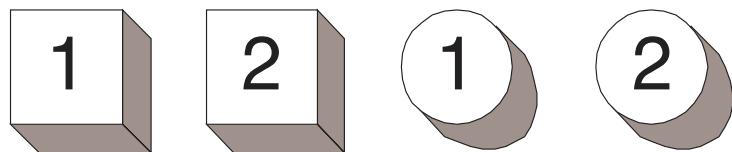
$$I_P(X, W|Z, Y) \& I_P(X, Y|Z) \Rightarrow I_P(X, \{Y, W\}|Z)$$

Intersection:

For positive distributions and for mutually disjoint sets X, Y, Z, W :

$$I_P(X, Y|\{Z, W\}) \& I_P(X, W|Z, Y) \Rightarrow I_P(X, \{Y, W\}|Z)$$

Example 1.17



Variable	Value	Outcomes Mapped to this Value
V	v1	All objects containing a "1"
	v2	All objects containing a "2"
S	s1	All square objects
	s2	All round objects
C	c1	All black objects
	c2	All white objects

Show $I_P(\{V\}, \{S\} | \{C\})$ (short form $I_P(V, S | C)$)

Chain rule

Given two random variables X_1 and X_2 defined on the same sample space Ω , then

$$\begin{aligned} P(x_1, x_2) &= P(x_2|x_1) \cdot P(x_1) \\ &= P(x_1|x_2) \cdot P(x_2) \end{aligned}$$

whenever $P(x_1, x_2) \neq 0$.

In general:

Given n random variables X_1, X_2, \dots, X_n defined on the same sample space Ω , then

$$P(x_1, x_2, \dots, x_n) = P(x_n|x_1, \dots, x_{n-1}) \cdot P(x_{n-1}|x_1, \dots, x_{n-2}) \cdot \dots \cdot P(x_2|x_1) \cdot P(x_1)$$

whenever $P(x_1, x_2, \dots, x_n) \neq 0$.

Probabilities as Rel. Frequencies (1)

⇒ Basis for learning from data

Given a (infinitely) repeatable identical experiment, e.g. n tosses of a coin or thumbtack

⇒ Relative frequency of each outcome approaches a limit.

$$P(\text{heads}) = \lim_{n \rightarrow \infty} \frac{\#\text{heads}}{n}$$

⇒ That limit is called the probability of the outcome.

Note: The infinite sequence only exists in theory.

How are relative frequencies related to ratios? = Expected limit of random experiments (e.g., drawing a card from a shuffled deck of cards; expected frequency of each card approaches ratio)

Relative frequencies are only defined relative to a random process.

Random process = repeatable experiment for which the infinite sequence of outcomes is assumed to be a random sequence.

Probabilities as Rel. Frequencies (2)

Random process --> Trials are probabilistically independent. The 1st to the $(n-1)$ th trial does not say anything about the n th trial.

$$P(x_n | x_{n-1}, \dots, x_1) = P(x_n)$$

iid = Independent and identically distributed

Sampling := Estimating a relative frequency for a given collective from a finite set of observations. The collective can be finite population (sampling with replacement) or an infinitely large population.

Weak law of large numbers = confidence in estimate for p given m independent trials with k successes:

Given $\epsilon, \delta > 0$, then

$$P\left(\left|p - \frac{k}{m}\right| < \epsilon\right) > 1 - \delta \quad \text{for} \quad m > \frac{2}{\delta\epsilon^2}$$

Rel. Frequency Approach to Prob. (3)

Goal: Determine confidence interval of estimate for p given m independent trials with k successes.

Given:

- Confidence level β with $0 \leq \beta \leq 1$
- Number m of independent trials
- Number k of successes

then

(θ_1, θ_2) is a β % confidence interval for p , i.e. β % of the time the interval generated will contain p .

with

$$\alpha = \frac{(1 - \beta)}{2}$$
$$\theta_1 = \frac{kF_\alpha(2k, 2[m - k + 1])}{(m - k + 1) + kF_\alpha(2k, 2[m - k + 1])}$$
$$\theta_2 = \frac{k}{(m - k + 1)F_{1-\alpha}(2[m - k + 1], 2k) + k}$$

where F is the F distribution.

<http://www.itl.nist.gov/div898/handbook/eda/section3/eda3673.htm>

http://en.wikipedia.org/wiki/F_distribution

Subjective / Bayesian Probabilities

Example: Given a soccer game between team A and B, how probable is it that team A wins?

- ⇒ The probability represents our belief only!
- ⇒ Not repeatable!
- ⇒ Two different people might have two different beliefs!
- ⇒ There are no objective values.

Subjective probabilities are called “Bayesian” because its proponents use Bayes' theorem to infer unknown probabilities from known ones.

Prior probability := probability of some event **prior to updating** its probability using new information.

Posteriori probability := probability of some event **after updating** its probability based on new information.

Example 1.22 (1)

Joe has a routine diagnostic chest X-ray to test for lung cancer. The test comes back positive. Does he have lung cancer?

The test has a false negative rate of **0.4** and a false positive rate of **0.02**:

$$P(\text{Test} = + | \text{LungCancer} = +) = 0.6$$

$$P(\text{Test} = - | \text{LungCancer} = +) = \textcolor{blue}{0.4}$$

$$P(\text{Test} = - | \text{LungCancer} = -) = 0.98$$

$$P(\text{Test} = + | \text{LungCancer} = -) = \textcolor{red}{0.02}$$

Joe wants to know is

$$P(\text{LungCancer} = + | \text{Test} = +) = ?$$

Example 1.22 (2)

Recall Bayes Formula:

$$\begin{aligned} P(\text{LungCancer} = + | \text{Test} = +) &= \frac{P(\text{Test} = + | \text{LungCancer} = +)P(\text{LungCancer} = +)}{P(\text{Test} = +)} \\ &= \frac{P(\text{Test} = + | \text{LungCancer} = +)P(\text{LungCancer} = +)}{\sum_{l \in \{-, +\}} P(\text{Test} = +, \text{LungCancer} = l)} \\ &= \frac{P(\text{Test} = + | \text{LungCancer} = +)P(\text{LungCancer} = +)}{P(\text{Test} = + | \text{LungCancer} = +)P(\text{LungCancer} = +) + P(\text{Test} = + | \text{LungCancer} = -)P(\text{LungCancer} = -)} \end{aligned}$$

Assume 1 out of a 1000 have lung cancer:

$$P(\text{LungCancer} = +) = 0.001$$

$$P(\text{LungCancer} = -) = 0.999$$

Then we get:

$$\begin{aligned} P(\text{LungCancer} = + | \text{Test} = +) &= \frac{(.6)(.001)}{(.6)(.001) + (.02)(.999)} \\ &= .029 \end{aligned}$$

2nd Approach to Random Variables

In statistics, random variables are usually not used as described so far:

- Identify sample space
- Determine probabilities of elementary events
- Determine random variables, and then
- Compute values in joint distributions.

Instead:

- Random variables are identified directly.
- Their Cartesian product of the set of all possible values of the random variables defines an implicit sample space.
- Consider each random variable as a function on this space that maps each tuple into the value of the random variable in the tuple.
- Define a joint probability distribution (see next slide on “Joint Prob. Distribution”)

Random variable X = a symbol representing any one of a set of values, called the space of X .

Joint Prob. Distribution

Definition 1.8: Let a set of n random variables $V = \{X_1, X_2, \dots, X_n\}$ be specified such that each X_i has a countably infinite space. A function, that assigns a real number $P(X_1 = x_1, X_2 = x_2, \dots, X_n = x_n)$ to every combination of values of the x_i 's such that the value of x_i is chosen from the space of X_i , is called **a joint probability distribution** of the random variables in V if it satisfies the following conditions:

1. For every combination of values of the x_i 's

$$0 \leq P(X_1 = x_1, X_2 = x_2, \dots, X_n = x_n) \leq 1$$

2. We have

$$\left[\sum_{x_1, x_2, \dots, x_n} P(X_1 = x_1, X_2 = x_2, \dots, X_n = x_n) \right] =$$
$$\left[\sum_{x_1} \dots \sum_{x_n} P(X_1 = x_1, X_2 = x_2, \dots, X_n = x_n) \right] = 1$$

Example 1.26

Let $\mathbb{V} = \{X, Y\}$, let X and Y have spaces $\{x1, x2\}^1$ and $\{y1, y2\}$ respectively, and let the following values be specified:

$$\begin{array}{ll} P(X = x1) = .2 & P(Y = y1) = .3 \\ P(X = x2) = .8 & P(Y = y2) = .7. \end{array}$$

Next define a joint probability distribution of X and Y as follows:

$$P(X = x1, Y = y1) = P(X = x1)P(Y = y1) = (.2)(.3) = .06$$

$$P(X = x1, Y = y2) = P(X = x1)P(Y = y2) = (.2)(.7) = .14$$

$$P(X = x2, Y = y1) = P(X = x2)P(Y = y1) = (.8)(.3) = .24$$

$$P(X = x2, Y = y2) = P(X = x2)P(Y = y2) = (.8)(.7) = .56.$$

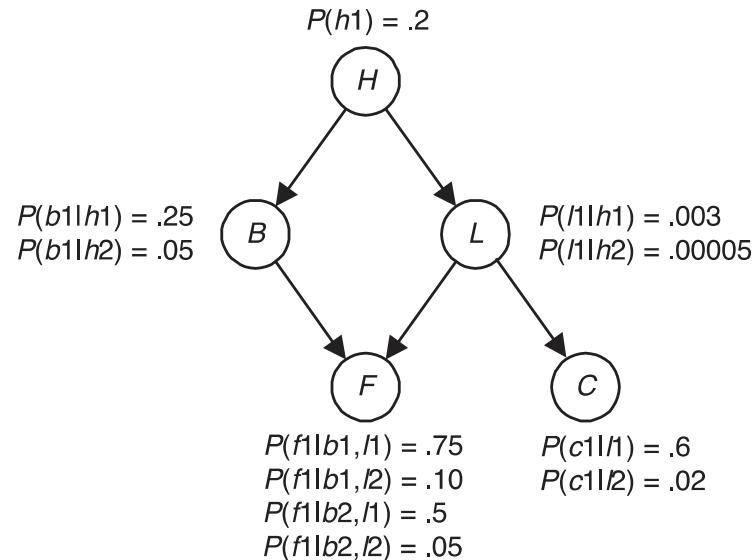
Since the values sum to 1, this is another way of specifying a joint probability distribution according to Definition 1.8. This is how we would specify the joint distribution if we felt X and Y were independent.

Bayesian Networks (1)

Notation: **Random variables** = capital letters (e.g., X, Y);
Values of random variable X = lower case letters x1, x2, and so forth.

Bayesian inference

- Fairly simple when it involves only two related variables.
- Much more complex when doing inference with many related variables.



Task: Do inference involving features that are not related via a direct influence,

e.g.,

$$P(B | H, F, C)=?$$

or

$$P(L | H, F, C)=?$$

Bayesian Networks (2)

Impractical solution:

- Assumes we know the joint probability distribution over the five variables:

$$P(b, h, f, c, l)$$

- Using then marginal distributions we get:

$$P(b_1|h_1, f_1, c_1) = \frac{P(b_1, h_1, f_1, c_1)}{P(h_1, f_1, c_1)} = \frac{\sum_l P(b, h, f, c, l)}{\sum_{b,l} P(b, h, f, c, l)}$$

Problems:

1. Generally joint probabilities are not available
2. 100 binary variables requires 2^{97} summations! The # of terms in a joint probability distribution is exponential in terms of the # of variables.

Directed Graphs

A **directed graph** is a pair (V, E) , where V is a finite, nonempty set whose elements are called **nodes** (or vertices), and E is a set of ordered pairs of distinct elements of V . Elements of E are called **edges** (or arcs), and if $(X, Y) \in E$, we say there is an edge from X to Y . If there is an edge from X to Y or from Y to X , we say X and Y are **adjacent**.

Suppose we have a set of nodes $\{X_1, X_2, \dots, X_k\}$, where $k \geq 2$, such $(X_{i-1}, X_i) \in E$ for $2 \leq i \leq k$. We call the set of edges connecting the k nodes a **path** from X_1 to X_k . The nodes X_2, \dots, X_{k-1} are called interior nodes on path $[X_1, X_2, \dots, X_k]$.

The **subpath** of path $[X_1, X_2, \dots, X_k]$ from X_i to X_j is the path $[X_i, X_{i+1}, \dots, X_j]$ where $1 \leq i < j \leq k$.

A **directed cycle** is a path from a node to itself. A **simple path** is a path containing no subpaths, which are directed cycles.

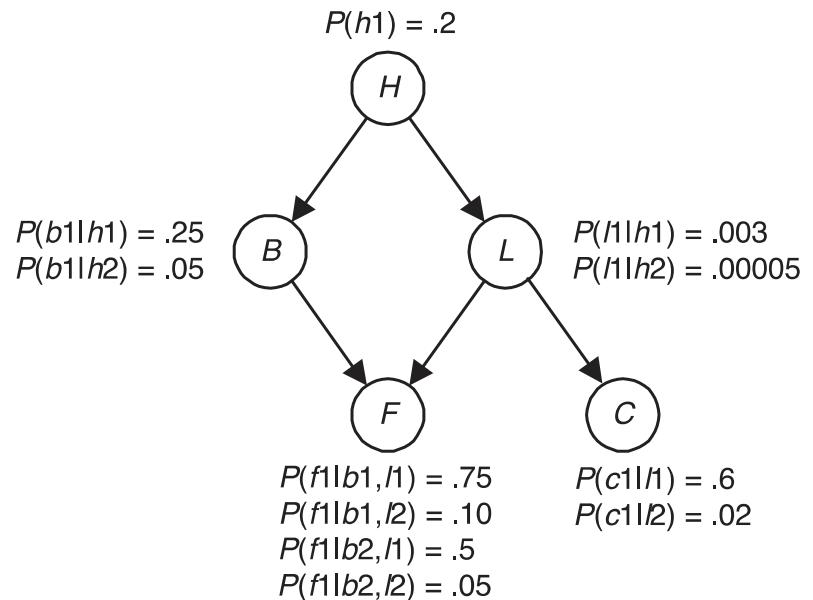
A directed graph G is called a **directed acyclic graph (DAG)** if it contains no directed cycles. Given a DAG $G = (V, E)$ and nodes X and Y in V :

- Y is called a **parent** of X if there is an edge from Y to X
- Y is called a **descendent** of X and X is called an **ancestor** of Y if there is a path from X to Y
- Y is called a **nondescendent** of X if Y is not a descendent of X .
- X is not considered a descendent of X because we require $k \geq 2$ in the definition of a path.

Determine Markov Conditions

Fill in the table:

- Who are the parents (PA)?



Node	PA	Conditional Independence
C		
B		
F		
L		

Definition 1.9

Definition 1.9: Suppose we have

- a joint probability distribution P of the random variables in some set V and
- a DAG $G=(V,E)$.

We say that (G,P) satisfies the **Markov condition** if for each variable $X \in V$, $\{X\}$ is conditionally independent of the set of all its non-descendents ND_X given the set of all its parents PA_X , i.e.,

$$I_P(\{X\}, ND_X | PA_X)$$

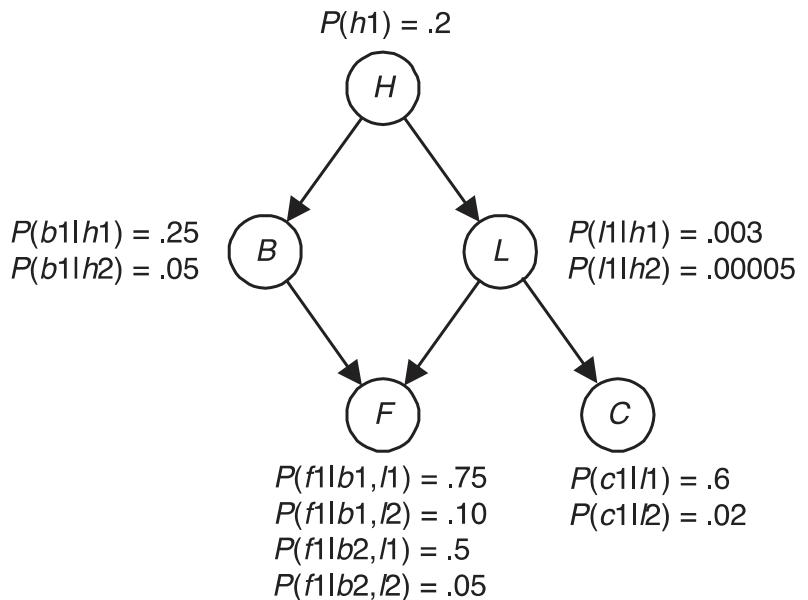
If X is a root, then $PA_X = \emptyset \Rightarrow I_P(\{X\}, ND_X)$.

Note:

- $I_P(\{X\}, ND_X | PA_X) \Rightarrow I_P(\{X\}, B | PA_X) \quad \forall B \subseteq ND_X$.
- $PA_X \subseteq ND_X$

Fill in the table:

- What conditional independencies does the Markov Condition implies?



Node	PA	Conditional Independence
C	{L}	
B	{H}	
F	{B,L}	
L	{H}	

Definition 1.9

Definition 1.9: Suppose we have

- a joint probability distribution P of the random variables in some set V and
- a DAG $G=(V,E)$.

We say that (G,P) satisfies the **Markov condition** if for each variable $X \in V$, $\{X\}$ is conditionally independent of the set of all its non-descendents ND_X given the set of all its parents PA_X , i.e.,

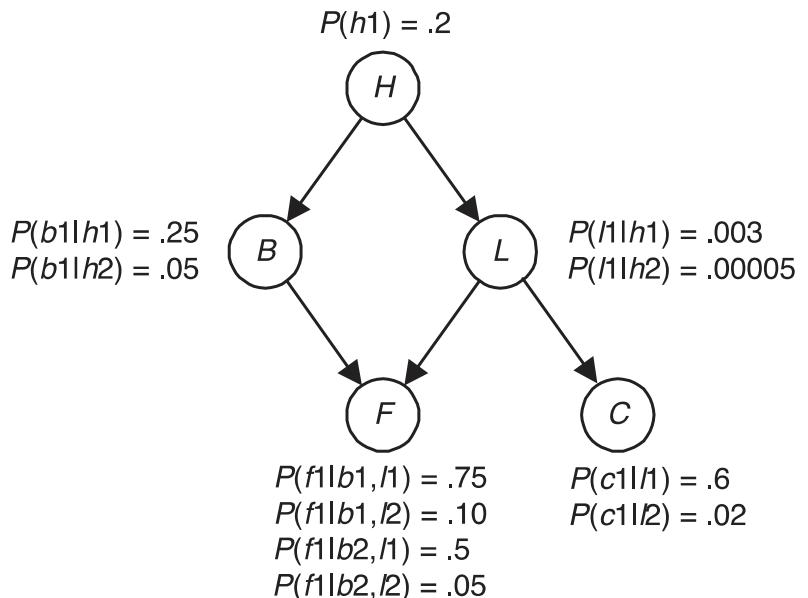
$$I_P(\{X\}, ND_X | PA_X)$$

If X is a root, then $PA_X = \emptyset \Rightarrow I_P(\{X\}, ND_X)$.

Note:

- $I_P(\{X\}, ND_X | PA_X) \Rightarrow I_P(\{X\}, B | PA_X) \quad \forall B \subseteq ND_X$.
- $PA_X \subseteq ND_X$

For the example: If (G,P) satisfy the Markov condition, then P must have conditional independencies of the table.



Node	PA	Conditional Independence
C	{L}	$I_P(\{C\}, \{H, B, F\} \{L\})$
B	{H}	$I_P(\{B\}, \{L, C\} \{H\})$
F	{B, L}	$I_P(\{F\}, \{H, C\} \{B, L\})$
L	{H}	$I_P(\{L\}, \{B\} \{H\})$

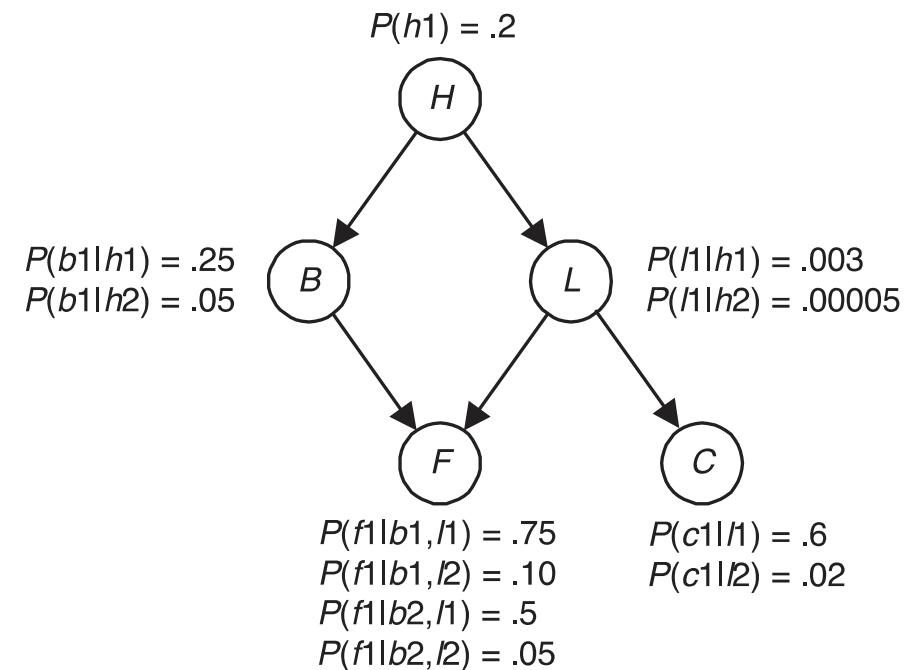
Theorem 1.4 (1)

Theorem 1.4: If (G, P) satisfies the Markov condition, then P is equal to the product of its conditional distributions of all nodes given values of their parents, whenever these conditional distributions exist.

Let X_1, X_2, \dots, X_n be the random variables in V . For a given set of values x_1, x_2, \dots, x_n let pa_i be the subset of these values containing the values of X_i 's parents. Then

$$P(x_n, x_{n-1}, \dots, x_1) = P(x_n|pa_n)P(x_{n-1}|pa_{n-1}) \dots P(x_1|pa_1)$$

whenever $P(pa_i) \neq 0$ for $1 \leq i \leq n$.



Example:

$$P(f, c, b, l, h) = P(f|b, l)P(c|l)P(b|h)P(l|h)P(h)$$

needs $2^5 - 1$ $= 31$ values	needs $4 + 2 + 2 + 2 + 1$ $= 11$ values
-------------------------------------	---

Theorem 1.4 (2)

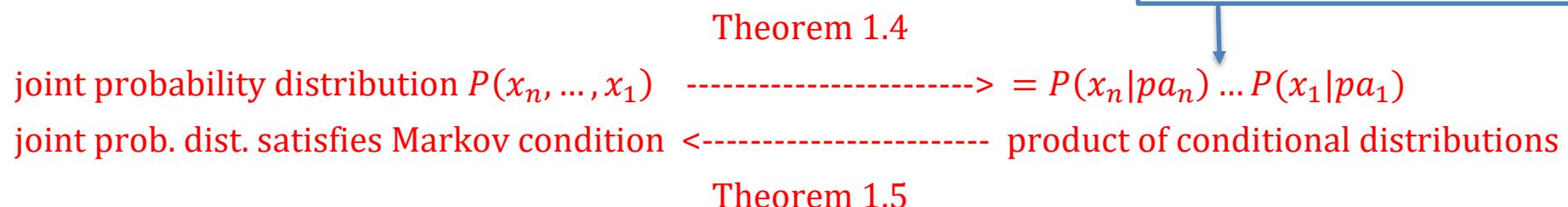
Theorem 1.4 often enables us to reduce the problem of determining a huge number of probability values to that of determining relatively few. The # of values in the joint distribution is exponential in terms of the # of variables. However, each of these values is uniquely determined by the conditional distributions, and if each node in the DAG does not have too many children, there are not many values in these distributions.

Example:

- n nodes, each node has two possible values and at most one parent
⇒ $2n$ probability values vs. $2^n - 1$
- n nodes, each node has two possible values and at most k parent
⇒ $2^k n$ probability values vs. $2^n - 1$

Theorem 1.5

Theorem 1.5: Let the DAG G be given in which each node is a random variable, and let a discrete conditional probability distribution of each node given values of its parents in G be specified. Then the product of these conditional distributions yields a joint probability distribution P of the variables, and (G, P) satisfies the Markov condition.

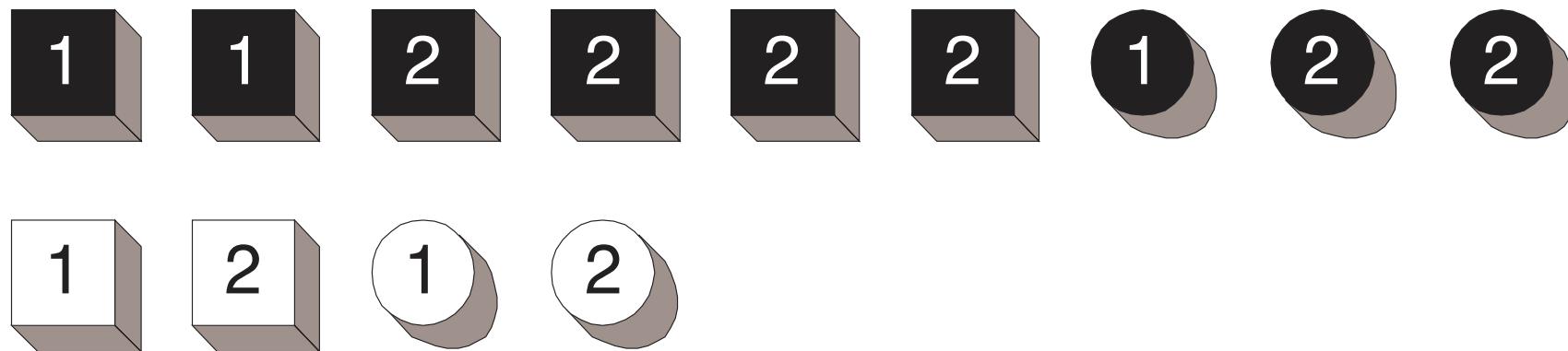


Bayesian Network := Let P be a joint probability distribution of the random variables in some set V , and $G=(V,E)$ be a DAG. We call (G, P) a Bayesian Network if (G, P) satisfies the Markov condition.

Theorem 1.4 --> P is the product of its conditional distributions in G . This is how P is always represented in a Bayesian network.

Theorem 1.5 --> Specify a DAG G and any discrete conditional distributions, we obtain a Bayesian network. This is how a Bayesian network is constructed in practice.

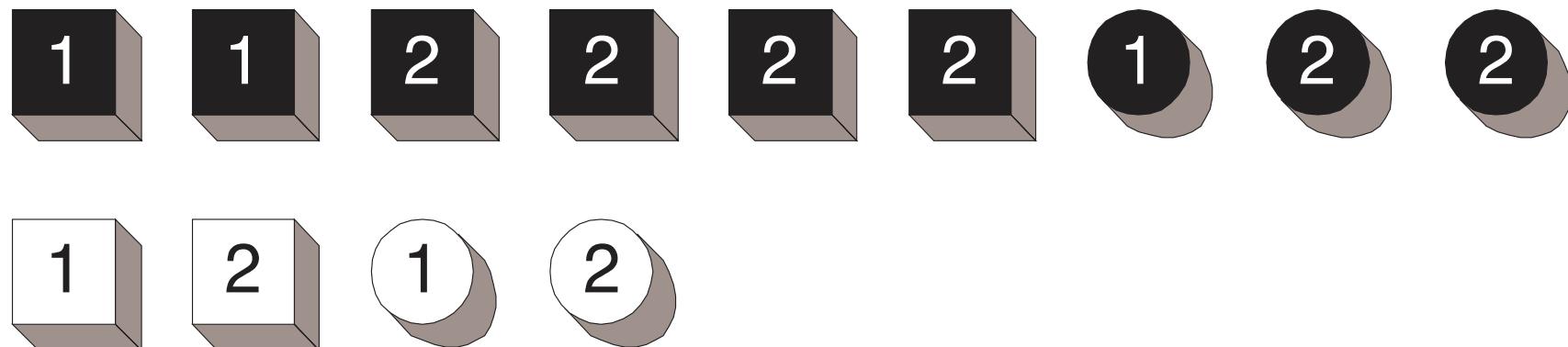
Example 1.35 (1)



C	V	S	P(c,s,v)
c1	v1	s1	/13 =
c2	v1	s1	/13 =
c1	v2	s1	/13 =
c2	v2	s1	/13 =
c1	v1	s2	/13 =
c2	v1	s2	/13 =
c1	v2	s2	/13 =
c2	v2	s2	/13 =

Var	Value	Outcomes Mapped to this Value
V	v1	All objects containing a "1"
	v2	All objects containing a "2"
S	s1	All square objects
	s2	All round objects
C	c1	All black objects
	c2	All white objects

Example 1.35 (1)



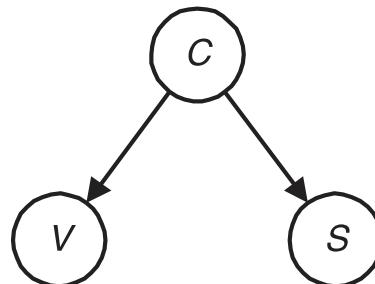
C	V	S	P(c,s,v)
c1	v1	s1	$2/13 = 0.1538 \dots$
c2	v1	s1	$1/13 = 0.0769 \dots$
c1	v2	s1	$4/13 = 0.3076 \dots$
c2	v2	s1	$1/13 = 0.3076 \dots$
c1	v1	s2	$1/13 = 0.3076 \dots$
c2	v1	s2	$1/13 = 0.3076 \dots$
c1	v2	s2	$2/13 = 0.1538 \dots$
c2	v2	s2	$1/13 = 0.3076 \dots$

Var	Value	Outcomes Mapped to this Value
V	v1	All objects containing a "1"
	v2	All objects containing a "2"
S	s1	All square objects
	s2	All round objects
C	c1	All black objects
	c2	All white objects

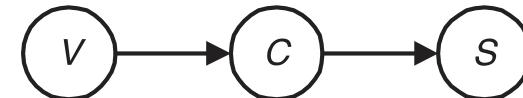
Example 1.35 (2)

From example 1.17 we know $I_P(\{V\}, \{S\} | \{C\})$.

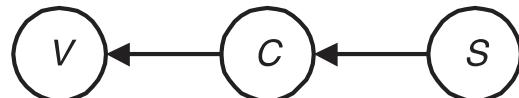
The graphs (a) – (c) satisfy this Markov condition. (d) does not satisfy this Markov condition.



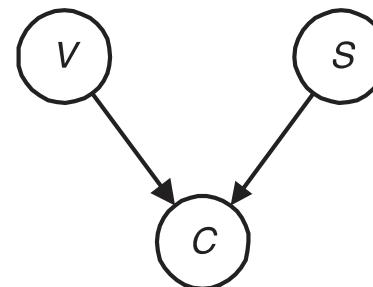
(a)



(b)



(c)



(d)

Python – Code: Example 1.35(3)

```
'''This is Example 1.35(3)'''
from bayesian.bbn import *
```

```
def f_C(C):
    '''C'''
    if C:
        return 9.0 / 13.0
    else:
        return 4.0 / 13.0
```

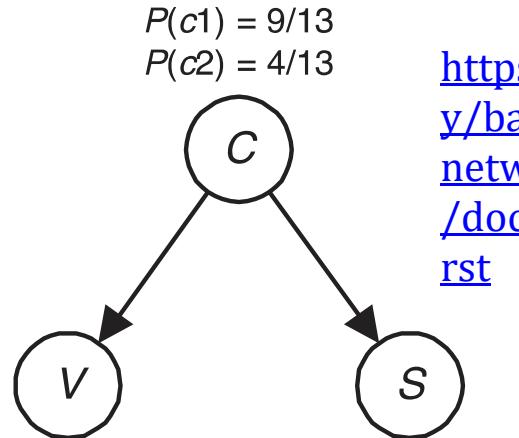
```
def f_V(V, C):
    '''V'''
    table = dict()
    table['tt'] = 1.0 / 3.0
    table['tf'] = 1.0 / 2.0
    table['ft'] = 1.0 - table['tt']
    table['ff'] = 1.0 - table['tf']
    key = ''
    key = key + 't' if V else key + 'f'
    key = key + 't' if C else key + 'f'
    return table[key]
```

```
def f_S(S, C):
    '''S'''
    table = dict()
    table['tt'] = 2.0 / 3.0
    table['tf'] = 1.0 / 2.0
    table['ft'] = 1.0 - table['tt']
    table['ff'] = 1.0 - table['tf']
    key = ''
    key = key + 't' if S else key + 'f'
```

```
key = key + 't' if C else key + 'f'
return table[key]
```

```
if __name__ == '__main__':
    g = build_bbn(
        f_C, f_V, f_S)
    g.q()
```

<https://github.com/eBay/bayesian-belief-networks>



$$\begin{aligned}P(c_1) &= 9/13 \\P(c_2) &= 4/13\end{aligned}$$

$$\begin{array}{ll}P(v_1|c_1) = 1/3 & P(s_1|c_1) = 2/3 \\P(v_2|c_1) = 2/3 & P(s_2|c_1) = 1/3\end{array}$$

$$\begin{array}{ll}P(v_1|c_2) = 1/2 & P(s_1|c_2) = 1/2 \\P(v_2|c_2) = 1/2 & P(s_2|c_2) = 1/2\end{array}$$

<https://github.com/eBay/bayesian-belief-networks/blob/master/docs/tutorial/tutorial.rst>

Python – Code : Example 1.35(4)

```

'''This is Example 1.35(4)'''
from bayesian.bbn import *

def f_V(V):
    '''V'''
    if V:
        return 5.0 / 13.0
    else:
        return 8.0 / 13.0

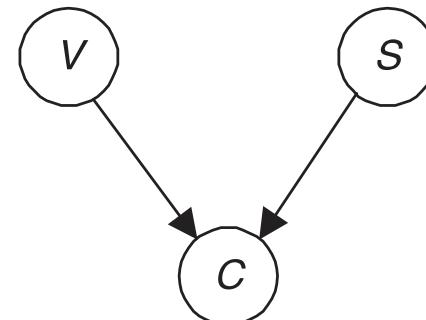
def f_S(S):
    '''S'''
    if S:
        return 8.0 / 13.0
    else:
        return 5.0 / 13.0

def f_C(C, V, S):
    '''C'''
    table = dict()
    table['ttt'] = 2.0 / 3.0
    table['ttf'] = 1.0 / 2.0
    table['tft'] = 4.0 / 5.0
    table['tff'] = 2.0 / 3.0
    table['ftt'] = 1.0 - table['ttt']
    table['ftf'] = 1.0 - table['ttf']
    table['fft'] = 1.0 - table['tft']
    table['fff'] = 1.0 - table['tff']
    key = ''
    key = key + 't' if C else key + 'f'
    key = key + 't' if V else key + 'f'
    key = key + 't' if S else key + 'f'
    return table[key]

if __name__ == '__main__':
    g = build_bbn(
        fC, fV, fS)
    g.q()
    #g.q(P='high')
    #g.q(D=True)
    #g.q(S=True)
    #g.q(C=True, S=True)
    #g.q(D=True, S=True)

P(v1) = 5/13          P(s1) = 8/13

```



$$\begin{aligned}
 P(c_1|v_1, s_1) &= 2/3 \\
 P(c_1|v_1, s_2) &= 1/2 \\
 P(c_1|v_2, s_1) &= 4/5 \\
 P(c_1|v_2, s_2) &= 2/3
 \end{aligned}$$

<https://github.com/eBay/bayesian-belief-networks>

<https://github.com/eBay/bayesian-belief-networks/blob/master/docs/tutorial/tutorial.rst>

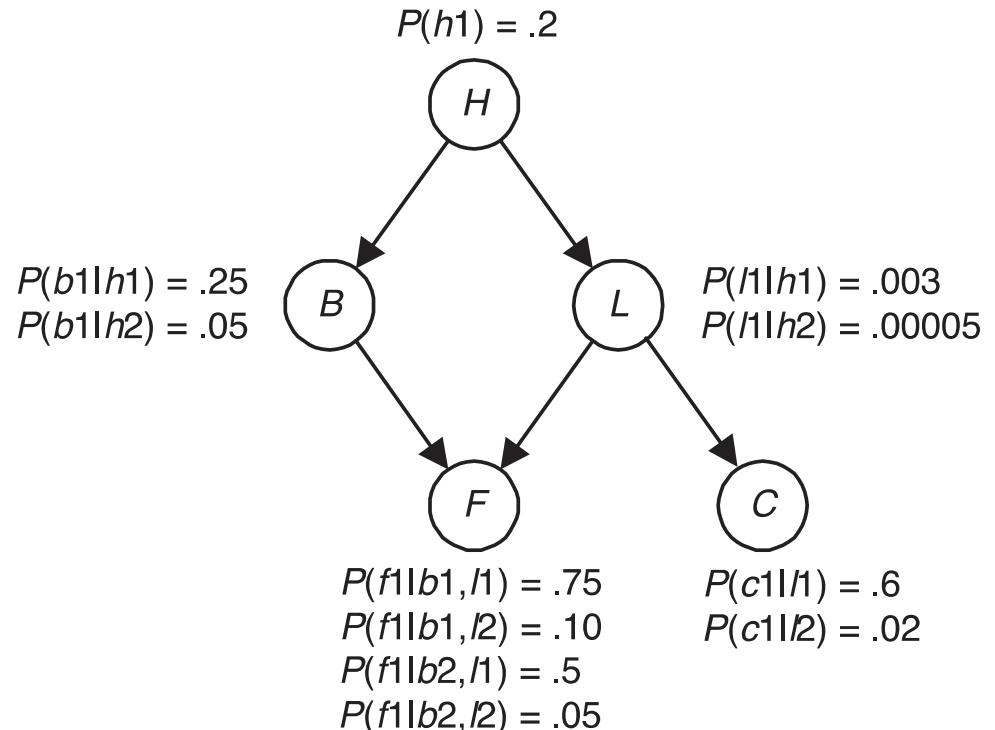
Warning

If we develop a Bayesian network from an arbitrary DAG and the conditionals of a probability distribution P relative to that DAG, **in general the resultant Bayesian network does not contain P** (because it violates the Markov condition).

We will argue that **if we construct a DAG using causal edges, we often have a DAG that satisfies the Markov condition with the relative frequency distribution of the variables.**

For our example right, we believe that we have modeled causal edges and that thus the joint probability does indeed satisfy the Markov condition.

Note the conditional probabilities are only relative frequency estimates.



Creating BNs Using Causal Edges (1)

Given a set of random variables V , if for every $X, Y \in V$ we draw an edge from X to Y iff X is a **direct cause** of Y relative to V , we call the resultant DAG a **causal DAG**.

What is causality (vs. correlation)?

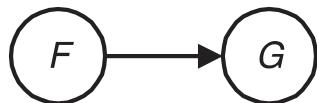
- ⇒ Can be defined based on manipulation
- ⇒ We say we **manipulate** X when we force X to take some value
- ⇒ We say **X causes Y** if there is some manipulation of X that leads to a change in the probability distribution of Y .
- ⇒ We say X being a **direct cause** of Y relative to V , if a manipulation of X changes the probability distribution of Y and there is no subset $W \subseteq V - \{X, Y\}$ such that if we instantiate the variables in W a manipulation of X no longer changes the probability of Y .
- ⇒ Direct causation depends on the set of variable in V , i.e., on the levels of detail of modeling.

Note: **Causation results in correlation; however, variables can be correlated without one causing the other** (e.g., often with values measured at the same time such DAX and hairline of Xyz at a given time).

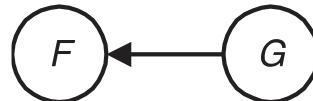
Correlation vs. Causation – Example (1)

Example:

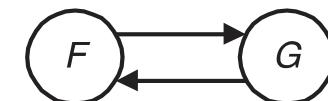
- F = Some Hair growth medicine
- G = Scalp hair growth



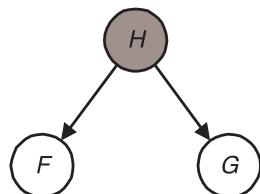
(a) F causes hair regrowth



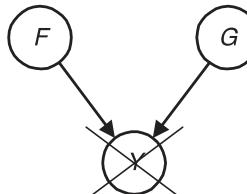
(b) Some other medication works, wherefore you try F, too.



(c) Causal/feedback loop



(d) Hidden cause H, e.g., concern H about hairline caused taking other hair growth medicine, too.



(e) Hidden effect Y, e.g., some peculiarity in the sample group under analysis, e.g. hypertension

(f) Total independence

Creating BNs Using Causal Edges (2)

A node is **instantiated** when we know its value for the entity currently being modeled.

We say X and Y have a **common cause**, if there is some variable that has causal paths into both X and Y. If X and Y have a common cause C, there is often a dependency between them through this common cause. (case (d))

Ordinarily, the instantiation of a **common effect** creates a dependency between its causes because each cause explains away the occurrence of the effect, thereby making the other cause less likely (case (e)):

- ⇒ Results in correlation between causes
- ⇒ Psychologists call it **discounting**.
- ⇒ Also called **selection bias**.

(G,P) satisfies the **faithfulness** condition if (G,P) satisfies the Markov condition and the only conditional independencies in P are those entailed by the Markov condition.

Causal Markov Assumption

If we create a causal DAG $G=(V,E)$ and assume the probability distribution of the variables in V satisfies the Markov condition with G , we say we are making the ***causal Markov assumption***.

The causal Markov assumption is ordinarily warranted if the following three conditions are satisfied:

1. There must be no hidden (= not modeled by a random variable, ≠ not observable) common causes
2. Selection bias must not be present (i.e., every observable or unobservable common effect is modeled), and
3. There must be no causal feedback loops.

Using causal edges is just **one** way to develop a DAG and a probability distribution that satisfy the Markov condition.

Non-causal example: (see slide Example 1.17 and 1.35 (2) on page 19 and page 41):

- We would not say that the **color** of the object has a causal influence on its **shape**.
- **The Markov condition is simply a property of the probabilistic relationships among the variables.**

SS 2014 – Bayesian Networks

More DAG/Probability Relationships

*University of Augsburg
Multimedia Computing and Computer Vision
Prof. Dr. Rainer Lienhart
Rainer.Lienhart@informatik.uni-augsburg.de
www.multimedia-computing.org*

Reference

Richard E. Neapolitan. **Learning Bayesian Networks.** *Prentice Hall Series in Artificial Intelligence*, ISBN 0-13-012534-2.

Don't forget. Reading the book chapters 1 – 6 is mandatory.

Chapter on ***More DAG/Probability Relationships***
(chapter 2)

Figures and text are taken from that book

Recap

Markov condition:

⇒ $I_P(\{X\}), ND_X | PA_X$ for $X \in V$

Attention: $\not\Rightarrow P(x_i | x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n) = P(x_i | pa_i)$

⇒ no edge --> no direct dependency

but an edge doesn't mean direct dependency

Faithfulness condition:

⇒ edge --> direct dependency

⇒ In future we require both conditions to be true for a Bayesian Network (G, P) .

Note: For some probabilities P , it is not possible to find a DAG with which P satisfy the Markov and faithfulness condition.

Entailed Conditional Independencies

If (G, P) satisfies the Markov condition, then each node in G is conditionally independent of the set of all its nondescendents given its parents.

If (G, P) satisfies the Markov condition there might be **additional conditional independencies** which P must satisfy other than the one based on a node's parents.

Definition 2.1: Let $G = (V, E)$ be a DAG, where V is a set of random variables. We say that, based on the Markov condition, G **entails** conditional independency $I_P(A, B | C)$ for $A, B, C \subseteq V$ if

$I_P(A, B | C)$ holds for every $P \in \mathbf{P}$

where \mathbf{P} is the set of all probability distributions P such that (G, P) satisfies the Markov condition. We also say the Markov condition entails the conditional independency for G .

We start with an example:

Example of Entailed Cond. Independencies (1)

We are modelling how professors obtain citations:

G: Graduate Program Quality

F: First Job Quality

B: # of Publications

C: # of Citations

Assume

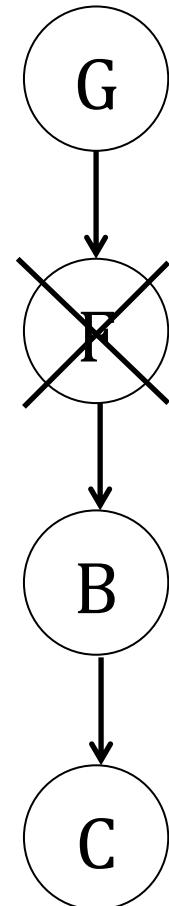
- DAG represents the causal relationships
- No hidden common causes
- No selection bias (= no hidden common effect)
- Some distribution P satisfies the Markov condition with the DAG (left),

Then

$$I_P = (\{C\}, \{F, G\} \mid \{B\})$$
$$I_P = (\{B\}, \{G\} \mid \{F\})$$

Additional (entailed) independency:

$$I_P = (\{C\}, \{G\} \mid \{F\})$$



Example of Entailed Cond. Independencies (2)

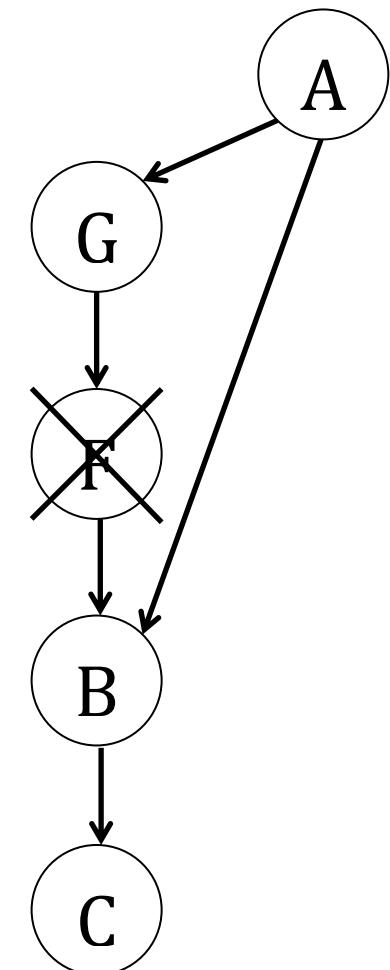
because

$$\begin{aligned} P(c|g, f) &= \sum_b P(c, b|f, g) \\ &= \sum_b P(c|b, f, g)P(b|f, g) \\ &= \sum_b P(c|b, f)P(b|f) \\ &= \sum_b P(c, b|f) \\ &= P(c|f) \end{aligned}$$

Example Entailed Cond. Independencies (3)

- A: Ability
- G: Graduate Program Quality
- F: First Job Quality
- B: # of Publications
- C: # of Citations

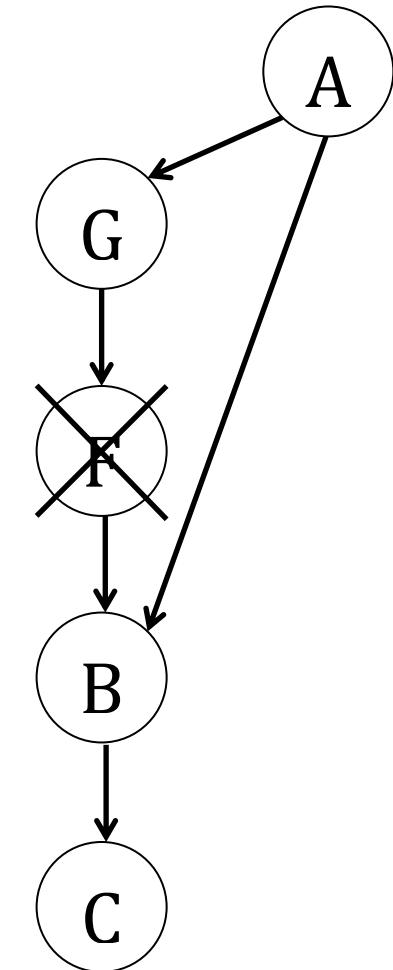
$$I_P = (\{C\}, \{G\} \mid \{F\}) \text{ ?}$$



Example Entailed Cond. Independencies (3b)

- A: Ability
- G: Graduate Program Quality
- F: First Job Quality
- B: # of Publications
- C: # of Citations

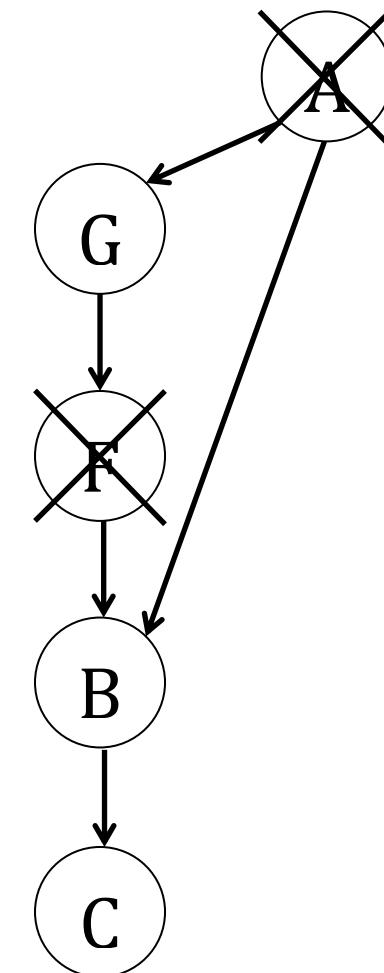
$$I_P = ((C), (G) \mid \{F\}) \text{ ?}$$



Example Entailed Cond. Independencies (4)

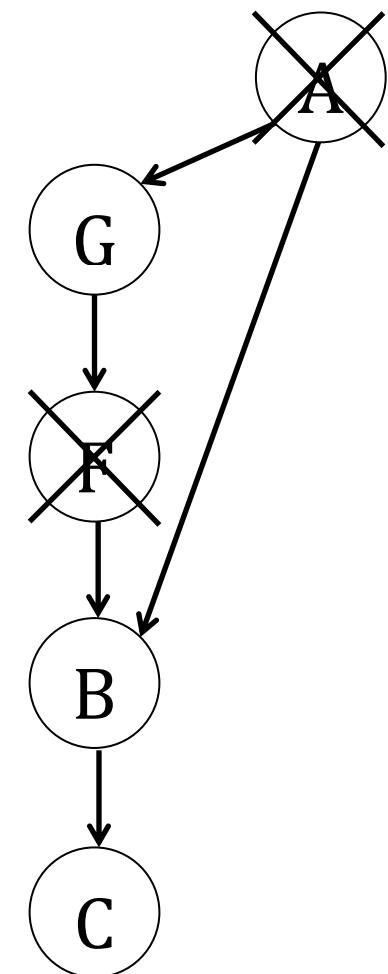
- A: Ability
- G: Graduate Program Quality
- F: First Job Quality
- B: # of Publications
- C: # of Citations

$$I_P = (\{C\}, \{G\} \mid \{A, F\}) \text{ ?}$$



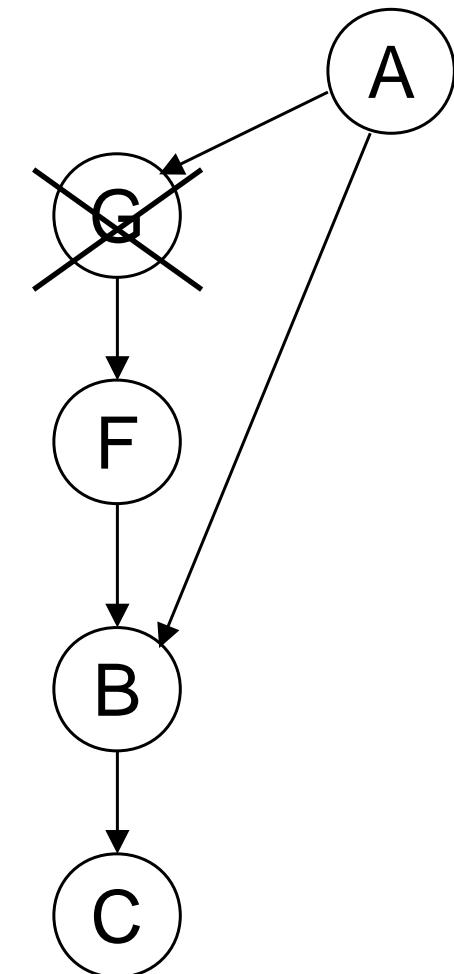
Example Entailed Cond. Independencies (4b)

$$I_P = (\{C\}, \{G\} \mid \{A, F\}) \quad \checkmark$$



Example Entailed Cond. Independencies (5)

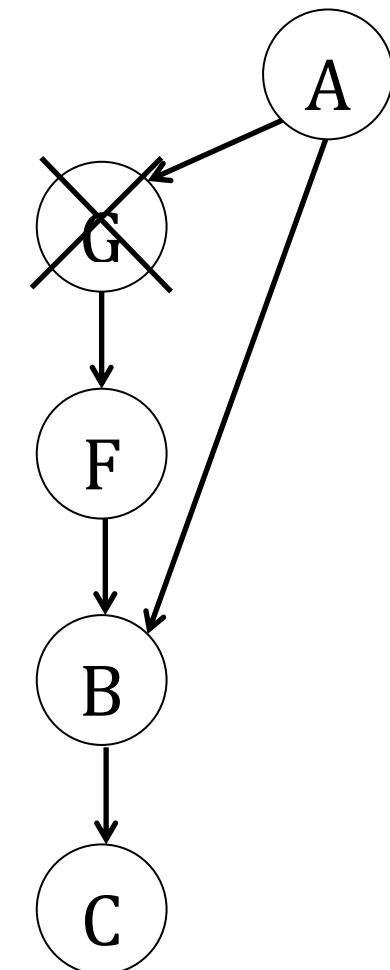
$$I_P = (\{F\}, \{A\} \mid \{G\}) \text{ ?}$$



Example Entailed Cond. Independencies (5b)

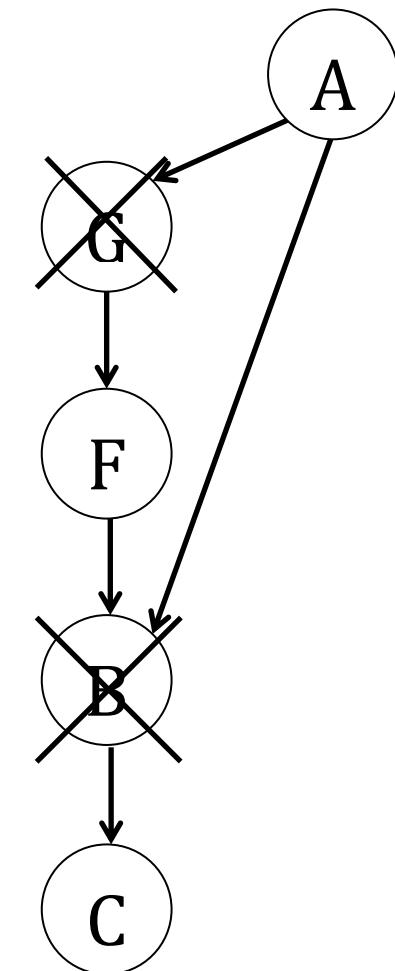
$$I_P = (\{F\}, \{A\} \mid \{G\}) \quad \checkmark$$

⇒ Direct result of Markov condition



Example Entailed Cond. Independencies (6)

$$I_P = (\{F\}, \{A\} \mid \{B, G\}) \text{ ?}$$

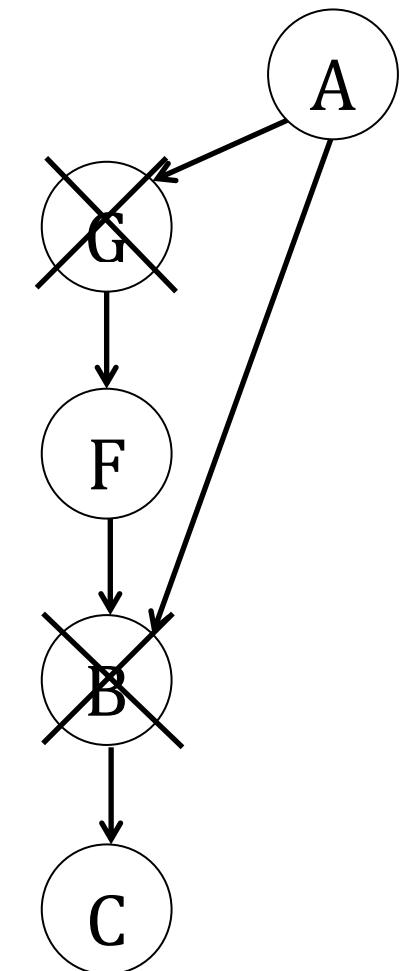


Example Entailed Cond. Independencies (6b)

$$I_P = (\{F\}, \{A\} \mid \{B, G\}) ?$$

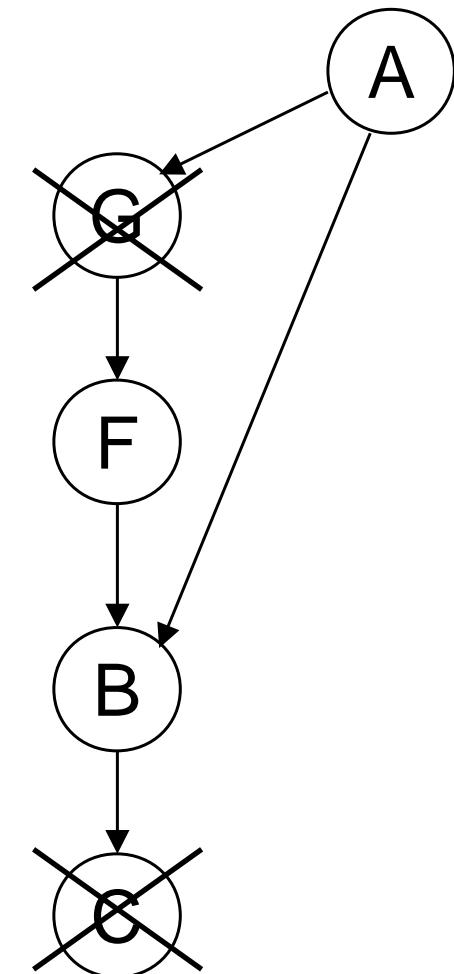
⇒ discounting

⇒ Uncoupled head-to-head meeting at B



Example Entailed Cond. Independencies (7)

$$I_P = (\{F\}, \{A\} \mid \{C, G\}) ?$$

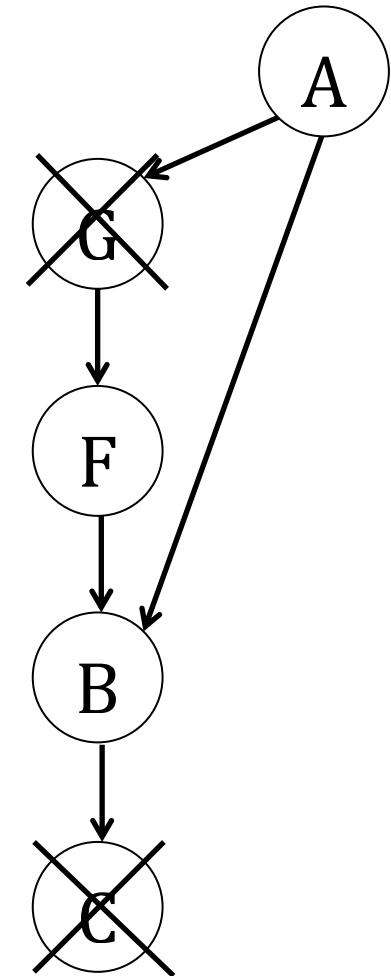


Example Entailed Cond. Independencies (7b)

$$I_P = (\{F\}, \{A\} \mid \{C, G\}) ?$$

⇒ discounting

⇒ Uncoupled head-to-head meeting at B



d-Separation (1)

A concept called d-Separation is introduced.

What we will show:

The Markov condition entails that all d-separations are conditional independencies.

Every conditional independency entailed by the Markov condition is identified by d-separation

⇒ d-Separation is used to identify all conditional independencies, which are common to all probability distributions $P \in \mathbf{P}$ satisfying the Markov condition with G.

Note: A specific P in \mathbf{P} might show even more conditional independencies.

d-Separation (2)

Suppose we have a DAG $G = (V, E)$, and a set of nodes $\{X_1, X_2, \dots, X_k\}$, where $k \geq 2$, such that $(X_{i-1}, X_i) \in E$ or $(X_i, X_{i-1}) \in E$ for $2 \leq i \leq k$. We call the set of edges connecting the k nodes a **chain** between X_1 and X_k . We denote the chain using both the sequence $[X_1, X_2, \dots, X_k]$ and the sequence $[X_k, X_{k-1}, \dots, X_1]$.

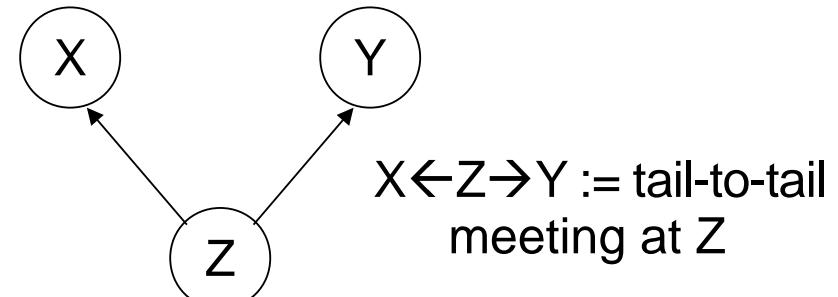
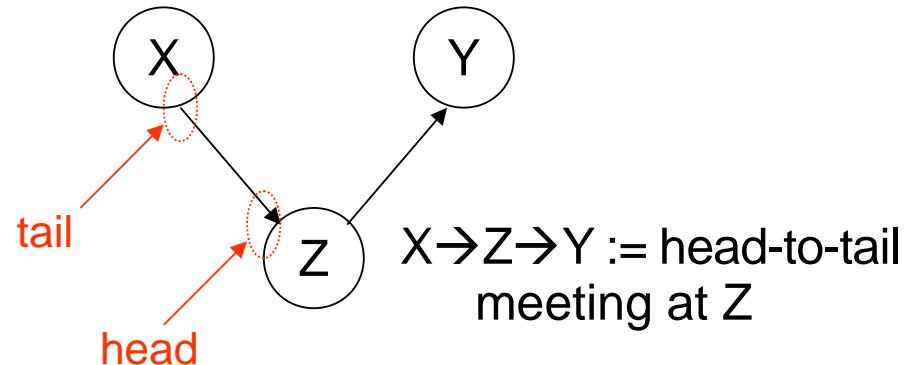
The nodes X_2, \dots, X_{k-1} are called **interior nodes** on chain $[X_1, X_2, \dots, X_k]$. The subchain of chain $[X_1, X_2, \dots, X_k]$ between X_i and X_j is the chain $[X_i, X_{i+1}, \dots, X_j]$ where $1 \leq i < j \leq k$.

A **cycle** is a chain between a node and itself. A **simple chain** is a chain containing no subchains, which are cycles.

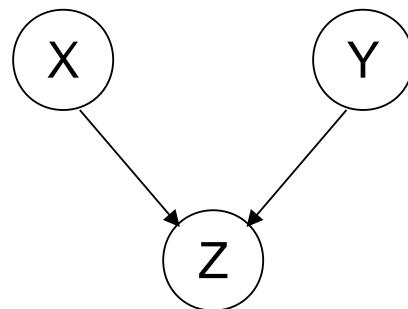
We often denote chains by showing undirected lines between the nodes in the chain. For example, we would denote the chain $[G, A, B, C]$ as $G - A - B - C$. If we want to show the direction of the edges, we use arrows. For example, to show the direction of the edges, we denote the previous chain as $G \leftarrow A \rightarrow B \rightarrow C$.

A chain containing two nodes, such as $X - Y$, is called a **link**. A directed link, such as $X \rightarrow Y$, represents an edge, and we will call it an edge. Given the edge $X \rightarrow Y$, we say the **tail of the edge** is at X and the **head of the edge** is Y .

d-Separation (3)



$X \rightarrow Z \leftarrow Y :=$ head-to-head
meeting at Z



X and Y are
not adjacent!

$X - Z - Y :=$
uncoupled
meeting at Z

Chain-Blocking

Definition 2.2: Let $G = (V, E)$ be a DAG, $A \subseteq V$, X and Y be distinct nodes in $V - A$, and be ρ a chain between X and Y. Then **chain** ρ is **blocked** by A if one of the following holds:

1. There is a node $Z \in A$ on the chain ρ , and the edges incident to Z on ρ meet head-to-tail at Z.
2. There is a node $Z \in A$ on the chain ρ , and the edges incident to Z on ρ meet tail-to-tail at Z.
3. There is a node Z, such that Z and all of Z's descendants are not in A, on the chain ρ , and the edges incident to Z on ρ meet head-to-head at Z.

A **chain** is called **active** given A if it is not blocked by A.

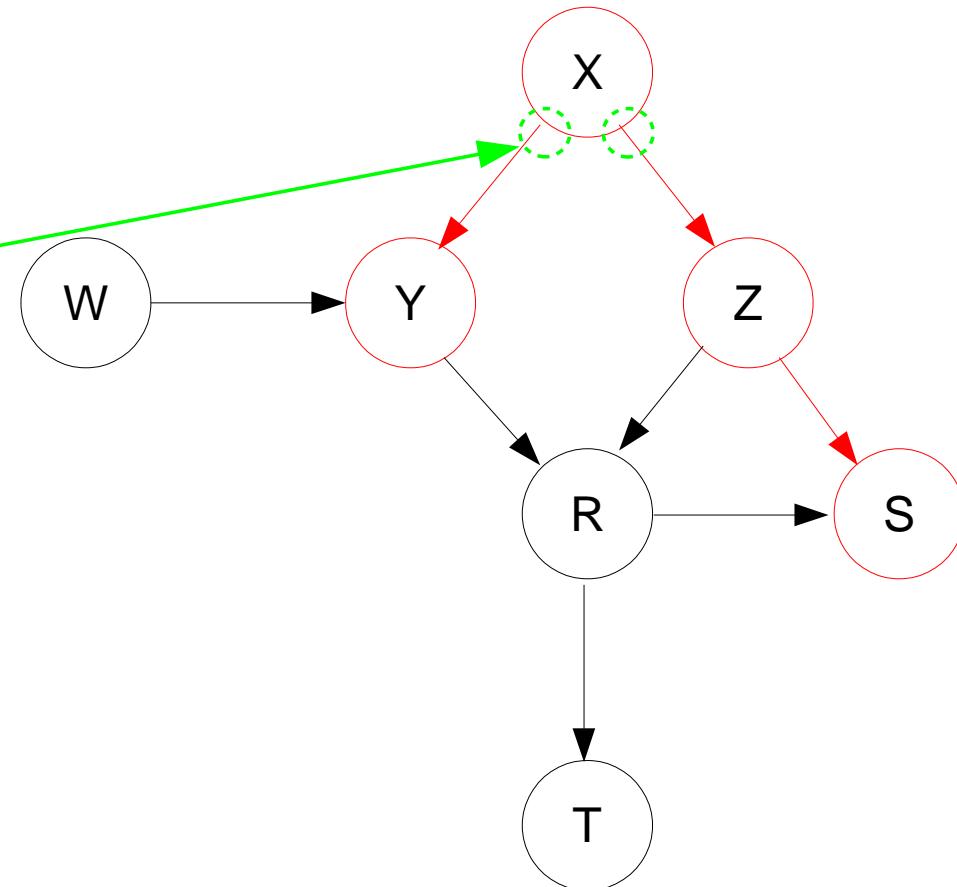
If there is an active chain ρ between node X and some other node, then every 3-node subchain U-V-W of ρ has one of the following properties:

1. U-V-W is not head-to-head at V and V is not in A.
2. U-V-W is head-to-head at V and V is or has a descendent in A.

Example 2.1 (1)

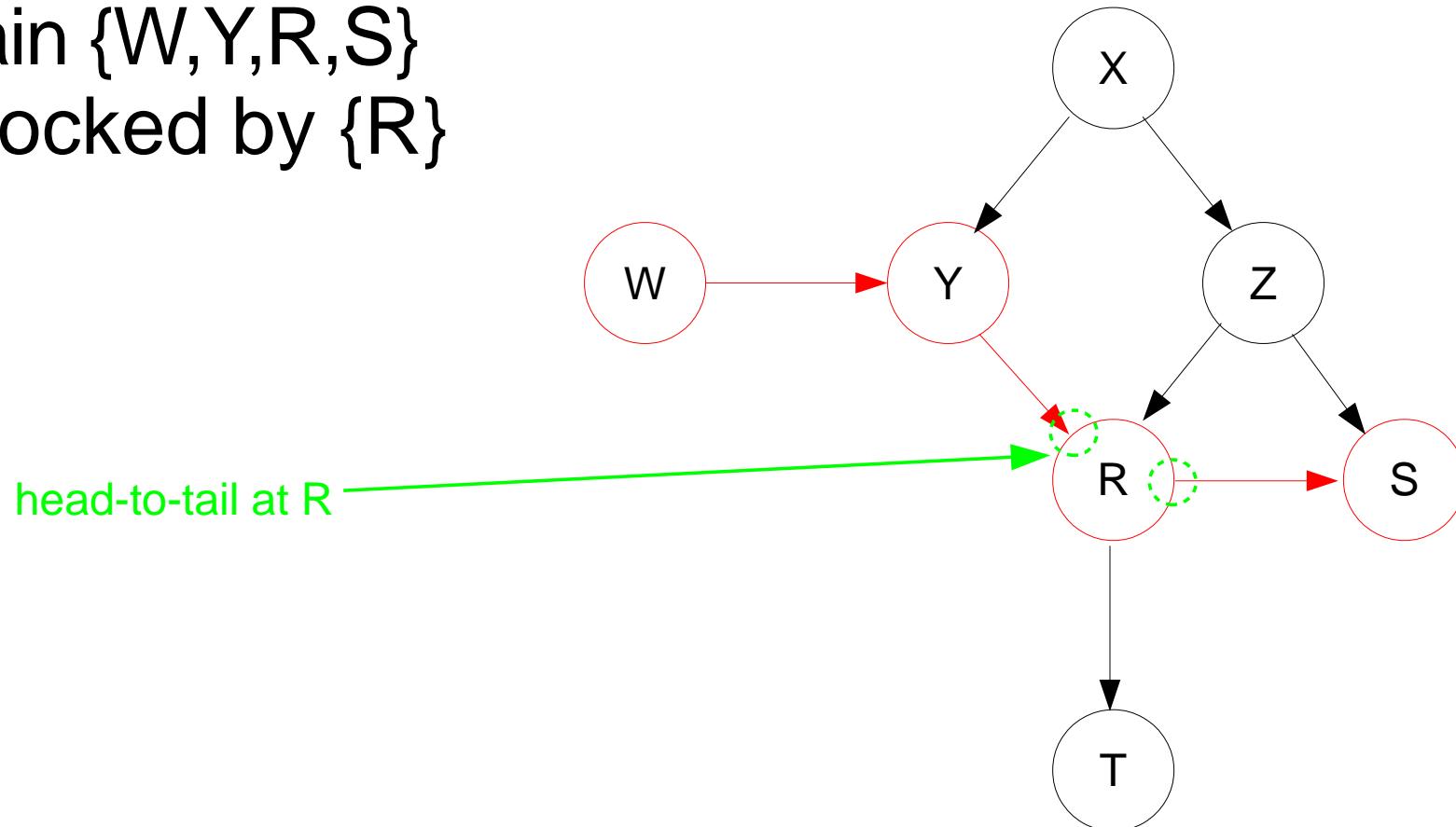
Chain {Y,X,Z,S}
blocked by {X}

Tail-to-tail at X



Example 2.1 (2)

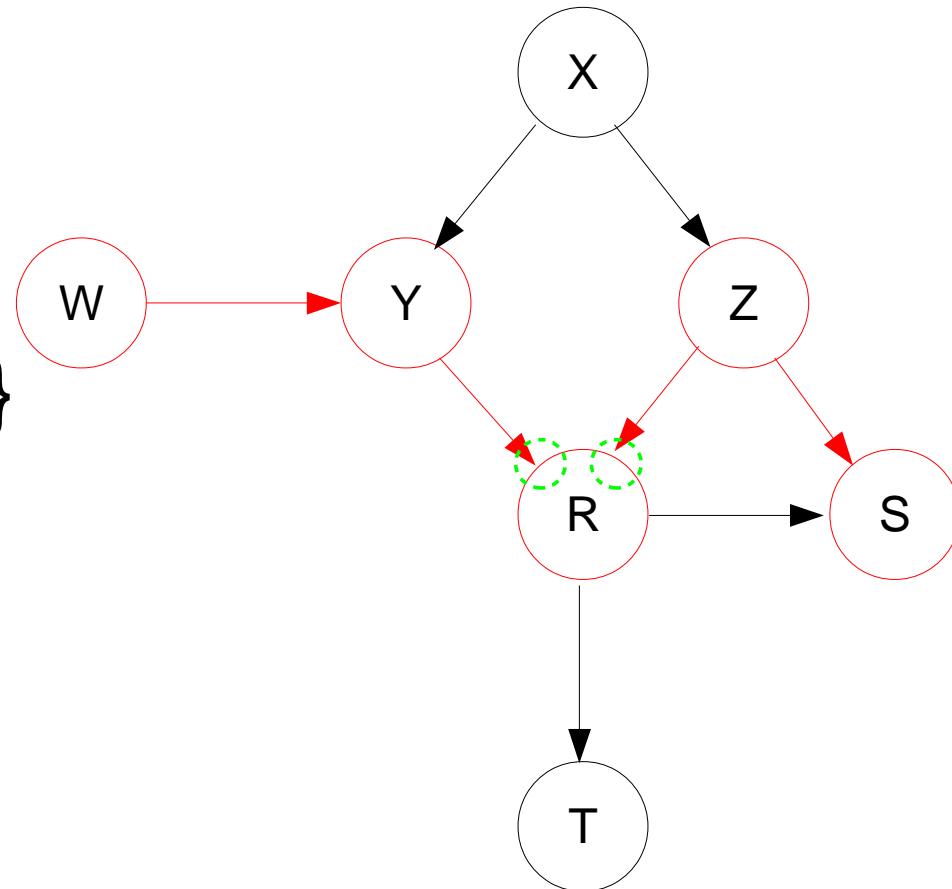
Chain {W,Y,R,S}
blocked by {R}



Example 2.1 (3)

Chain {W,Y,R,Z,S}

- not blocked by {R}
- blocked by {Ø}
- not blocked by {T}



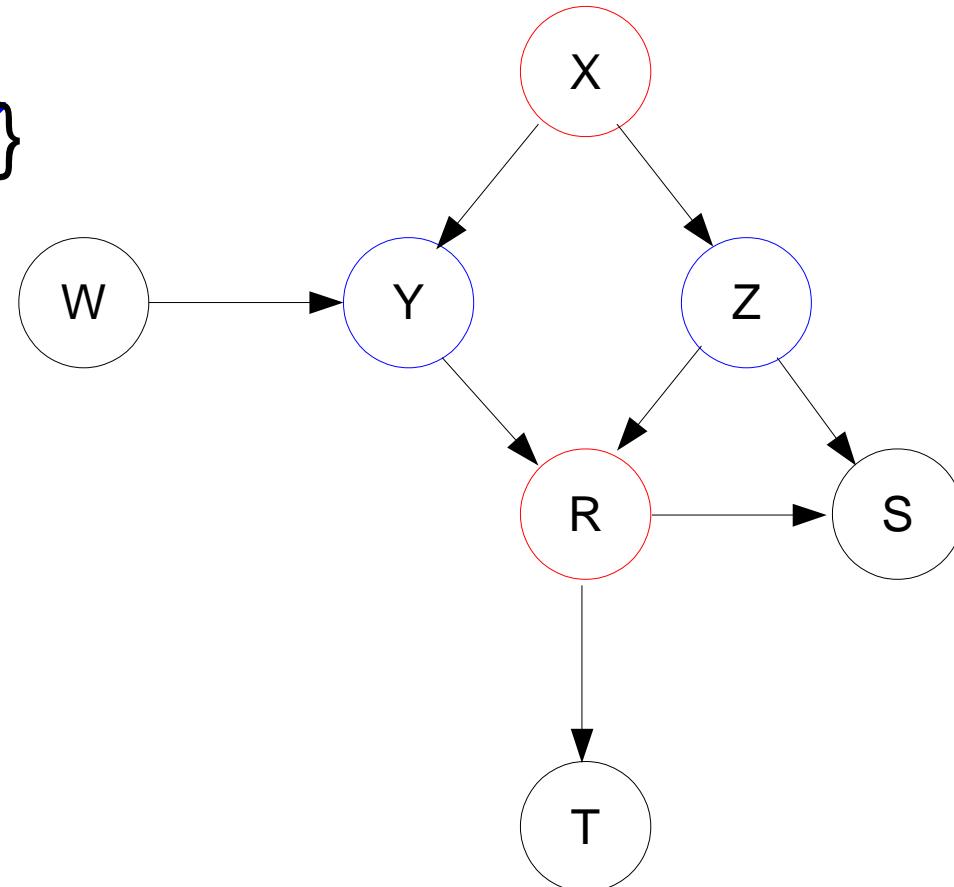
Definition of d-Separation of Nodes

Definition 2.3: Let $G = (V, E)$ be a DAG, $A \subseteq V$, and X and Y be distinct nodes in $V-A$. We say X and Y are **d-separated** by A in G if every chain between X and Y is blocked by A .

Note: Every chain between X and Y is blocked by A iff every simple chain between X and Y is blocked by A .

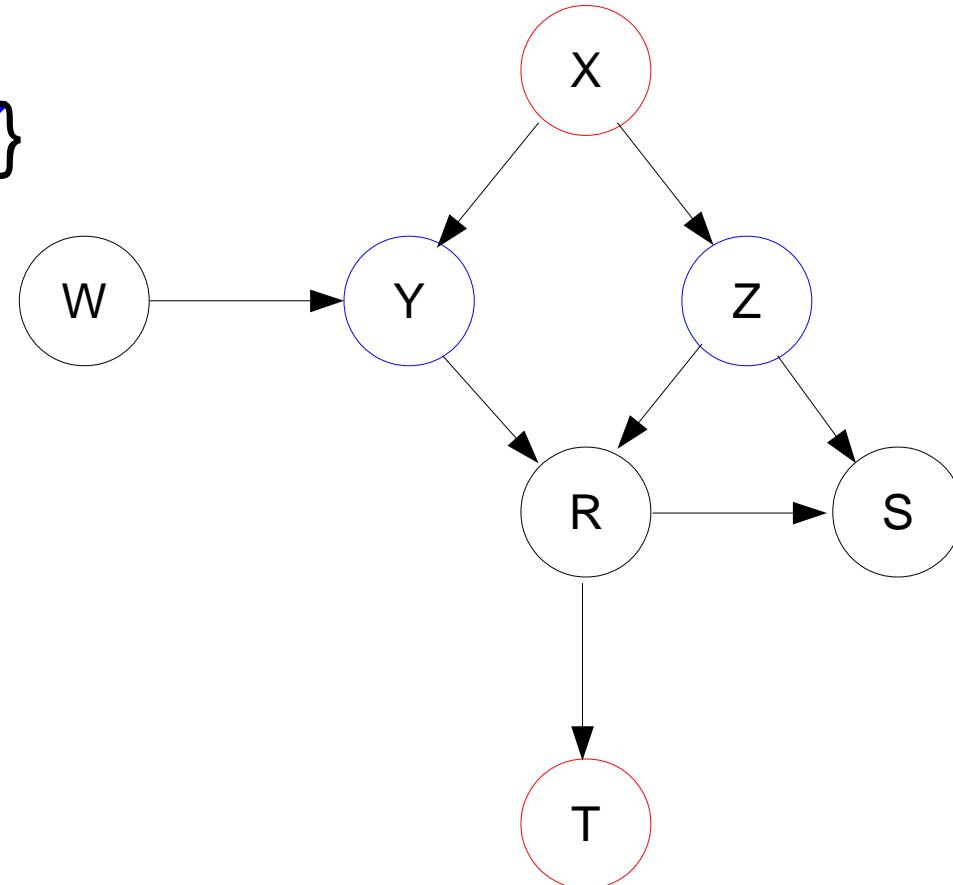
Example 2.2 (1)

X and R are d-separated by $\{Y, Z\}$



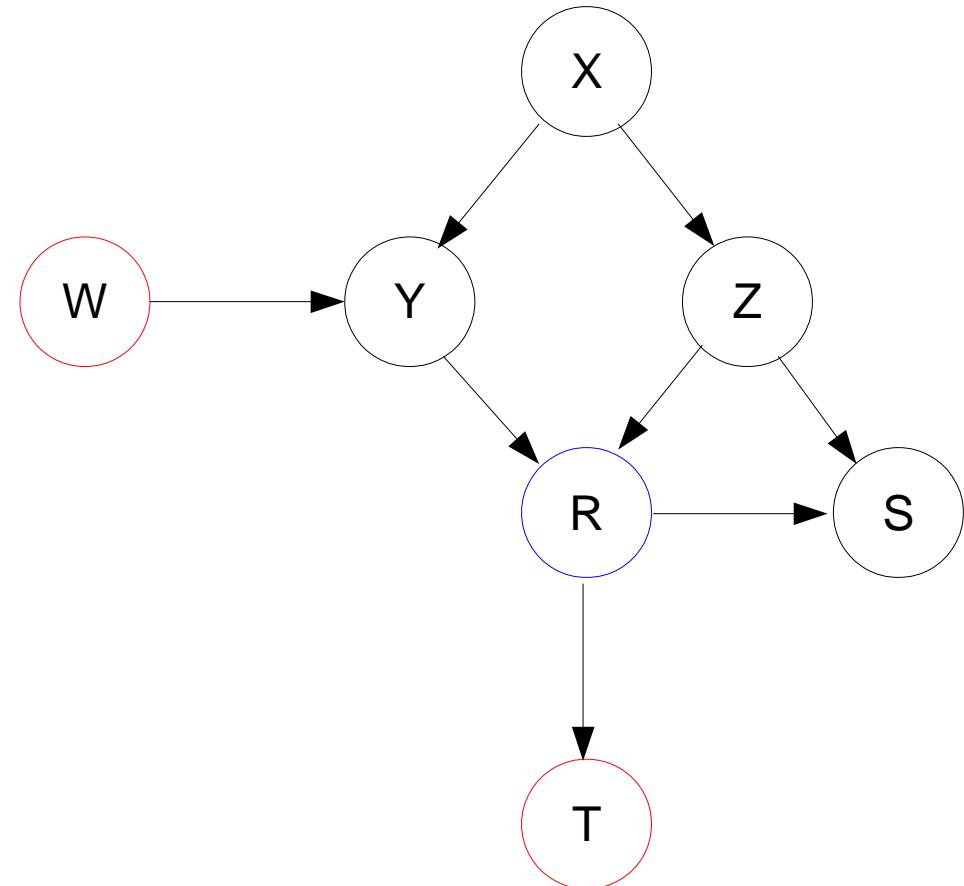
Example 2.2 (2)

X and T are d-separated by {Y,Z}



Example 2.2 (3)

W and T are d-separated by $\{R\}$



Definition of d-Separation of Sets of Nodes

Definition 2.4: Let $G = (V, E)$ be a DAG, and A, B, and C be mutually disjoint subsets of V. We say A and B are **d-separated** by C in G if for every $X \in A$ and $Y \in B$, X and Y are d-separated by C. We write

$$I_G(A, B | C)$$

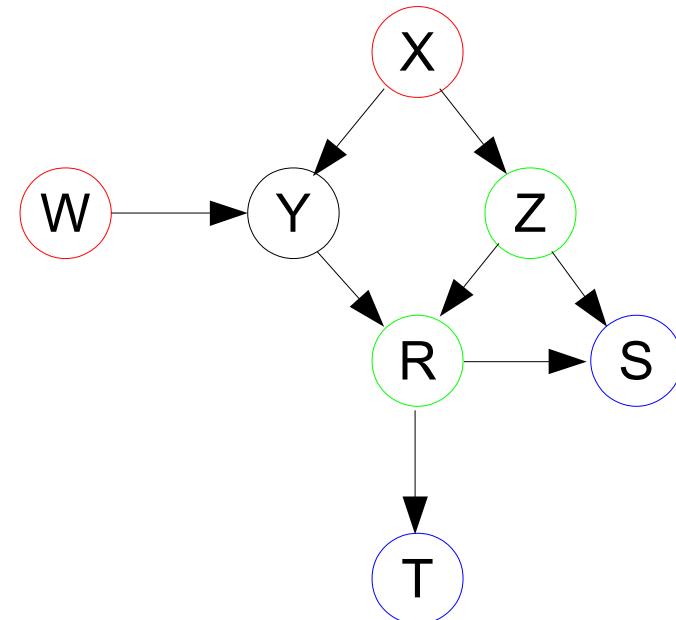
If $C = \emptyset$, we write only

$$I_G(A, B)$$

Example:

$$I_G(\{W, X\}, \{S, T\} | \{R, Z\})$$

because every chain between W and S, W and T, X and S, and X and T is blocked by $\{R, Z\}$.



Lemma 2.1: All d-Separations are Conditional Independencies

Lemma 2.1: Let P be a probability distribution of the variables in V and $G = (V, E)$ be a DAG. Then (G, P) satisfies the Markov condition iff, for every three mutually disjoint subsets $A, B, C \subseteq V$, whenever A and B are d-separated by C , A and B are conditionally independent in P given C .

That is, (G, P) satisfies the Markov condition iff

$$I_G(A, B, |C) \Rightarrow I_P(A, B|C)$$

Therefore, if (G, P) satisfies the Markov condition, we say G is an **independence map** of P .

Definition 2.5: Let V be a set of random variables, and A_1, B_1, C_1, A_2, B_2 and C_2 be subsets of V . We say conditional independency $I_P(A_1, B_1 | C_1)$ is **equivalent** to conditional independency $I_P(A_2, B_2 | C_2)$ if for every probability distribution P of V , $I_P(A_1, B_1 | C_1)$ holds iff $I_P(A_2, B_2 | C_2)$ holds.

Theorem 2.1: Every Entailed Conditional Independency is Identified by d-Separation

Lemma 2.3: Let $G = (V, E)$ be a DAG, and \mathbf{P} be the set of all probability distributions P such that (G, P) satisfies the Markov condition. Then for every three mutually disjoint subsets $A, B, C \subseteq V$,

$$I_P(A, B | C) \forall P \in \mathbf{P} \Rightarrow I_G(A, B | C)$$

We say conditional independency $I_P(A, B | C)$ is **identified by d-separation** in G if one of the following holds:

1. $I_G(A, B | C)$.
2. A, B, and C are not mutually disjoint, A', B', C' are mutually disjoint, $I_P(A, B | C)$ and $I_P(A', B' | C')$ are equivalent, and we have $I_G(A', B' | C')$.

Theorem 2.1: Based on the Markov condition, a DAG G entails all and only those conditional independencies that are identified by d-separation in G.

Note: A particular distribution P , that satisfies the Markov condition with G , may have conditional independencies that are not identified by d-separation. --> See Example on slide 'Theorem 2.5'

Finding d-Separations (1)

See the book (Section 2.1.3. pp. 80-86) for an algorithm that finds in a DAG (G, V) the subset $D \subseteq V$ that contains all nodes which are d-separated from every node in $B \subseteq V$ by $A \subseteq V$. That is

Given $B \subseteq V$ and $A \subseteq V$, the algorithm finds the maximal set $D \subseteq V$, such that $I_G(B, D|A)$.

Why is this useful?

- ⇒ Assume you want to determine $P(B|A)$ with $B, A \subseteq V$.
- ⇒ Which variables $D \subseteq V$ are irrelevant?
- ⇒ *In other words:* If we want to improve the precision of our estimation of $P(B|A)$, we need to improve the estimates of $V - \{B, D\}$. Further, only these variables have to be part of a (parameter) sensitivity analysis.

Markov Equivalence (1)

Different DAGs can have the same d-separations.

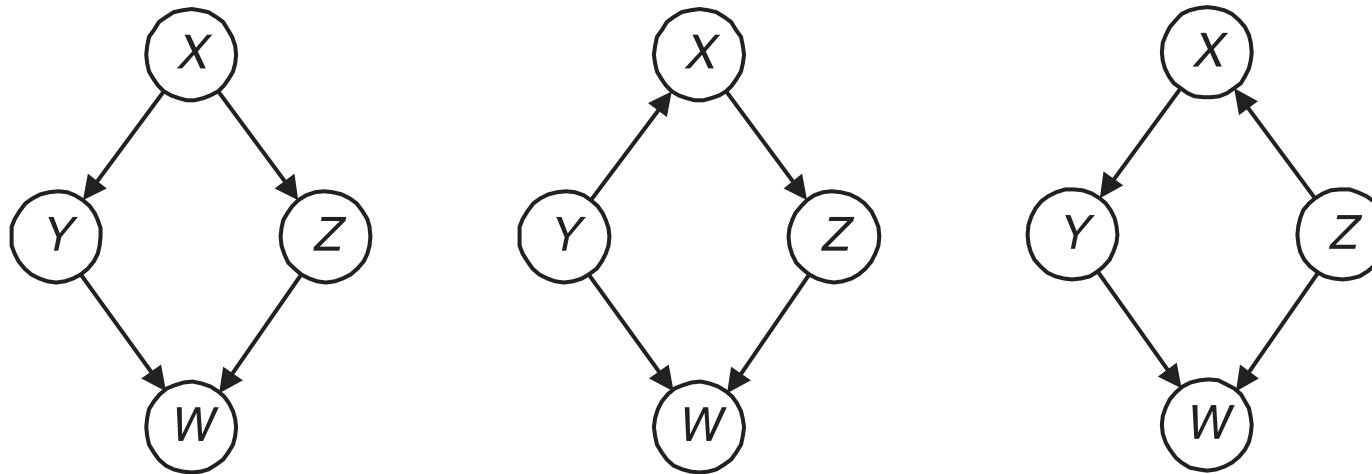


Figure 2.12: These DAGs are Markov equivalent, and there are no other DAGs Markov equivalent to them.

All and only d-separations:

- $I_G(\{Y\}, \{Z\} \mid \{X\})$
- $I_G(\{X\}, \{W\} \mid \{Y, Z\})$

Markov Equivalence (2)

Definition 2.7: Let $G_1 = (V, E_1)$ and $G_2 = (V, E_2)$ be two DAGs containing the same set of nodes V . Then G_1 and G_2 are called **Markov equivalent** if for every three mutually disjoint subsets $A, B, C \subseteq V$, A and B are d-separated by C in G_1 iff A and B are d-separated by C in G_2 . That is

$$I_{G_1}(A, B | C) \Leftrightarrow I_{G_2}(A, B | C)$$

Theorem 2.3: Two DAGs are Markov equivalent iff, based on the Markov condition, they entail the same conditional independencies I_P .

Identifying Markov Equivalence

Lemma 2.4: Let $G = (V, E)$ be a DAG and $X, Y \in V$. Then X and Y are adjacent in G iff they are not d-separated by some set in G .

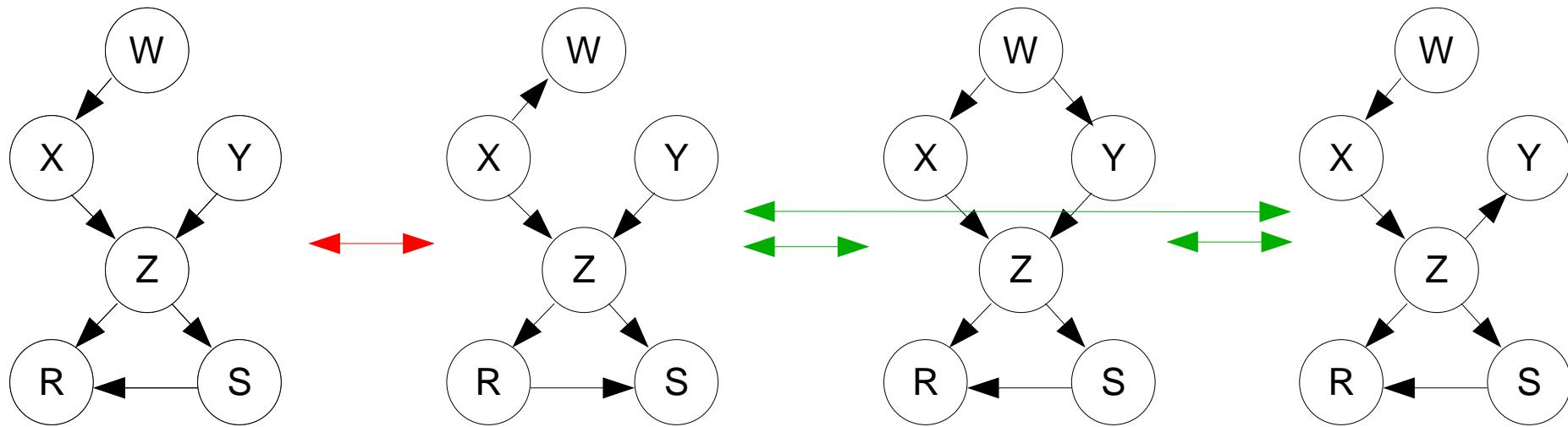
Corollary 2.2: Let $G = (V, E)$ be a DAG and $X, Y \in V$. Then X and Y are d-separated by some set, they are d-separated either by the set consisting of the parents of X or the set consisting of the parents of Y .

Lemma 2.5: Suppose we have a DAG $G = (V, E)$ and an uncoupled meeting $X—Z—Y$. Then the following are equivalent:

1. $X—Z—Y$ is a head-to-head meeting
2. There exists a set not containing Z that d-separated X and Y .
3. All the sets containing Z do not d-separated X and Y .

Theorem 2.4

Theorem 2.4: Two DAGs G_1 and G_2 are Markov equivalent iff they have the same links (edges without regard for direction) and the same set of uncoupled head-to-head meetings



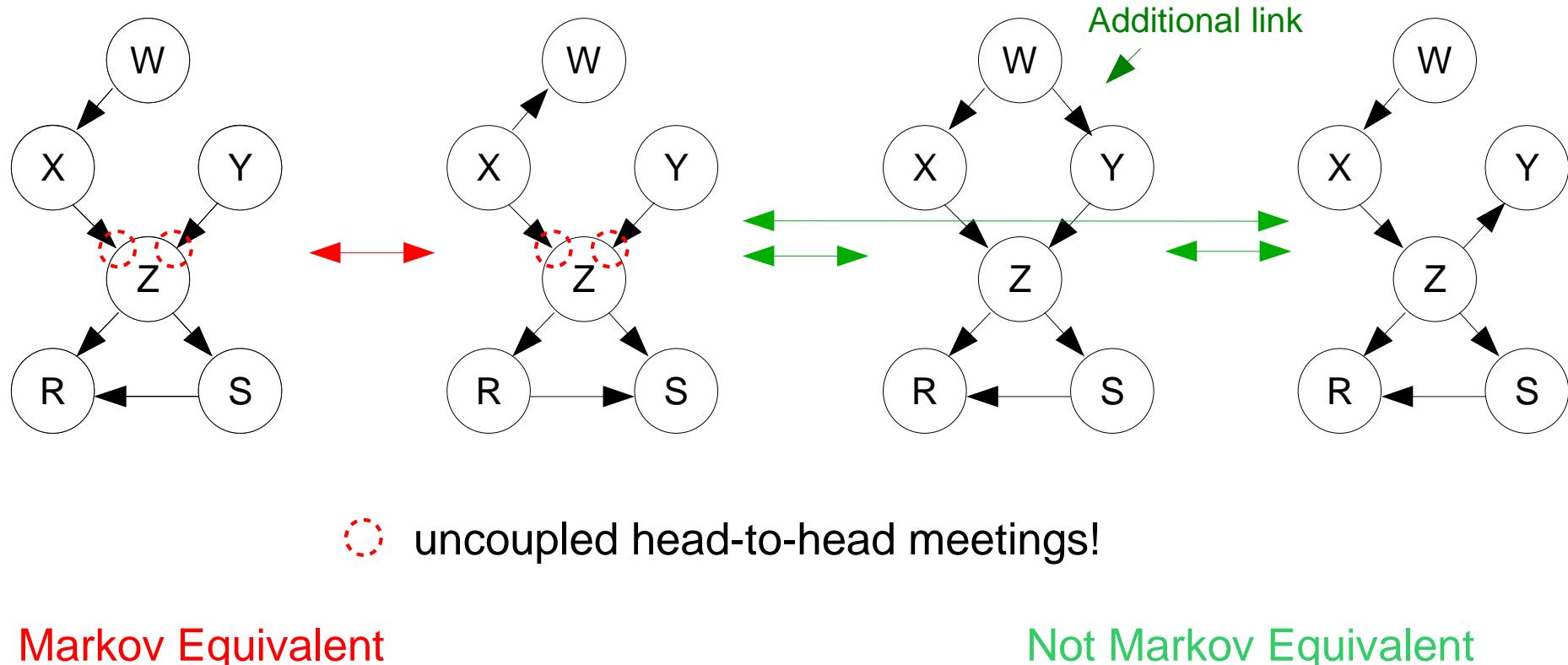
Search for uncoupled head-to-head meetings!

Markov Equivalent

Not Markov Equivalent

Theorem 2.4

Theorem 2.4: Two DAGs G_1 and G_2 are Markov equivalent iff they have the same links (edges without regard for direction) and the same set of uncoupled head-to-head meetings



Graph of a Markov equivalence Class

A **Markov equivalence class** can be represented by a graph with

1. Same links and
2. Same uncoupled head-to-head meetings

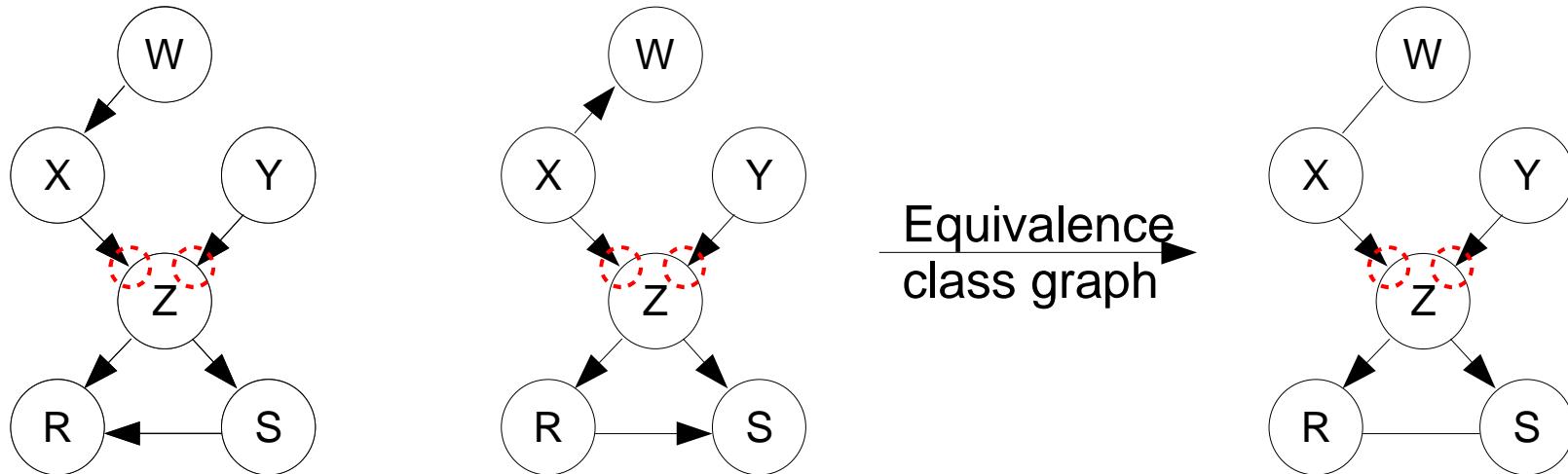
as the DAGs in the class.

Important: Any assignment of directions to the undirected edges in this graph, that does not create a new uncoupled head-to-head meeting or a directed cycle, yields a member of the equivalence class.

Often there are edges other than uncoupled head-to-head meetings, which must be oriented the same in Markov equivalent DAGs. For example, if all DAGs in a given Markov equivalence class have the edge $X \rightarrow Y$, and the uncoupled meeting $X \rightarrow Y - Z$ is not head-to-head, then all the DAGs in the equivalence class must have $Y - Z$ oriented as $Y \rightarrow Z$.

DAG pattern gp (1)

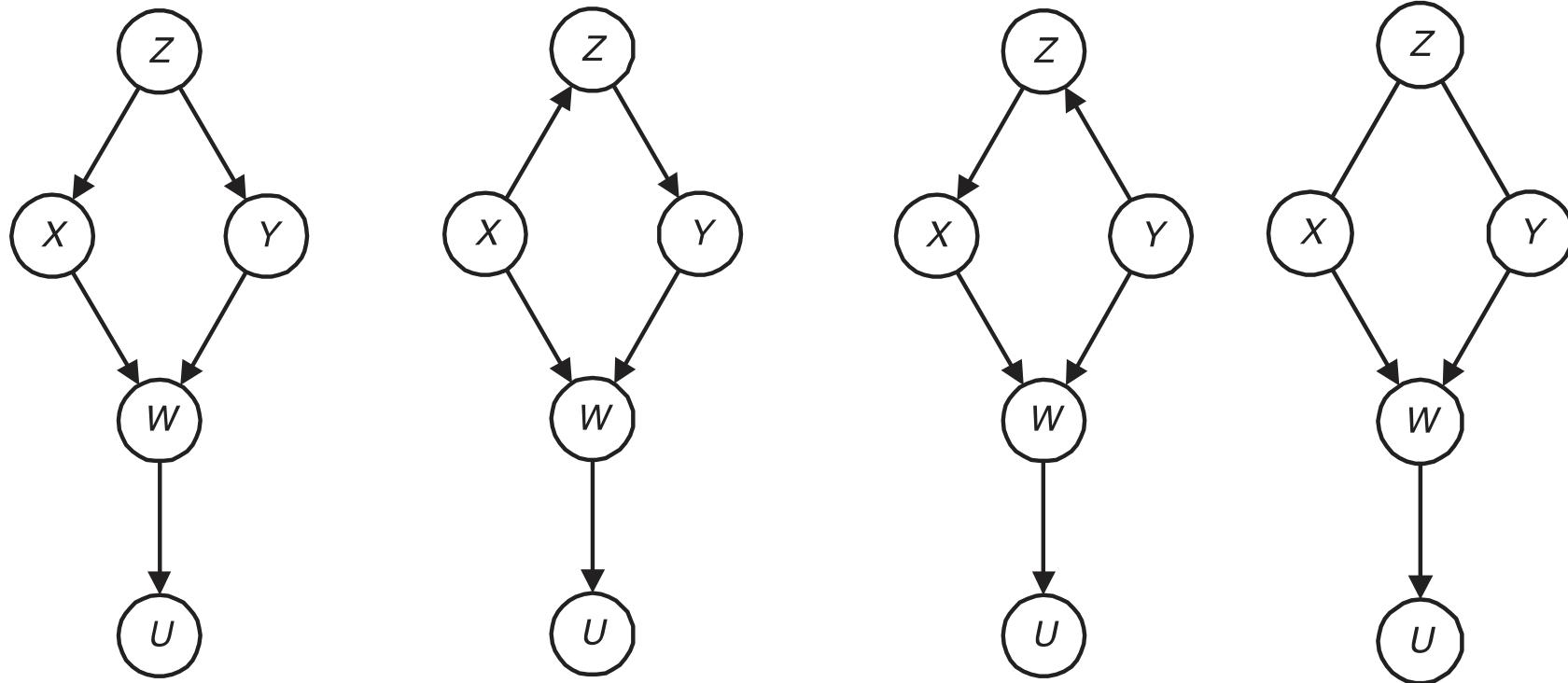
DAG pattern for a Markov equivalence class = graph that has the same links as the DAGs in the equivalence class and has oriented all and only the edges **common to all** of the DAGs in the equivalence class. The directed links in a DAG pattern are called **compelled edges**.



Definition 2.8: Let gp be a DAG pattern whose nodes are the elements of V , and let A , B , and C be mutually disjoint subsets of V . We say A and B are d -separated by C in gp if A and B are d -separated by C in any (and therefore every) DAG G in the Markov equivalence class represented by gp .

Thus the DAG pattern gp representing the equivalence class is an **independence map** of P .

DAG pattern gp (2)

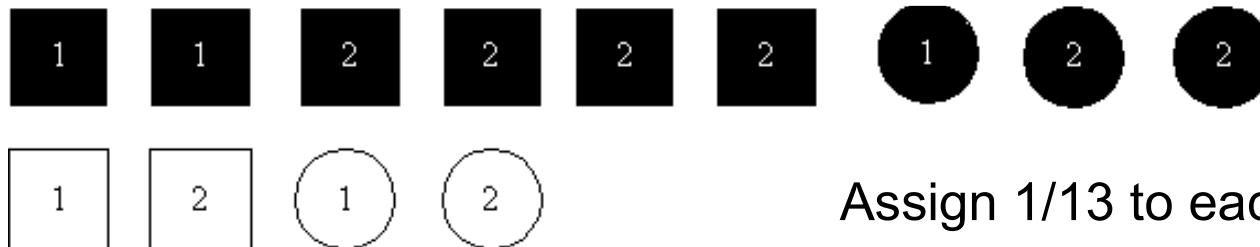


Markov Condition – Entailing Dependencies (1)

The Markov condition **only** entails independencies.

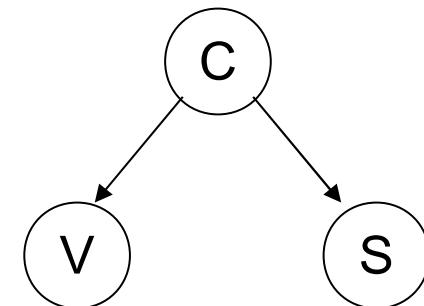
⇒ Many uninformative DAGs can satisfy the Markov condition with a given distribution P .

Example 2.7 (1)



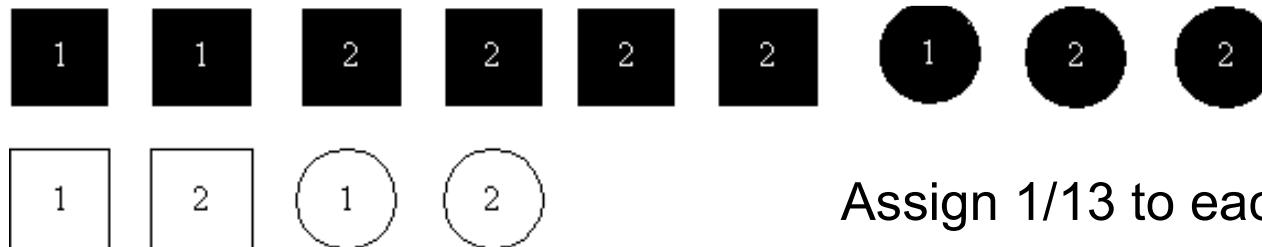
Assign 1/13 to each object $\rightarrow I_P(\{V\}, \{S\} \parallel C)$

Variable	Value	Outcomes Mapped to this Value
V	v1	All objects containing a “1”
	v2	All objects containing a “2”
S	s1	All square objects
	s2	All round objects
C	c1	All black objects
	c2	All white objects



\rightarrow satisfies Markov condition

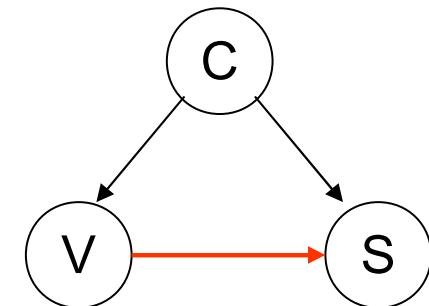
Example 2.7 (2)



Assign 1/13 to each object $\rightarrow I_P(\{V\}, \{S\} || \{C\})$

Variable	Value	Outcomes Mapped to this Value
V	v1	All objects containing a “1”
	v2	All objects containing a “2”
S	s1	All square objects
	s2	All round objects
C	c1	All black objects
	c2	All white objects

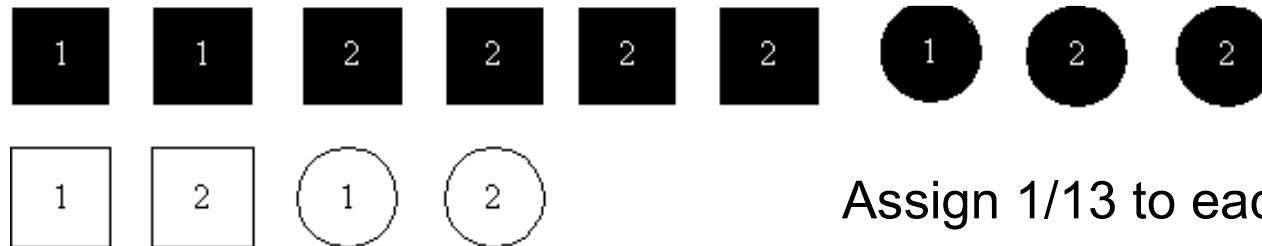
$$\begin{aligned} P(c, v, s) &= P(v, s|c) * P(c) \\ &= P(s|v, c) * P(v|c) * P(c) \end{aligned}$$



- satisfies Markov condition
- Any distribution does so (no entailing independencies)

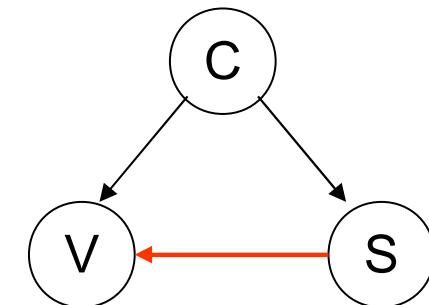
Complete DAG $G = (V, E)$ is one in which there is an edge between every pair of node, i.e. for every $X, Y \in V$, either $(X, Y) \in E$ or $(Y, X) \in E$.

Example 2.7 (3)



Assign 1/13 to each object $\rightarrow I_P(\{V\}, \{S\} | \{C\})$

$$\begin{aligned} P(c, v, s) &= P(v, s | c) * P(c) \\ &= P(v | s, c) * P(s | c) * P(c) \end{aligned}$$



- \rightarrow satisfies Markov condition
- \rightarrow Any distribution does so (no entailing independencies)

A complete DAG doesn't tell us anything about P .

Faithfulness

Given a probability distribution P of the variables in some set V and $X, Y \in V$, we say there is a **direct dependency** between X and Y in P if X and Y are not conditionally independent given any subset of $V - \{X, Y\}$.

Markov condition implies

“absence of edge between X and $Y \Rightarrow$ no direct dependency”.

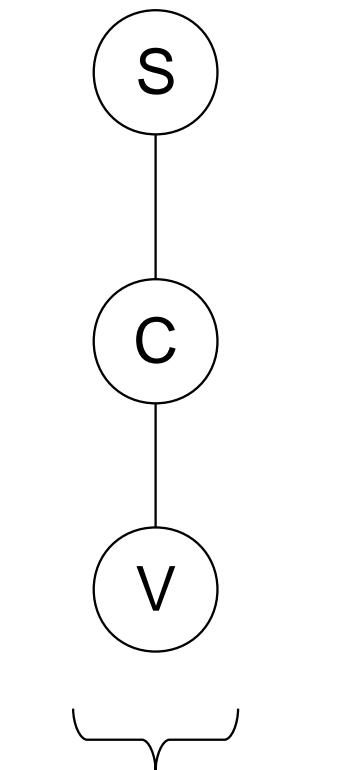
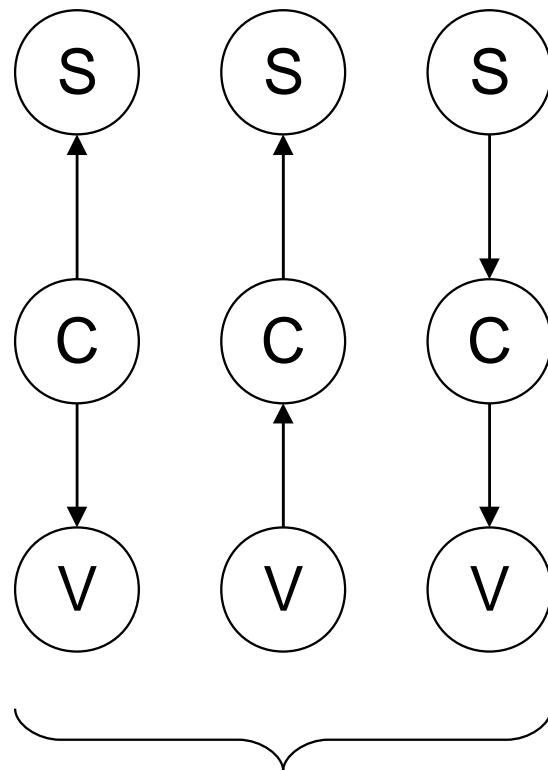
However, the existence of an edge between X and Y doesn't mean that there is a dependency.

But we want an edge to mean that there is a direct dependency!

Definition 2.9: Suppose we have a joint probability distribution P of the random variables in some set V and a DAG $G = (V, E)$. We say that (G, P) satisfies the **faithfulness condition** if, based on the Markov condition, G entails all and only conditional independencies in P . That is, the following two conditions hold:

1. (G, P) satisfies the Markov condition.
2. All conditional independencies in P are entailed by G , based on the Markov condition.

Example 2.8 (with values from Example 2.7)



For faithfulness show:

$$\text{Not } I_P(\{V\}, \{S\})$$

$$\text{Not } I_P(\{V\}, \{C\})$$

$$\text{Not } I_P(\{S\}, \{C\})$$

$$\text{Not } I_P(\{V\}, \{C\} | \{S\})$$

$$\text{Not } I_P(\{C\}, \{S\} | \{V\})$$

implicitly from this it follows:

$$\text{Not } I_P(\{C\}, \{S, V\})$$

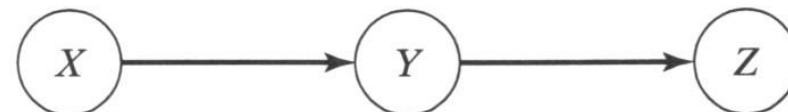
$$\text{Not } I_P(\{S\}, \{C, V\})$$

$$\text{Not } I_P(\{V\}, \{C, S\})$$

Theorem 2.5

When (G, P) satisfies the faithfulness condition, we say P and G are **faithful** to each other, and we say G is a **perfect map** of P . When (G, P) does not satisfy the faithfulness, we say they are **unfaithful** to each other.

Theorem 2.5: Suppose we have a joint probability distribution P of the random variables in some set V and a DAG $G=(V,E)$. Then (G,P) satisfies the faithfulness condition iff all and only conditional independencies in P are identified by d-separation in G .



$$\begin{array}{lll} P(x_1) = a & P(y_1|x_1) = 1 - (b + c) & P(z_1|y_1) = e \\ P(x_2) = 1 - a & P(y_2|x_1) = c & P(z_2|y_1) = 1 - e \\ & P(y_3|x_1) = b & \\ & & P(z_1|y_2) = e \\ & P(y_1|x_2) = 1 - (b + d) & P(z_2|y_2) = 1 - e \\ & P(y_2|x_2) = d & \\ & P(y_3|x_2) = b & P(z_1|y_3) = f \\ & & P(z_2|y_3) = 1 - f \end{array}$$

**Example:
not faithful**

For this (\mathbb{G}, P) , we have $I_P(\{X\}, \{Z\})$ but not $I_{\mathbb{G}}(\{X\}, \{Z\})$

Faithful DAG Pattern gp, Perfect map

Theorem 2.6: If (G, P) satisfies the faithfulness condition, then P satisfies this condition with all and only those DAGs that are Markov equivalent to G . Furthermore, if we let gp be the DAG pattern corresponding to this Markov equivalence class, the d-separations in gp identify all and only conditional dependencies in P . We say that gp and P are *faithful* to each other, and gp is a *perfect map* of P .

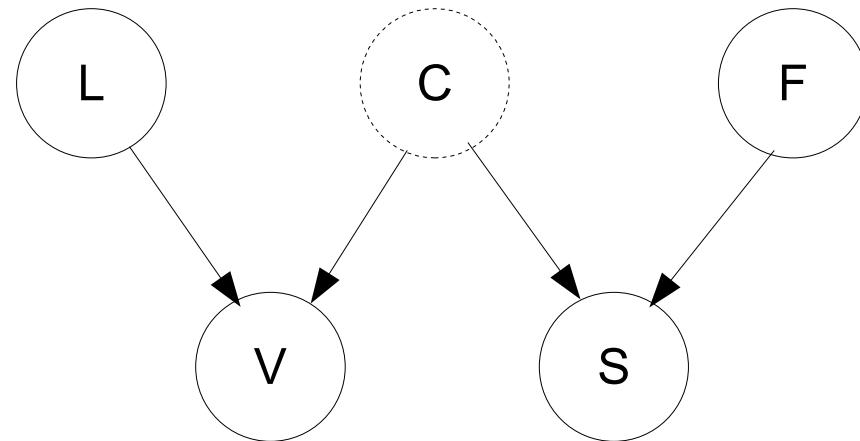
A distribution P *admits a faithful DAG representation* if P is faithful to some DAG (and therefore some DAG pattern).

⇒ **BUT:** not every P admits a faithful DAG representation.

Important goal: Whenever P admits a faithful DAG representation, we want to find it!

Example 2.11 (1)

Let P be a joint probability distribution of the variables in $W=\{L,V,C,S,F\}$, and $G=(W,E)$ be the following graph:



Let further (G,P) satisfies the faithfulness condition.

Then, the only independencies involving only the variables V, S, L and F are:

$$\begin{array}{lll} I_P(\{L\}, \{F, S\}) & I_P(\{L\}, \{S\}) & I_P(\{L\}, \{F\}) \\ I_P(\{F\}, \{L, V\}) & I_P(\{F\}, \{V\}) \end{array}$$

Example 2.11 (2)

Take the marginal distribution $P(v,s,l,f)$ of $P(v,s,c,l,f)$, i.e.,

$$P(v, s, l, f) = \sum_c P(v, s, c, l, f)$$

Theorem 2.5

\Rightarrow A DAG is faithful to $P' := P(v, s, l, f)$ iff all and only conditional independencies in P' are identified by d-separation in G .

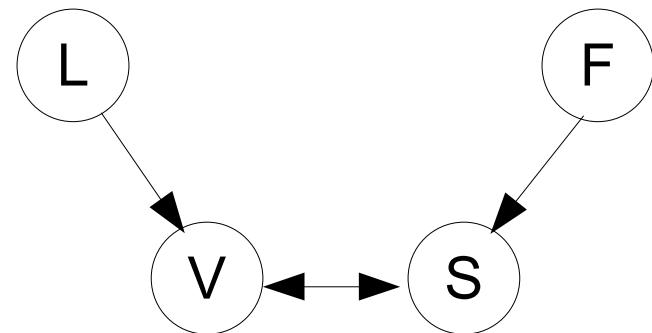
$$\begin{array}{lll} I_G(\{L\}, \{F, S\}) & I_G(\{L\}, \{S\}) & I_G(\{L\}, \{F\}) \\ I_G(\{F\}, \{L, V\}) & I_G(\{F\}, \{V\}) \end{array}$$

Lemma 2.4

\Rightarrow Links in G are $L-V$, $V-S$, and $S-F$
 $\Rightarrow L-V-S$ and $V-S-F$ are uncoupled meetings.

Lemma 2.5

- $\Rightarrow I_G(\{L\}, \{S\})$ implies $L-V-S$ is an uncoupled head-to-head meeting.
- $\Rightarrow I_G(\{V\}, \{F\})$ implies $V-S-F$ is an uncoupled head-to-head meeting.



- \Rightarrow The result graph is not a DAG
- \Rightarrow Contradiction
- $\Rightarrow P'$ does not have a faithful DAG representation.

Theorem 2.7

Theorem 2.7: Suppose we have a joint probability distribution P of the random variables in some set V and a DAG $G=(V,E)$. Then if P admits a faithful DAG representation, gp is the DAG pattern faithful to P iff the following two conditions hold:

1. X and Y are adjacent in gp iff there is no subsets $S \subseteq V$ such that $I_P(\{X\}, \{Y\}|S)$. That is, X and Y are adjacent iff there is a direct dependency between X and Y .
2. $X—Z—Y$ is a head-to-head meeting in gp iff $Z \in S$ implies $\neg I_P(\{X\}, \{Y\}|S)$.

Embedded Faithfulness (1)

The distribution $P(v, s, l, f)$ in Example 2.11 does not admit a faithful DAG representation. However, it is the marginal of a distribution which does, namely $P(v, s, c, l, f)$.

Definition 2.9: Suppose we have a joint probability distribution P of the random variables in some set V and a DAG $G=(V,E)$. We say that (G,P) satisfies the **faithfulness condition** if, based on the Markov condition, G entails all and only conditional independencies in P . That is, the following two conditions hold:

1. (G,P) satisfies the Markov condition.
2. All conditional independencies in P are entailed by G , based on the Markov condition.

Definition 2.10: Let P be a joint probability distribution of the variables in V where $V \subseteq W$, and let $G=(W,E)$ be a DAG. We say (G,P) satisfies the **embedded faithfulness condition** if the following two conditions hold:

1. Based on the Markov condition, G entails only conditional independencies in P for subset including only elements of V .
2. All conditional independencies in P are entailed by G , based on the Markov condition.

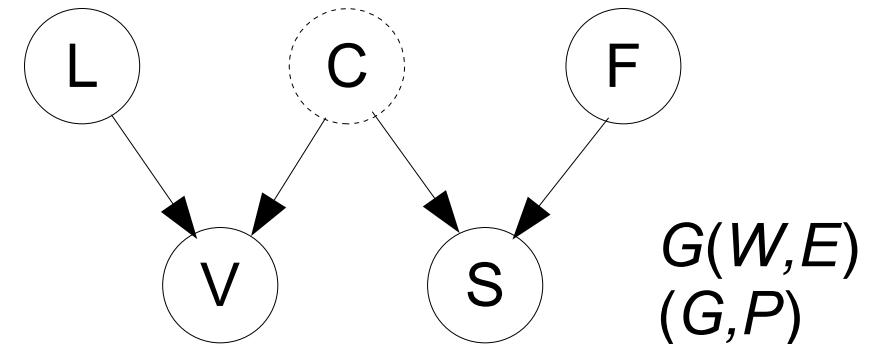
We say **P is embedded faithfully in G** .

Embedded Faithfulness (2)

Definition 2.10: Let P be a joint probability distribution of the variables in V where $V \subseteq W$, and let $G=(W,E)$ be a DAG. We say (G,P) satisfies the *embedded faithfulness condition* if the following two conditions hold:

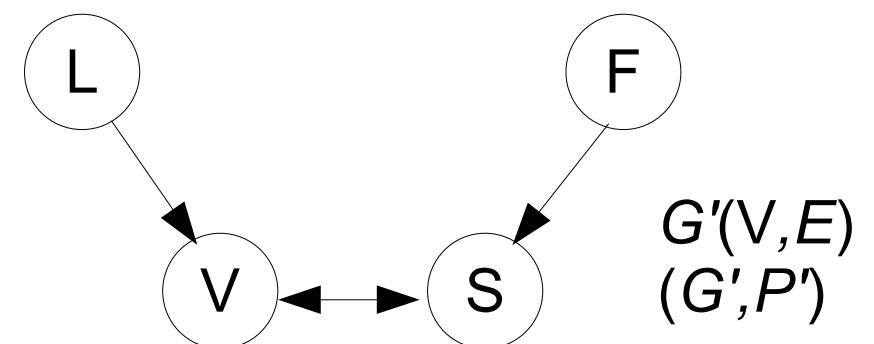
1. Based on the Markov condition, G entails only conditional independencies in P for subset including only elements of V .
2. All conditional independencies in P are entailed by G , based on the Markov condition.

We say **P is embedded faithfully in G** .



$$\begin{array}{lll} I_P(\{L\}, \{F, S\}) & I_P(\{L\}, \{S\}) & I_P(\{L\}, \{F\}) \\ I_P(\{F\}, \{L, V\}) & I_P(\{F\}, \{V\}) & \end{array}$$

Faithfulness is a special case of embedded faithfulness in which $W = V$.



Embedded Faithfulness (3)

Theorem 2.8: Let P be a joint probability distribution of the variables in W with $V \subseteq W$, and $G=(W,E)$. If (G,P) satisfies the faithfulness condition, and P' is the marginal distribution of V , then (G,P') satisfies the embedded faithfulness condition.

Theorem 2.9: Let P be a joint probability distribution of the variables in V where $V \subseteq W$, and $G=(W,E)$. Then (G,P) satisfies the embedded faithfulness condition iff all and only conditional independencies in P are identified by d-separation in G restricted to elements of V .

Two important Notes:

- Not every probability distribution P can be embedded faithfully into a DAG.
- If a distribution can be embedded faithfully, there are an infinite number of non-Markov equivalent DAGs in which it can be embedded faithfully.

Minimality (1)

Definition 2.11: Suppose we have a joint probability distribution P of the random variables in some set V and a DAG $G=(V,E)$. We say that (G,P) satisfies the **minimality condition** if the following two conditions hold:

- (G,P) satisfies the Markov condition.
- If we remove any edges from G , the resultant DAG no longer satisfies the Markov condition with P .

Theorem 2.11: Suppose we have a joint probability distribution P of the random variables in some set V and a DAG $G=(V,E)$.

- If (G,P) satisfies the faithfulness condition, then (G,P) satisfies the minimality condition.
- However, (G,P) can satisfy the minimality condition without satisfying the faithfulness condition.

Minimality (2)

Every probability distribution P satisfies the minimality condition with some DAG.

The following theorem gives a method for constructing such a DAG.

Theorem 2.12: Suppose we have a joint probability distribution P of the random variables in some set V . Create an arbitrary ordering of the nodes in V . For each $X \in V$, let B_X be the set of all nodes that come before X in the ordering, and let PA_X be a minimal subset of B_X such that $I_P(\{X\}, B_X | PA_X)$.

Create a DAG G by placing an edge from each node in PA_X to X . Then (G, P) satisfies the minimality condition. Furthermore, if P is strictly positive (i.e., there are no probability values equal 0), then PA_X is unique relative to the ordering.

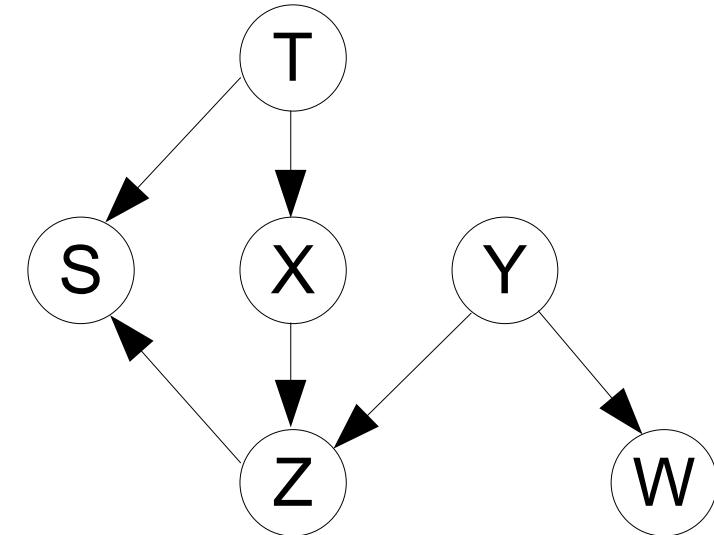
NOTE: A DAG satisfying the minimality condition with a distribution is not necessarily minimal with respect to have a minimal number of edges! However, a faithful DAG is minimal in this sense, too.

Markov Blanket (1)

Definition 2.12: Let V be a set of random variables, P be their joint probability distribution, and $X \in V$. Then a **Markov blanket** M_X of X is any set of variables such that X is conditionally independent of all the other variables given M_X . That is,

$$I_P(\{X\}, V - (M_X \cup \{X\}) | M_X)$$

Theorem 2.13: Suppose (G, P) satisfies the Markov condition. Then, for each variable X , the set of all parents of X , children of X , and parents of children of X is a Markov blanket of X .



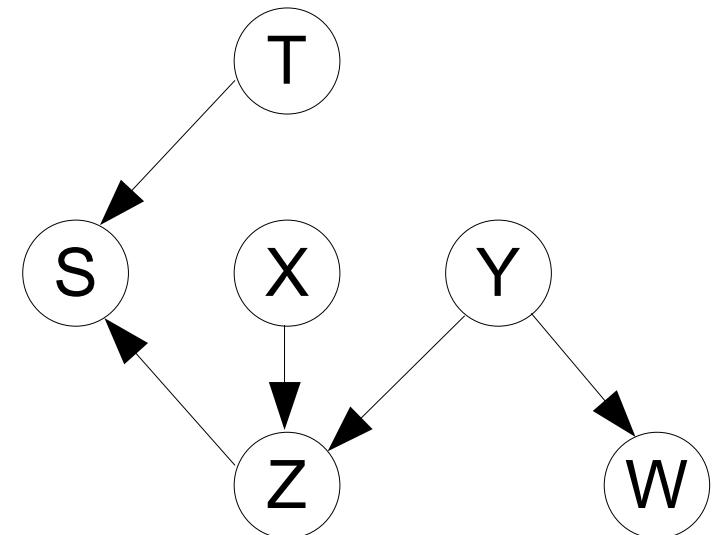
Examples:

$$M_X = \{ T, Y, Z \}$$

Markov Blanket (2)

Definition 2.12: Let V be a set of random variables, P be their joint probability distribution, and $X \in V$. Then a **Markov blanket** M_X of X is any set of variables such that X is conditionally independent of all the other variables given M_X . That is,

$$I_P(\{X\}, V - (M_X \cup \{X\}) | M_X)$$



Theorem 2.13: Suppose (G, P) satisfies the Markov condition. Then, for each variable X , the set of all parents of X , children of X , and parents of children of X is a Markov blanket of X .

Examples:

$$M_X = \{ T, Y, Z \}$$

--> not minimal any more

$$M_X = \{ Y, Z \}$$

Markov Boundary

Definition 2.13: Let V be a set of random variables, P be their joint probability distribution, and $X \in V$. Then a **Markov boundary** of X is any Markov blanket such that none of its proper subsets is a Markov blanket of X .

Theorem 2.14: Suppose (G, P) satisfies the faithfulness condition. Then for each variable X , the set of all parents of X , children of X , and parents of children of X is the unique Markov boundary of X .

Theorem 2.15: Suppose P is a strictly positive probability distribution of the variables in V . Then for each $X \in V$ there is a unique Markov boundary of X .

SS 2015 – Bayesian Networks

Face Detection

*University of Augsburg
Multimedia Computing and Computer Vision,
Prof. Dr. Rainer Lienhart
Rainer.Lienhart@informatik.uni-augsburg.de
www.multimedia-computing.org*

Reference

BEISPIEL Face Detection

SS 2015 – Bayesian Networks

Inference with Discrete Variables

University of Augsburg

Multimedia Computing and Computer Vision,

Prof. Dr. Rainer Lienhart

Rainer.Lienhart@informatik.uni-augsburg.de

www.multimedia-computing.org

Reference

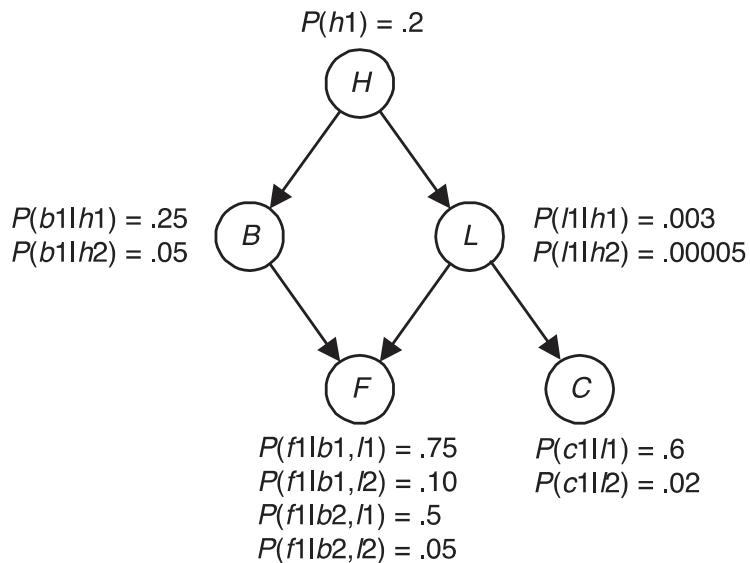
Richard E. Neapolitan. **Learning Bayesian Networks.** *Prentice Hall Series in Artificial Intelligence*, ISBN 0-13-012534-2.

Don't forget. Reading the book chapters 1 – 6 is mandatory.

Chapter on ***Inference: Discrete Variables***
(chapter 3)

Figures and text are taken from that book

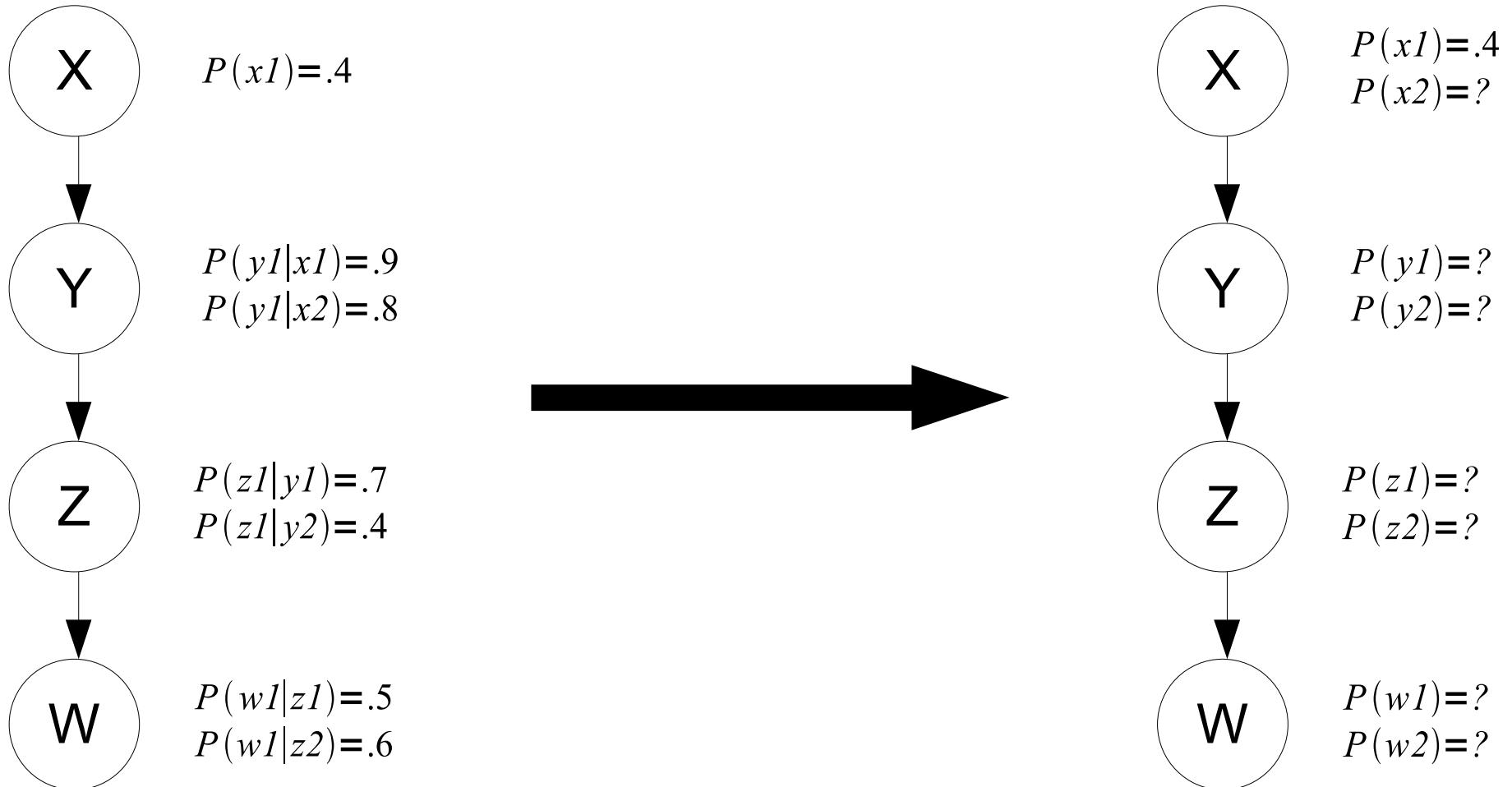
Repeat Example



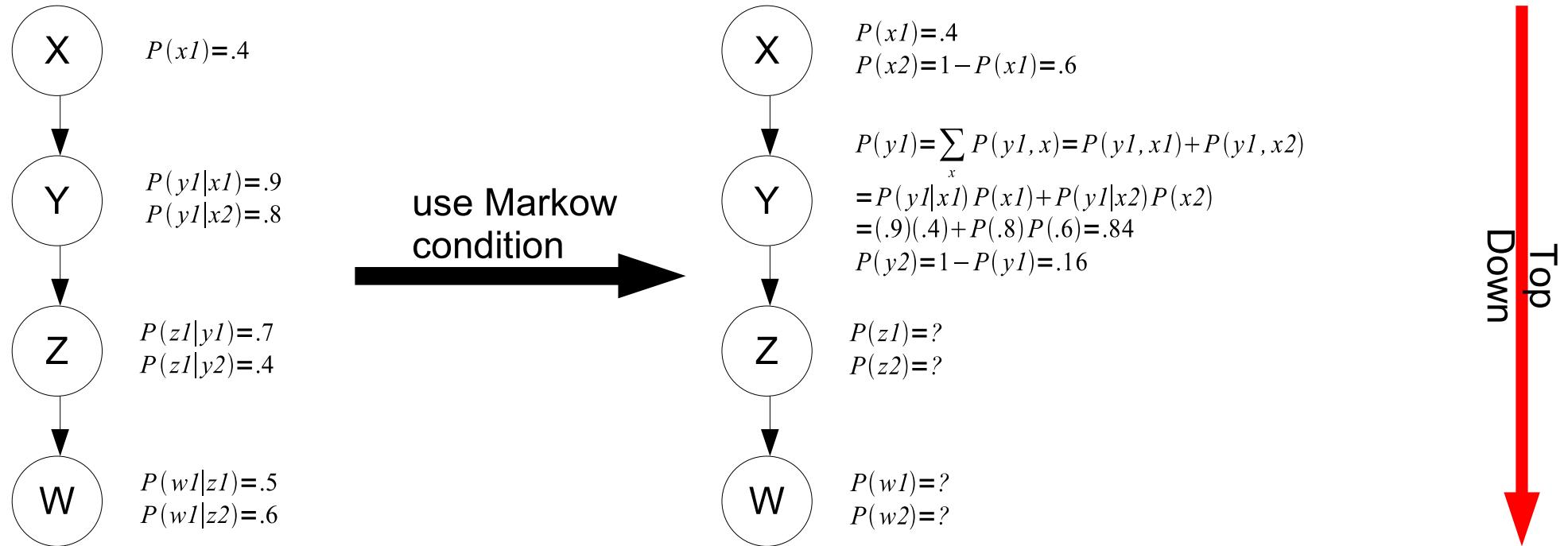
Feature	Value	When the Feature Takes this Value
H	h_1	There is a history of smoking
	h_2	There is no history of smoking
B	b_1	Bronchitis is present
	b_2	Bronchitis is absent
L	l_1	Lung cancer is present
	l_2	Lung cancer is absent
F	f_1	Fatigue is present
	f_2	Fatigue is absent
C	c_1	Chest X-ray is positive
	c_2	Chest X-ray is negative

- $P(b1|h1, c1) =?$ or $P(f1|h1, c1) =?$
 - ⇒ Brute force algorithm: Bad idea. Requires full joint probability $P(H, B, L, F, C)$
 - Ordinarily not available prob. distribution
 - Exponential space and time complexity
 - ⇒ Will develop algorithm to perform that type of inference.

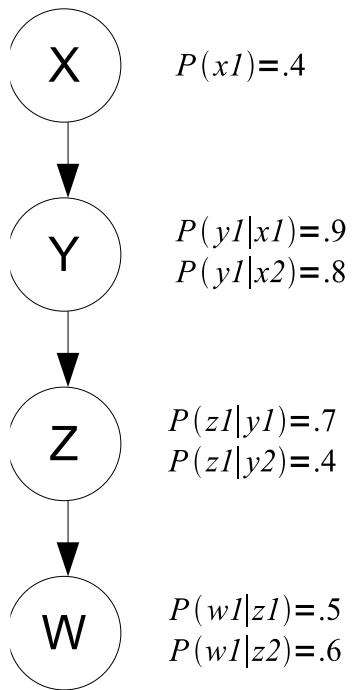
Example – Down Propagation (1) Prior Probabilities



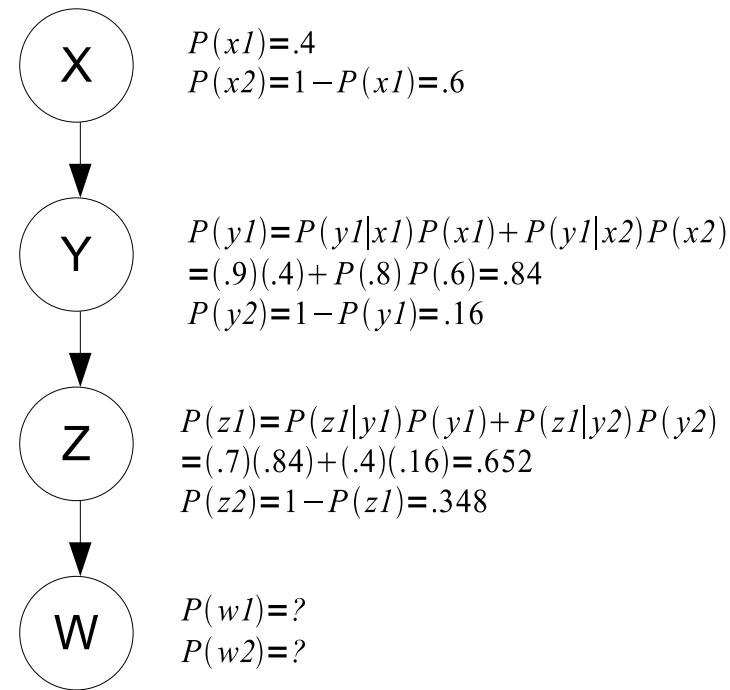
Example – Down Propagation (2) of Evidence (Instantiated Variable)



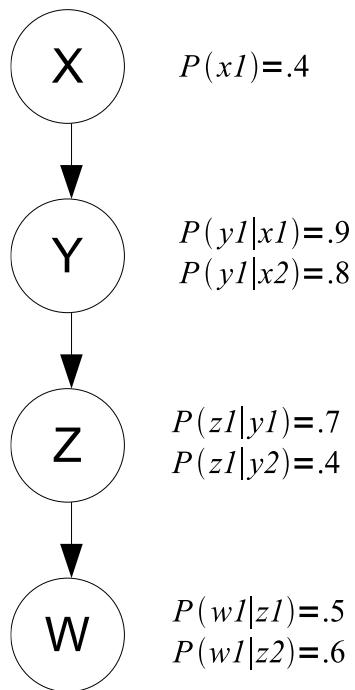
Example – Down Propagation (3)



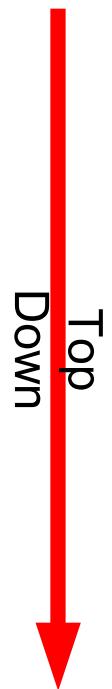
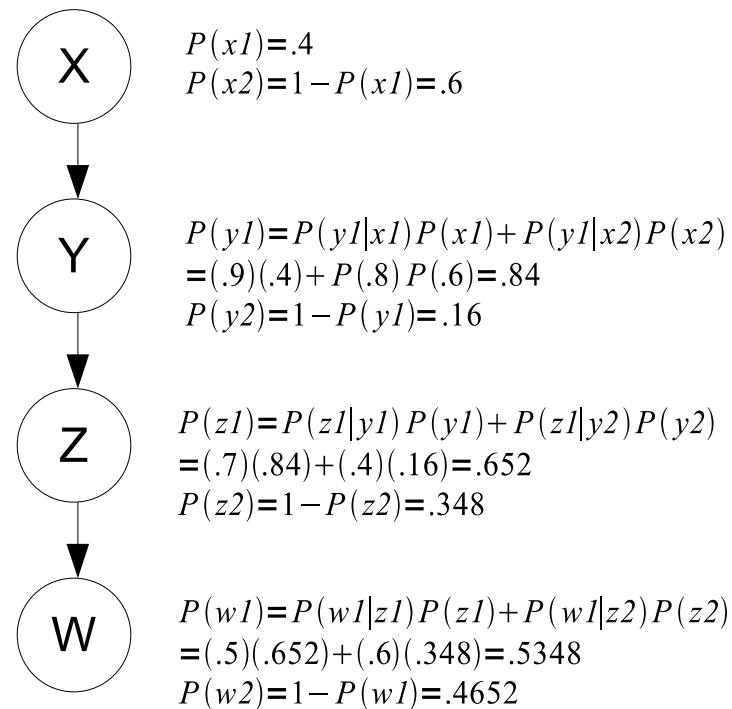
use Markow condition



Example – Down Propagation (4)



use Markow condition



'Messages' are passed down from the top to the bottom.

⇒ Works on arbitrarily long linked lists and trees.

Example – Down Propagation (5)

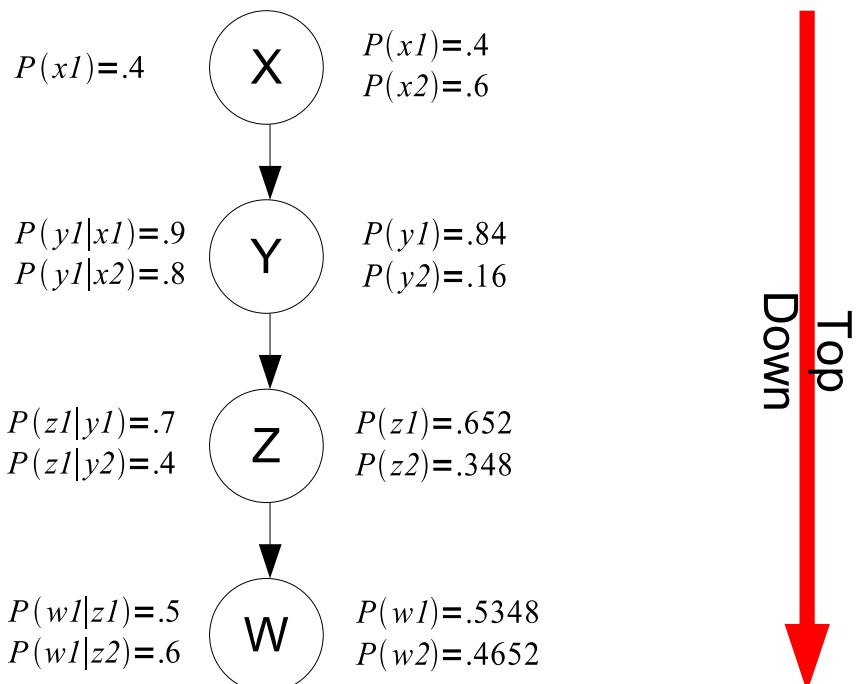
Add Evidence:

Assume X is instantiated to $x1$:

$$P(y1|x1) = .9$$

$$\begin{aligned} P(z1|x1) &= P(z1|y1, x1)P(y1|x1) + P(z1|y2, x1)P(y2|x1) \\ &= P(z1|y1)P(y1|x1) + P(z1|y2)P(y2|x1) \\ &= (.7)(.9) + (.4)(.1) = .67 \end{aligned}$$

$$\begin{aligned} P(w1|x1) &= P(w1|z1, x1)P(z1|x1) + P(w1|z2, x1)P(z2|x1) \\ &= P(w1|z1)P(z1|x1) + P(w1|z2)P(z2|x1) \\ &= (.8)(.67) + (.6)(.33) = .734 \end{aligned}$$



'Messages' are passed down from the top to the bottom.

⇒ Works on arbitrarily long linked lists and trees.

Example shows how to use down propagation of messages to compute the conditional probabilities of variables below the instantiated variable.

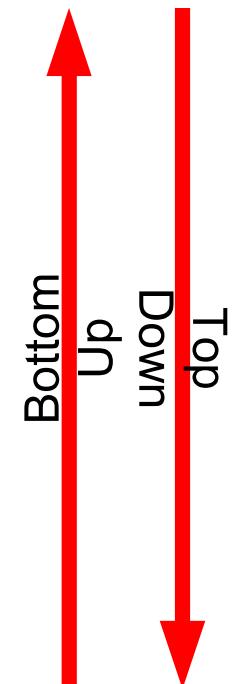
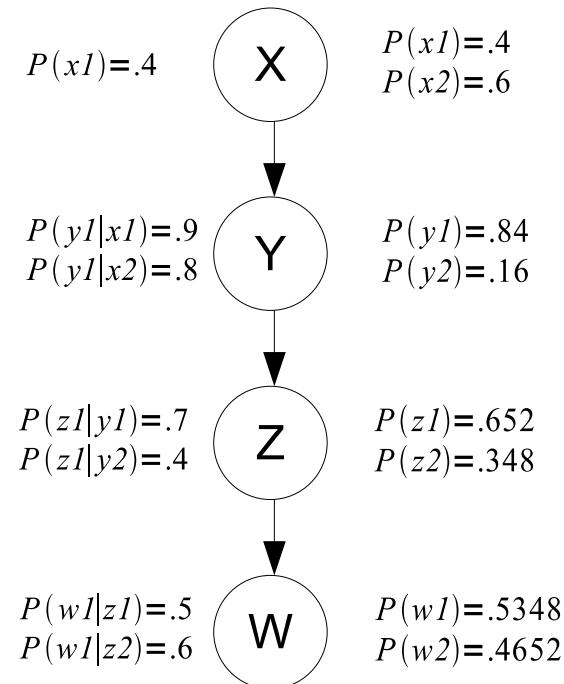
Example – Upward Propagation (1)

Assume only W is instantiated to $w1$:

$$P(z1|w1)=?$$

$$P(y1|w1)=?$$

TIP: Apply Bayes



Example – Upward Propagation (2)

Assume only W is instantiated to $w1$:

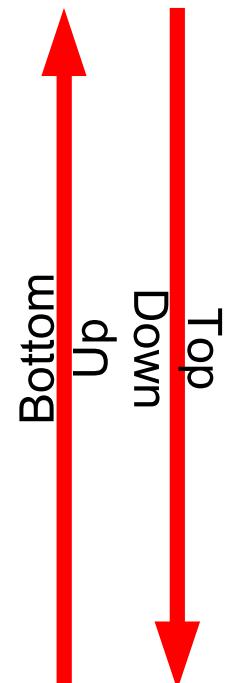
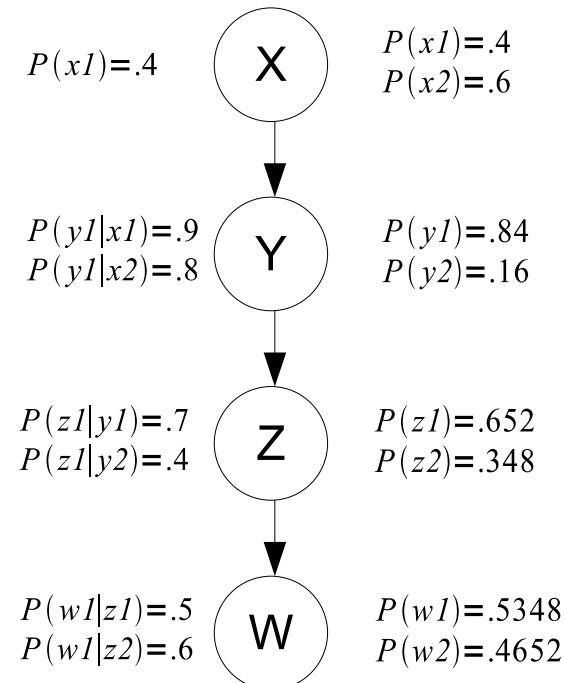
$$P(z1|w1) = \frac{P(w1|z1)P(z1)}{P(w1)} = \frac{(.5)(.652)}{.5348} = 0.6096$$

$$P(y1|w1) = \frac{P(w1|y1)P(y1)}{P(w1)} = \frac{P(w1|y1)(.84)}{.5348}$$

'Messages' are passed upward from the bottom to the top

'Messages' are passed down from the top to the bottom

$$\begin{aligned} P(w1|y1) &= P(w1|z1, y1)P(z1|y1) + P(w1|z2, y1)P(z2|y1) \\ &= P(w1|z1)P(z1|y1) + P(w1|z2)P(z2|y1) \\ &= (.5)(.7) + (.6)(.3) = 0.53 \end{aligned}$$



Example – Upward Propagation (3)

Assume W is instantiated to $w1$:

$$P(z1|w1) = .6096$$

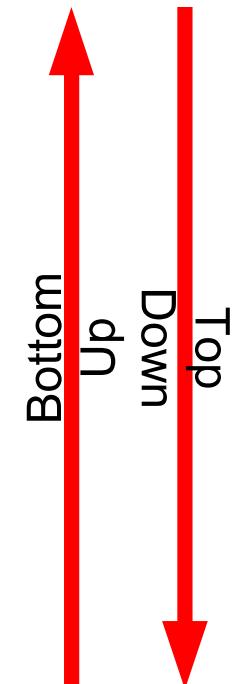
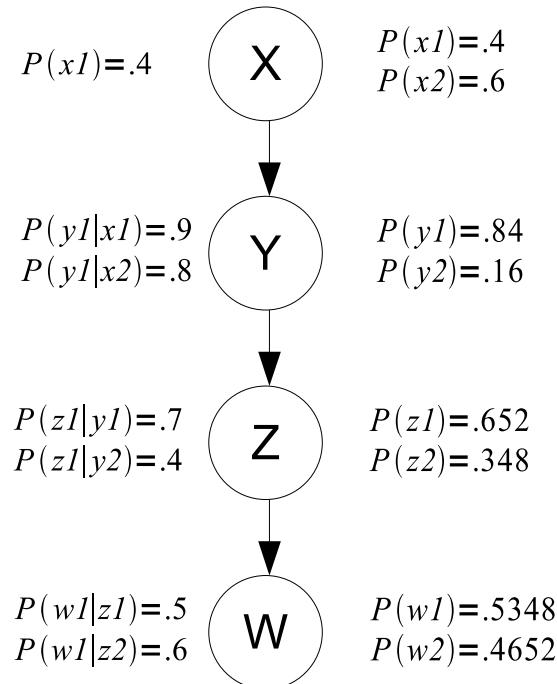
$$P(y1|w1) = \frac{(.53)(.84)}{.5348} = 0.832$$

$$P(x1|w1) = \frac{P(w1|x1)P(x1)}{P(w1)} = \frac{P(w1|x1)(.4)}{.5348}$$

'Messages' are passed down from the top to the bottom



$$\begin{aligned} P(w1|x1) &= P(w1|y1)P(y1|x1) + P(w1|y2)P(y2|x1) \\ &= P(w1|y1)(.9) + P(w1|y2)(.1) \\ &= [P(w1|z1, y1)P(z1|y1) + P(w1|z2, y1)P(z2|y1)](.9) + [P(w1|z1, y2)P(z1|y2) + P(w1|z2, y2)P(z2|y2)](.1) \\ &= [P(w1|z1)P(z1|y1) + P(w1|z2)P(z2|y1)](.9) + [P(w1|z1)P(z1|y2) + P(w1|z2)P(z2|y2)](.1) \\ &= [(.) (.7) + (.6) (.3)](.9) + [(.) (.4) + (.6) (.6)](.1) \\ &= ... \end{aligned}$$



Python - Code

```
from bayesian.bbn import *

def fX(X):
    '''X'''
    if X:
        return 0.4
    else:
        return 0.6

def fY(Y, X):
    '''Y'''
    table = dict()
    table['tt'] = 0.9
    table['tf'] = 0.8
    table['ft'] = 1.0 - table['tt']
    table['ff'] = 1.0 - table['tf']
    key = ''
    key = key + 't' if Y else key + 'f'
    key = key + 't' if X else key + 'f'
    return table[key]

def fZ(Z, Y):
    '''Z'''
    table = dict()
    table['tt'] = 0.7
    table['tf'] = 0.4
    table['ft'] = 1.0 - table['tt']

    table['ff'] = 1.0 - table['tf']
    key = ''
    key = key + 't' if Z else key + 'f'
    key = key + 't' if Y else key + 'f'
    key = key + 't' if X else key + 'f'
    return table[key]

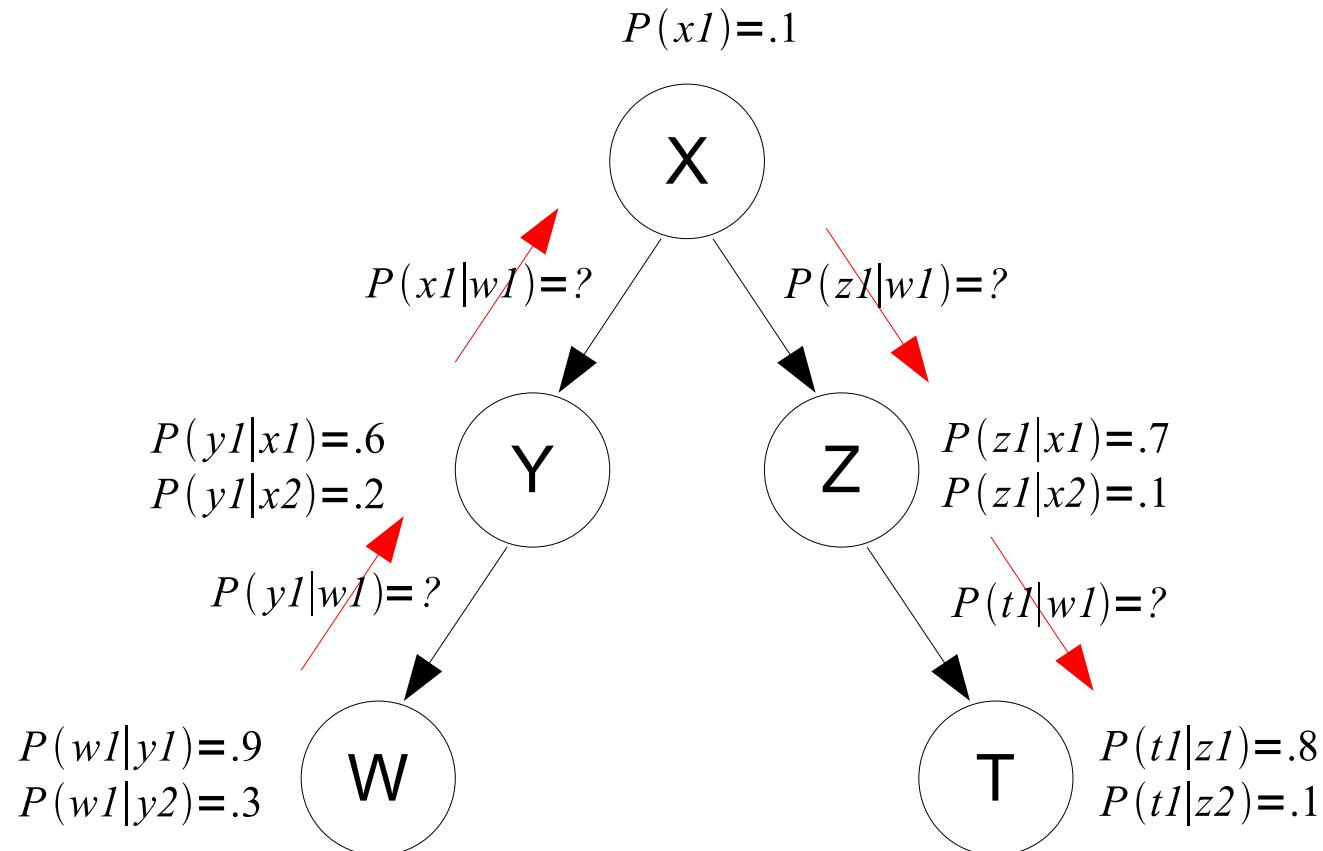
    table['ff'] = 1.0 - table['tf']
    key = ''
    key = key + 't' if Z else key + 'f'
    key = key + 't' if Y else key + 'f'
    return table[key]

def fW(W, Z):
    '''W'''
    table = dict()
    table['tt'] = 0.5
    table['tf'] = 0.6
    table['ft'] = 1.0 - table['tt']
    table['ff'] = 1.0 - table['tf']
    key = ''
    key = key + 't' if W else key + 'f'
    key = key + 't' if Z else key + 'f'
    return table[key]

if __name__ == '__main__':
    g = build_bbn(
        fx, fY, fz, fw)
    g.q()
    g.q(X=True)
    g.q(W=True)
```

Tree Example

Assume W is instantiated to $w1$:



Inference Task (1)

- We observe the values of some variables in set $E \subseteq V$.
- They are called the **evidence variables** $E = \{X_{e1}, \dots, X_{e|E|}\}$
- Inference
 - ⇒ Compute conditional probability $P(X_i|E)$ for all **non-evidential nodes** $X_i \in V - E$.

Inference Algorithms

- Exact inference algorithms
 - Belief propagation (e.g., Pearl's Message Passing Algorithm)
 - Junction tree
 - SPI (Symbolic Probabilistic Inference)
 - ⇒ Computationally involving – in worst case NP-hard
- Approximate inference algorithms
 - Using Logic Sampling
 - Using Likelihood Weighting

Inference Task (2)

In other words:

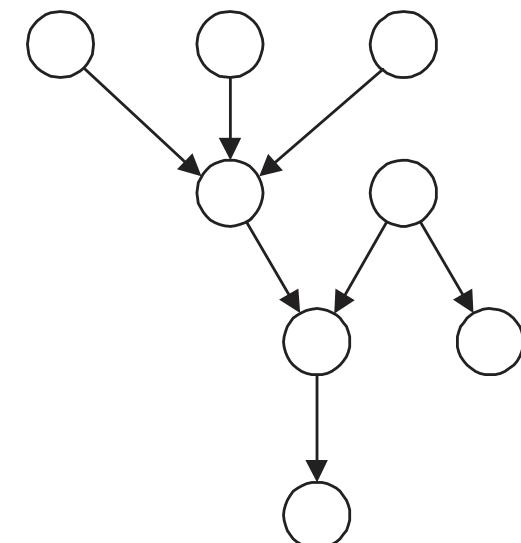
- Given a set \mathbf{a} of values of a set \mathbf{A} of instantiated variables, determine $P(x|\mathbf{a})$ for all values x of each variable X in the network.

Definitions:

- Rooted Tree** := a DAG in which there is
 1. a unique node called the root, which has no parent,
 2. every other node has precisely one parent, and
 3. every node is a descendent of the root.
- Singly Connected DAG** := if there is at most one chain between any two nodes. Otherwise it is called ***multiply connected***.

Every tree is singly connected.

However, nodes in a singly connected DAG can have more than one parent.



Pearl's Message-Passing Algorithm (1)

We will derive Pearl's algorithm for the case in which

- The DAG is a rooted tree and
- Each node has precisely two children

Let \mathbf{D}_X be the subset of \mathbf{A} containing all members of \mathbf{A} that are in the subtree rooted at X (therefore, including X if $X \in \mathbf{A}$), and \mathbf{N}_X be the subset of \mathbf{A} containing all members of \mathbf{A} that are nondescendants of X . Recall X is a nondescendent of X ; so this set includes X if $X \in \mathbf{A}$. Note also that $A = \mathbf{N}_X \cup \mathbf{D}_X$. This situation is depicted in the Figure. We have for each value of x ,

$$\begin{aligned}
 P(x|\mathbf{a}) &= P(x|\mathbf{d}_X, \mathbf{n}_X) \\
 &= \frac{P(\mathbf{d}_X, \mathbf{n}_X|x)P(x)}{P(\mathbf{d}_X, \mathbf{n}_X)} \\
 &= \frac{P(\mathbf{d}_X|x)P(\mathbf{n}_X|x)P(x)}{P(\mathbf{d}_X, \mathbf{n}_X)} \\
 &= \frac{P(\mathbf{d}_X|x)P(x|\mathbf{n}_X)P(\mathbf{n}_X)P(x)}{P(x)P(\mathbf{d}_X, \mathbf{n}_X)} \\
 &= \beta P(\mathbf{d}_X|x)P(x|\mathbf{n}_X),
 \end{aligned}$$

d-separation

Bayes

Constant that does not depend on the value of x

Pearl's Message-Passing Algorithm (2)

We will develop functions $\lambda(x)$ and $\pi(x)$ such

$$\begin{aligned}\lambda(x) &\simeq P(\mathbf{d}_X|x) \\ \pi(x) &\simeq P(x|\mathbf{n}_X).\end{aligned}$$

By \simeq we mean ‘proportional to’. That is, $\pi(x)$, for example, may not equal $P(x|\mathbf{n}_X)$, but it equals a constant times $P(x|\mathbf{n}_X)$, where the constant does not depend on the value of x . Once we do this, due to Equality 3.1, we will have

$$P(x|\mathbf{a}) = \alpha \lambda(x) \pi(x),$$

where α is a normalizing constant that does not depend on the value of x .

1. Develop $\lambda(x)$: We need

$$\lambda(x) \simeq P(\mathbf{d}_X|x). \quad (3.2)$$

CASE 1: $X \in \mathcal{A}$ and X ’s value is \hat{x} . Since $X \in \mathcal{D}_X$,

$$P(\mathbf{d}_X|x) = 0 \quad \text{for } x \neq \hat{x}.$$

So to achieve Proportionality 3.2, we can set

$$\begin{aligned}\lambda(\hat{x}) &\equiv 1 \\ \lambda(x) &\equiv 0 \quad \text{for } x \neq \hat{x}.\end{aligned}$$

Pearl's Message-Passing Algorithm (3)

CASE 2: $X \notin A$ and X is a leaf.

In this case $\mathbf{d}_X = \emptyset$, the empty set of the variables, and so

$$P(\mathbf{d}_X|x) = P(\emptyset|x) = 1 \text{ for all values of } x.$$

So to achieve Proportionality 3.2, we can set

$$\lambda(x) \equiv 1 \quad \text{for all values of } x.$$

CASE 3: $X \notin A$ and X is a nonleaf. Let Y be X 's left child. W be X 's right child. Then since $X \notin A$,

$$\mathbf{D}_X = \mathbf{D}_Y \cup \mathbf{D}_W.$$

This situation is depicted in the figure right.

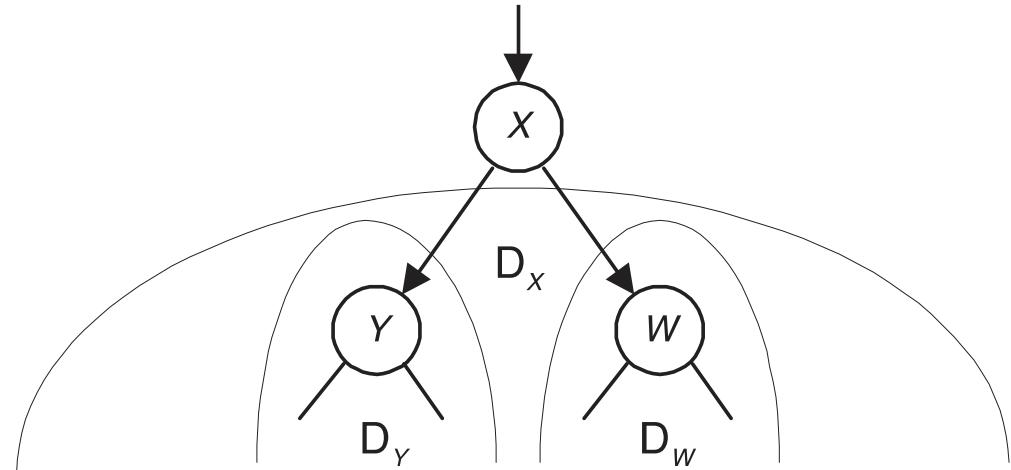


Figure 3.5: If X is not in A , then $\mathbf{D}_X = \mathbf{D}_Y \cup \mathbf{D}_W$.

Pearl's Message-Passing Algorithm (4)

Lambda messages
are upward messages

$$\begin{aligned} P(\mathbf{d}_X|x) &= P(\mathbf{d}_Y, \mathbf{d}_W|x) \\ &= P(\mathbf{d}_Y|x)P(\mathbf{d}_W|x) \\ &= \sum_y P(y|x)P(\mathbf{d}_Y|y) \sum_w P(w|x)P(\mathbf{d}_W|w) \\ &\leq \sum_y P(y|x)\lambda(y) \sum_w P(w|x)\lambda(w). \end{aligned}$$

The second equality is due to *d-separation* and the third to the law of total probability. So we can achieve Proportionality 3.2 by defining for all values of x ,

$$\begin{aligned} \lambda_Y(x) &\equiv \sum_y P(y|x)\lambda(y) \\ \lambda_W(x) &\equiv \sum_w P(w|x)\lambda(w), \end{aligned}$$

and setting

$$\lambda(x) \equiv \lambda_Y(x)\lambda_W(x) \quad \text{for all values of } x.$$

Pearl's Message-Passing Algorithm (5)

2. Develop $\pi(x)$: We need

$$\pi(x) \simeq P(x|\mathbf{n}_X). \quad (3.3)$$

CASE 1: $X \in \mathcal{A}$ and X 's value is \hat{x} . Due to the fact that $X \in \mathcal{N}_X$,

$$\begin{aligned} P(\hat{x}|\mathbf{n}_X) &= P(\hat{x}|\hat{x}) = 1 \\ P(x|\mathbf{n}_X) &= P(x|\hat{x}) = 0 \quad \text{for } x \neq \hat{x}. \end{aligned}$$

So we can achieve Proportionality 3.3 by setting

$$\begin{aligned} \pi(\hat{x}) &\equiv 1 \\ \pi(x) &\equiv 0 \quad \text{for } x \neq \hat{x}. \end{aligned}$$

CASE 2: $X \notin \mathcal{A}$ and X is the root. In this case $\mathbf{n}_X = \emptyset$, the empty set of random variables, and so

$$P(x|\mathbf{n}_X) = P(x|\emptyset) = P(x) \quad \text{for all values of } x.$$

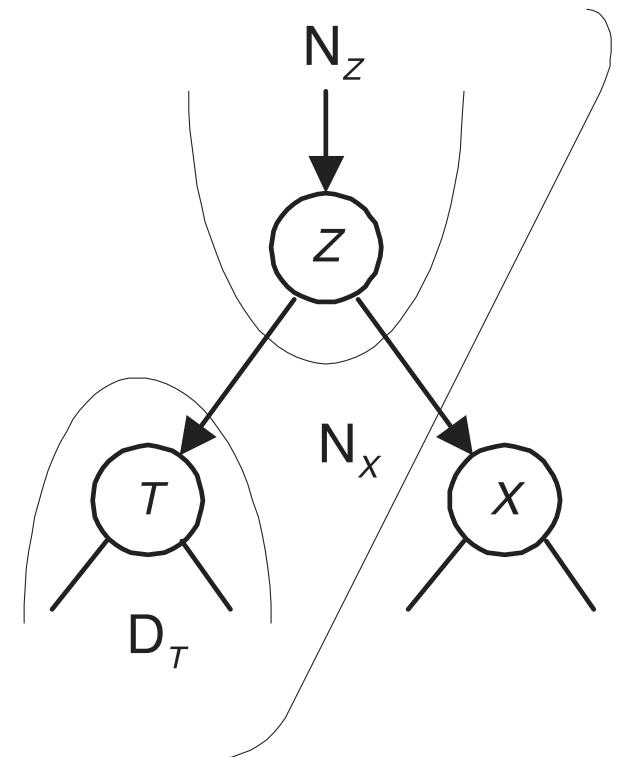
So we can achieve Proportionality 3.3 by setting

$$\pi(x) \equiv P(x) \quad \text{for all value of } x.$$

Pearl's Message-Passing Algorithm (6)

CASE 3: $X \notin \mathbf{A}$ and X is not the root. Without loss of generality assume X is Z 's right child, and let T be Z 's left child. Then $\mathbf{N}_X = \mathbf{N}_Z \cup \mathbf{D}_T$. The situation is depicted in the figure. We have

$$\begin{aligned}
 P(x|\mathbf{n}_X) &= \sum_z P(x|z)P(z|\mathbf{n}_X) \\
 &= \sum_z P(x|z)P(z|\mathbf{n}_Z, \mathbf{d}_T) \\
 &= \sum_z P(x|z) \frac{P(\mathbf{n}_Z, \mathbf{d}_T|z)P(z)}{P(\mathbf{n}_Z, \mathbf{d}_T)} \\
 &= \sum_z P(x|z) \frac{P(\mathbf{n}_Z|z)P(\mathbf{d}_T|z)P(z)}{P(\mathbf{n}_Z, \mathbf{d}_T)} \\
 &= \sum_z P(x|z) \frac{P(z|\mathbf{n}_Z)P(\mathbf{n}_Z)P(\mathbf{d}_T|z)P(z)}{P(z)P(\mathbf{n}_Z, \mathbf{d}_T)} \\
 &= \sum_z P(x|z) \frac{P(z|\mathbf{n}_Z)P(\mathbf{n}_Z)P(\mathbf{d}_T|z)}{P(\mathbf{n}_Z, \mathbf{d}_T)} \\
 &= \gamma \sum_z P(x|z)P(z|\mathbf{n}_Z)P(\mathbf{d}_T|z) \\
 &= \gamma \sum_z P(x|z)\pi(z)\lambda_T(z)
 \end{aligned}$$



Pearl's Message-Passing Algorithm (7)

We can achieve Proportionality 3.3 by defining for all values of z ,

$$\pi_X(z) \equiv \pi(z)\lambda_T(z),$$

and setting

$$\pi(x) = \sum_z P(x|z)\pi_X(z) \quad \text{for all values of } x.$$

Pearl's Message-Passing Algorithm (8)

1. Define λ messages:

For each child Y of X , for all values of x ,

$$\lambda_Y(x) \equiv \sum_y P(y|x)\lambda(y).$$

2. Define λ values:

If $X \in \mathbf{A}$ and X 's value is \hat{x} ,

$$\begin{aligned}\lambda(\hat{x}) &\equiv 1 \\ \lambda(x) &\equiv 0 \quad \text{for } x \neq \hat{x}.\end{aligned}$$

If $X \notin \mathbf{A}$ and X is a leaf, for all values of x ,

$$\lambda(x) \equiv 1.$$

If $X \notin \mathbf{A}$ and X is a nonleaf, for all values of x ,

$$\lambda(x) \equiv \prod_{U \in \text{CH}_X} \lambda_U(x),$$

where CH_X is the set of all children of X .

Pearl's Message-Passing Algorithm (9)

3. Define π messages:

Let Z be a parent of X . Then for all values of z ,

$$\pi_X(z) \equiv \pi(z) \prod_{U \in \text{CH}_Z - \{X\}} \lambda_U(z).$$

4. Define π values:

If $X \in \mathbf{A}$ and X 's value is \hat{x} ,

$$\begin{aligned}\pi(\hat{x}) &\equiv 1 \\ \pi(\hat{x}) &\equiv 0 \quad \text{for } x \neq \hat{x}.\end{aligned}$$

If $X \notin \mathbf{A}$ and X is the root, for all values of x ,

$$\pi(x) \equiv P(x).$$

If $X \notin \mathbf{A}$, X is not the root, and Z is the parent of X , for all values of x ,

$$\pi(x) \equiv \sum_z P(x|z)\pi_X(z).$$

Pearl's Message-Passing Algorithm (10)

5. Given the definitions above, for each variable X , we have for all values of x ,

$$P(x|\mathbf{a}) = \alpha\lambda(x)\pi(x) ,$$

where α is a normalizing constant.

Implementation – Alternative 2 (1)

Routine *initial_tree* is first called as follows:

```
initial_tree( (G, P), A, a, P(x|a) );
```

After this call, **A** and **a** are both empty, and for every variables X , for every value of x , $P(x|a)$ is the conditional probability of x given **a**, which, since **a** is empty, is the prior probability of x . Each time a variable V is instantiated for \hat{v} , routine *update-tree* is called as follows:

```
update_tree( (G, P), A, a, V, \hat{v}, P(x|a) );
```

After this call, V has been added to **A**, \hat{v} has been added to **a**, and for every variables X , for every value of x , $P(x|a)$ has been updated to be the conditional probability of x given the new value of **a**.

Implementation – Alternative 1

The algorithm can be implemented as an object-oriented program, in which each node is an object that communicates with the other nodes by passing λ and π messages.

Make suggestions, now!

Algorithm 3.1 Inference-in-Trees

Problem: Given a Bayesian network whose DAG is a tree, determine the probabilities of the values of each node conditional on specified values of the nodes in some subset.

Inputs: Bayesian network (\mathbb{G}, P) whose DAG is a tree, where $\mathbf{G} = (\mathbb{V}, \mathbb{E})$, and a set of values \mathbf{a} of a subset $A \subseteq \mathbb{V}$.

Outputs: The Bayesian network (\mathbb{G}, P) updated according to the values in \mathbf{a} . The λ and π values and messages and $P(x|\mathbf{a})$ for each $X \in \mathbb{V}$ are considered part of the network.

Implementation – Alternative 2 (3)

```
void initial_tree (Bayesian-network& (G, P) where G = (V, E),
                   set-of-variables& A, set-of-variable-values& a)
{
    A = ∅; a = ∅;
    for (each X ∈ V) {
        for (each value x of X)
            λ(x) = 1;                                // Compute λ values.
        for (the parent Z of X)                      // Does nothing if X is the a root.
            for (each value z of Z)
                λX(z) = 1;                         // Compute λ messages.
    }
    for (each value r of the root R) {
        P(r|a) = P(r);                            // Compute P(r|a).
        π(r) = P(r);                            // Compute R's π values.
    }
    for (each child X of R)
        send_π_msg(R, X);
}
```

Implementation – Alternative 2 (4)

```
void update_tree (Bayesian-network& (G, P) where G = (V, E),
                  set-of-variables& A, set-of-variable-values& a,
                  variable V, variable-value  $\hat{v}$ )
{
    A = A ∪ {V}; a = a ∪ { $\hat{v}$ };                                // Add V to A.
     $\lambda(\hat{v}) = 1$ ;  $\pi(\hat{v}) = 1$ ;  $P(\hat{v}|a) = 1$ ;          // Instantiate V to  $\hat{v}$ .
    for (each value of  $v \neq \hat{v}$ ) {
         $\lambda(v) = 0$ ;  $\pi(v) = 0$ ;  $P(v|a) = 0$ ;
    }
    if (V is not the root && V's parent Z  $\notin$  A)
        send_lambda_msg(V, Z);
    for (each child X of V such that X  $\notin$  A)
        send_pi_msg(V, X);
}
```

Implementation – Alternative 2 (5)

```

void send_λ_msg(node Y, node X)    // For simplicity ( $\mathbb{G}, P$ ) is
{                                         // not shown as input.
    for (each value of  $x$ ) {
         $\lambda_Y(x) = \sum_y P(y|x)\lambda(y);$            // Y sends X a λ message.

         $\lambda(x) = \prod_{U \in \text{CH}_X} \lambda_U(x);$        // Compute X's λ values.

         $P(x|a) = \alpha\lambda(x)\pi(x);$                  // Compute  $P(x|a)$ .
    }

    normalize  $P(x|a)$ ;
    if (X is not the root and X's parent  $Z \notin A$ )
        send_λ_msg(X, Z);
    for (each child W of X such that  $W \neq Y$  and  $W \notin A$ )
        send_π_msg(X, W);
}

```

Implementation – Alternative 2 (6)

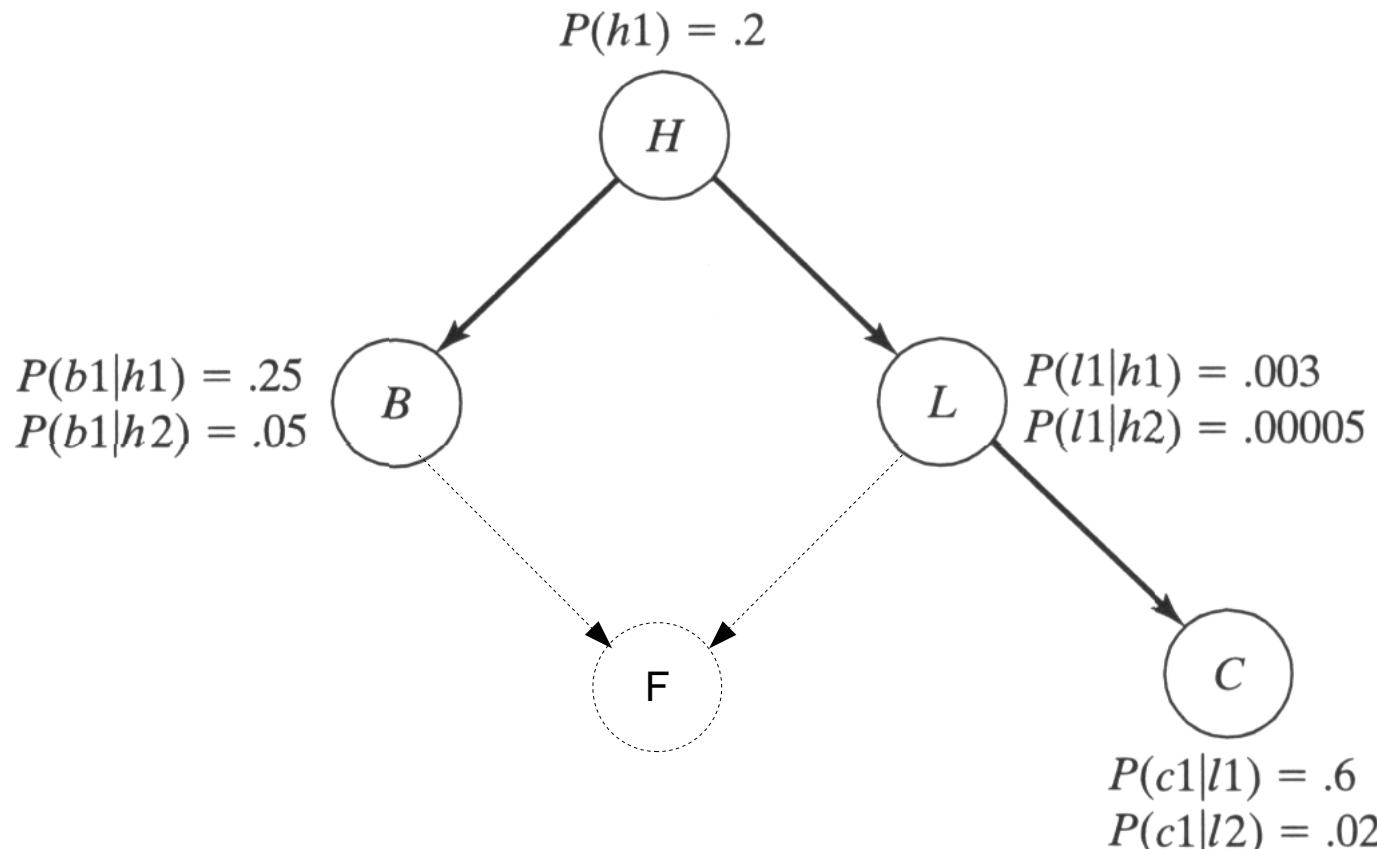
```
parent    child
void send_π_msg(node Z, node X)      // For simplicity ( $\mathbb{G}, P$ ) is
{                                         // not shown as input.
    for (each value of  $z$ )
         $\pi_X(z) = \pi(z) \prod_{Y \in \text{CH}_Z - \{X\}} \lambda_Y(z);$       // Z sends X a π message.

    for (each value of  $x$ ) {
         $\pi(x) = \sum_z P(x|z)\pi_X(z);$                                 // Compute X's π values.

         $P(x|\mathbf{a}) = \alpha \lambda(x)\pi(x);$                           // Compute  $P(x|\mathbf{a})$ .
    }

    normalize  $P(x|\mathbf{a});$ 
    for (each child  $Y$  of  $X$  such that  $Y \notin \mathbf{A}$ )
        send_π_msg(X, Y);
}
```

Example 3.3 – Rooted Tree



Task: Go through the first steps of the algorithm: `initial_tree()` and `update_tree()` for $B=b1$.

Example 3.3 – *initial_tree((G, P), A, a)* (1)

$A = \emptyset;$

$a = \emptyset;$

$\lambda(h1) = 1; \lambda(h2) = 1; \quad // Compute \lambda values.$

$\lambda(b1) = 1; \lambda(b2) = 1;$

$\lambda(l1) = 1; \lambda(l2) = 1;$

$\lambda(c1) = 1; \lambda(c2) = 1;$

$\lambda_B(h1) = 1; \lambda_B(h2) = 1; \quad // Compute \lambda messages.$

$\lambda_L(h1) = 1; \lambda_L(h2) = 1;$

$\lambda_C(l1) = 1; \lambda_C(l2) = 1;$

$P(h1|\emptyset) = P(h1) = .2; \quad // Compute P(h|\emptyset).$

$P(h2|\emptyset) = P(h2) = .8;$

$\pi(h1) = P(h1) = .2; \quad // Compute H's \pi values.$

$\pi(h2) = P(h2) = .8;$

send_ π *_msg(H, B);*

send_ π *_msg(H, L);*

Example 3.3 – *initial_tree((G, P), A, a)* (2)

The call

send_π_msg(H, B);

results in the following steps:

$$\begin{aligned}\pi_B(h1) &= \pi(h1)\lambda_L(h1) = (.2)(1) = .2; && // H \text{ sends } B \text{ a } \pi \text{ message.} \\ \pi_B(h2) &= \pi(h2)\lambda_L(h2) = (.8)(1) = .8;\end{aligned}$$

$$\begin{aligned}\pi(b1) &= P(b1|h1)\pi_B(h1) + P(b1|h2)\pi_B(h2); && // Compute B's \pi values. \\ &= (.25)(.2) + (.05)(.8) = .09;\end{aligned}$$

$$\begin{aligned}\pi(b2) &= P(b2|h1)\pi_B(h1) + P(b2|h2)\pi_B(h2); \\ &= (.75)(.2) + (.95)(.8) = .91;\end{aligned}$$

$$\begin{aligned}P(b1|\emptyset) &= \alpha\lambda(b1)\pi(b1) = \alpha(1)(.09) = .09\alpha; && // Compute P(b|\emptyset). \\ P(b2|\emptyset) &= \alpha\lambda(b2)\pi(b2) = \alpha(1)(.91) = .91\alpha;\end{aligned}$$

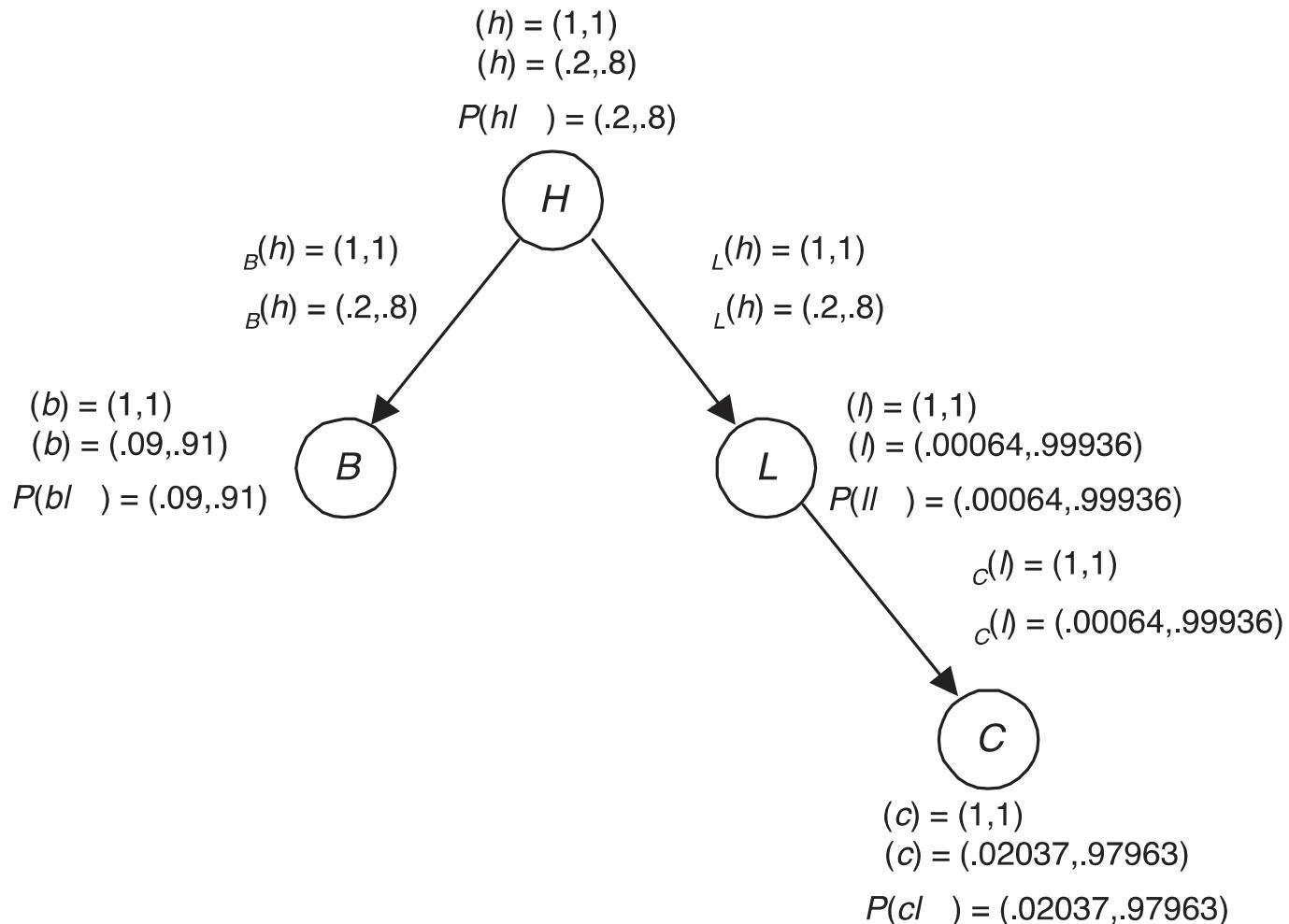
$$P(b1|\emptyset) = \frac{.09\alpha}{.09\alpha+.91\alpha} = .09;$$

$$P(b1|\emptyset) = \frac{.91\alpha}{.09\alpha+.91\alpha} = .91;$$

... and so on ...

b2

Example 3.3 – *initial_tree((G, P), A, a)* (3)



Example 3.3 – *update_tree((G, P), A, a, B, b1)* (1)

Example 3.4 Consider again the Bayesian network in Figure 3.7 (a). Suppose B is instantiated for $b1$. That is, we find out the patient has bronchitis. Next we show the steps in the algorithm when the network's values are updated according to this instantiation.

The call

update_tree((G, P), A, a, B, b1);

results in the following steps:

$$A = \emptyset \cup \{B\} = \{B\};$$

$$a = \emptyset \cup \{b1\} = \{b1\};$$

$$\lambda(b1) = 1; \pi(b1) = 1; P(b1|\{b1\}) = 1; \quad // Instantiate B for b1.$$

$$\lambda(b2) = 0; \pi(b2) = 0; P(b2|\{b1\}) = 0;$$

send_λ_msg(B, H);

Example 3.3 – *update_tree((G, P), A, a, B, b1)* (2)

The call

send_λ_msg(B, H);

results in the following steps:

$$\begin{aligned}\lambda_B(h1) &= P(b1|h1)\lambda(b1) + P(b2|h1)\lambda(b2); && // B \text{ sends } H \text{ a } \lambda \\ &= (.25)(1) + .75(0) = .25; && // \text{message.}\end{aligned}$$

$$\begin{aligned}\lambda_B(h2) &= P(b1|h2)\lambda(b1) + P(b2|h2)\lambda(b2); \\ &= (.05)(1) + .95(0) = .05;\end{aligned}$$

$$\begin{aligned}\lambda(h1) &= \lambda_B(h1)\lambda_L(h1) = (.25)(1) = .25; && // \text{Compute } H \text{'s } \lambda \\ \lambda(h2) &= \lambda_B(h2)\lambda_L(h2) = (.05)(1) = .05; && // \text{values.}\end{aligned}$$

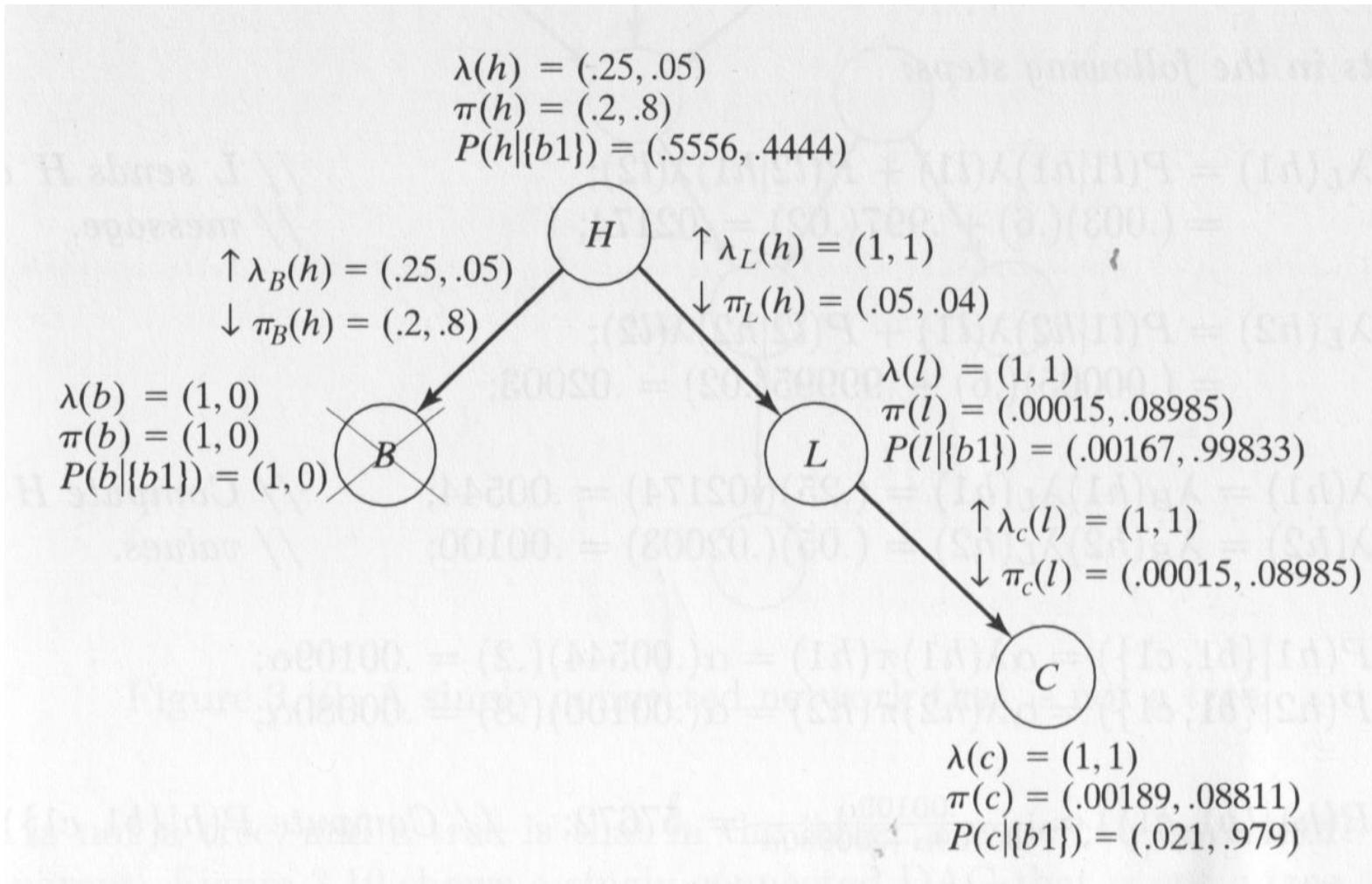
$$\begin{aligned}P(h1|\{b1\}) &= \alpha\lambda(h1)\pi(h1) = \alpha(.25)(.2) = .05\alpha; && // \text{Compute } P(h|\{b1\}). \\ P(h2|\{b1\}) &= \alpha\lambda(h2)\pi(h2) = \alpha(.05)(.8) = .04\alpha;\end{aligned}$$

$$P(h1|\{b1\}) = \frac{.05\alpha}{.05\alpha+.04\alpha} = .5556;$$

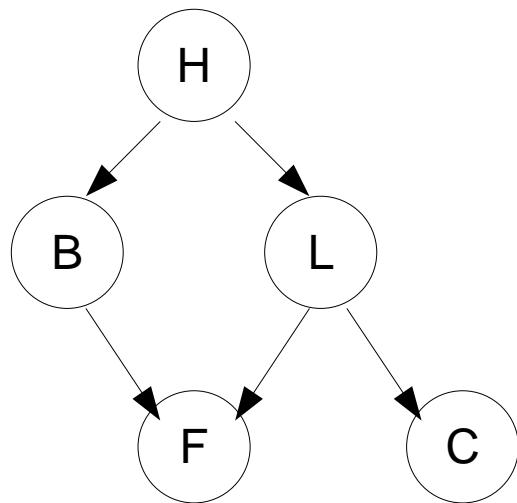
$$P(h2|\{b1\}) = \frac{.04\alpha}{.04\alpha+.05\alpha} = .4444;$$

send_π_msg(H, L);

Example 3.3 – *update_tree((G, P), A, a, B, b1)* (3)



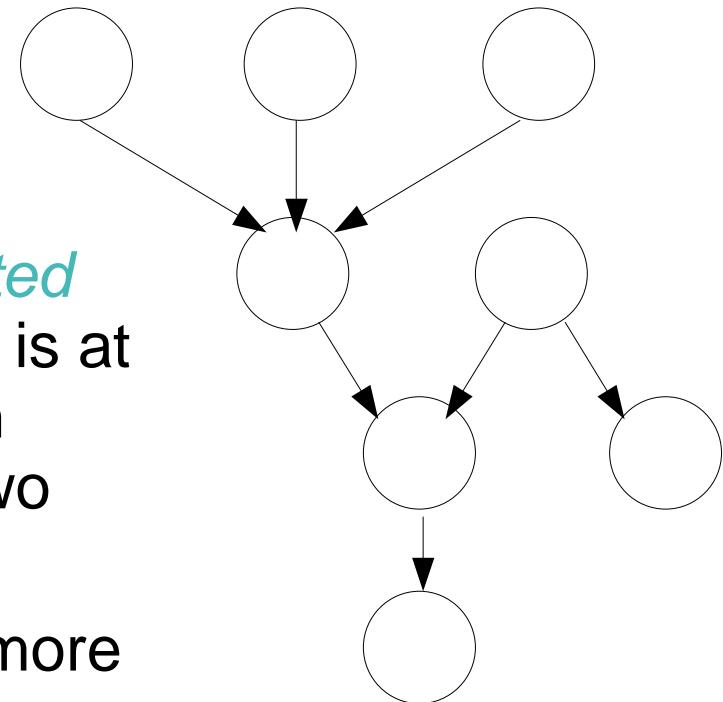
Inference in Singly Connected Networks



Multiply connected

- *Singly Connected DAG* := if there is at most one chain between any two nodes.

=> nodes can have more than parent!



Singly connected

We will workout the differences between the inference algorithm for trees and singly connected DAGs.

Inference in Singly Connected Networks – Algorithm (1)

Define λ messages:

5-2-3-4-1

For each child Y of X , for all values of x ,

Tree:

$$\lambda_Y(x) \equiv \sum_y P(y|x)\lambda(y).$$

Singly connected DAG:

$$\lambda_Y(x) \equiv \sum_y \left[\sum_{w_1, w_2, \dots, w_k} \left(P(y|x, w_1, w_2, \dots, w_k) \prod_{i=1}^k \pi_Y(w_i) \right) \right] \lambda(y).$$

where W_1, W_2, \dots, W_k are the other parents of Y .

(Message from child Y to parent X)

Tree & Singly connected DAG

Define λ values:

If $X \in \mathbf{A}$ and X 's value is \hat{x} ,

$$\begin{aligned}\lambda(\hat{x}) &\equiv 1 \\ \lambda(x) &\equiv 0 \quad \text{for } x \neq \hat{x}.\end{aligned}$$

If $X \notin \mathbf{A}$ and X is a leaf, for all values of x ,

$$\lambda(x) \equiv 1.$$

If $X \notin \mathbf{A}$ and X is a nonleaf, for all values of x ,

$$\lambda(x) \equiv \prod_{U \in \text{CH}_X} \lambda_U(x),$$

where CH_X is the set of all children of X .

Tree & Singly connected DAG

Define π messages:

Let Z be a parent of X . Then for all values of z ,

$$\pi_X(z) \equiv \pi(z) \prod_{U \in \text{CH}_Z - \{X\}} \lambda_U(z).$$

(Message from parent Z to a child X)

Tree & Singly connected DAG

Define π values:

If $X \in \mathbf{A}$ and X 's value is \hat{x} ,

$$\begin{aligned}\pi(\hat{x}) &\equiv 1 \\ \pi(x) &\equiv 0 \quad \text{for } x \neq \hat{x}.\end{aligned}$$

If $X \notin \mathbf{A}$ and X is the root, for all values of x ,

$$\pi(x) \equiv P(x).$$

Tree

If $X \notin A$, X is not the root, and Z is the parent of X , for all values of x ,

$$\pi(x) \equiv \sum_z P(x|z)\pi_X(z).$$

Singly connected DAG

If $X \notin A$, X is not a root, and Z_1, Z_2, \dots, Z_j are the parents of X , for all values of x ,

$$\pi(x) = \sum_{z_1, z_2, \dots, z_j} P(x|z_1, z_2, \dots, z_j) \prod_{i=1}^j \pi_X(z_i).$$

Tree & Singly connected DAG

Given the definitions above, for each variable X , we have for all values of x ,

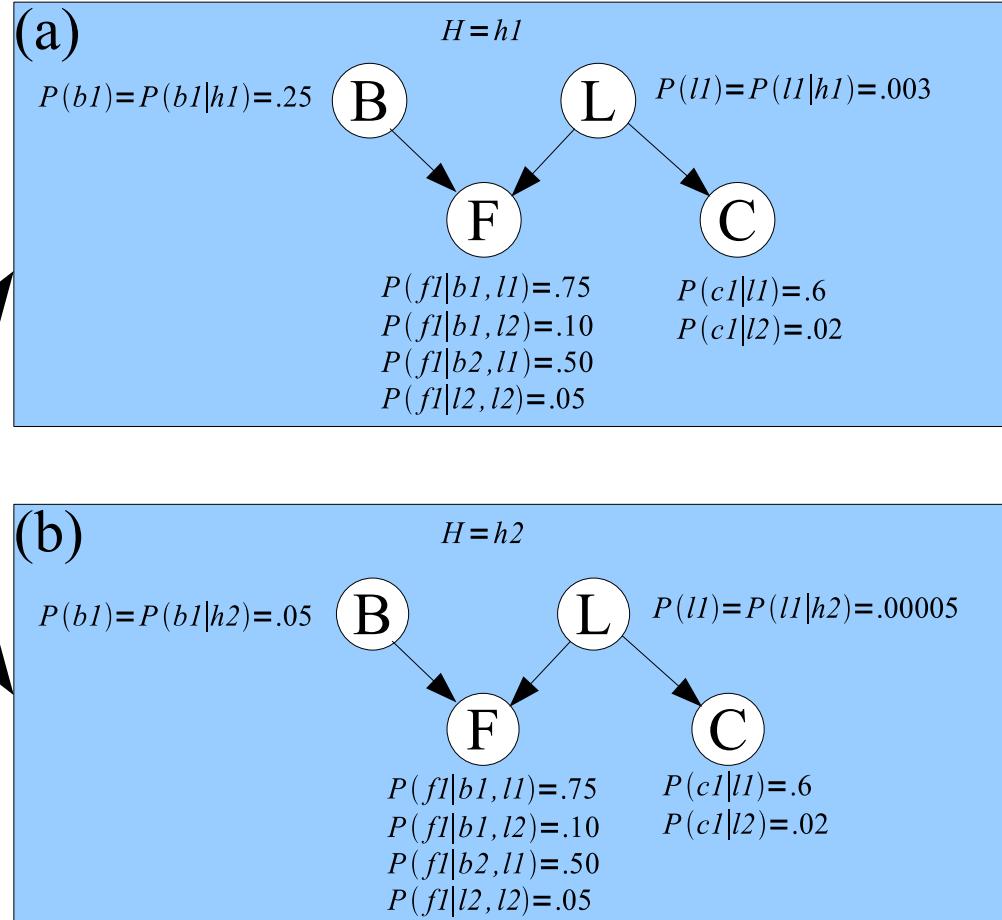
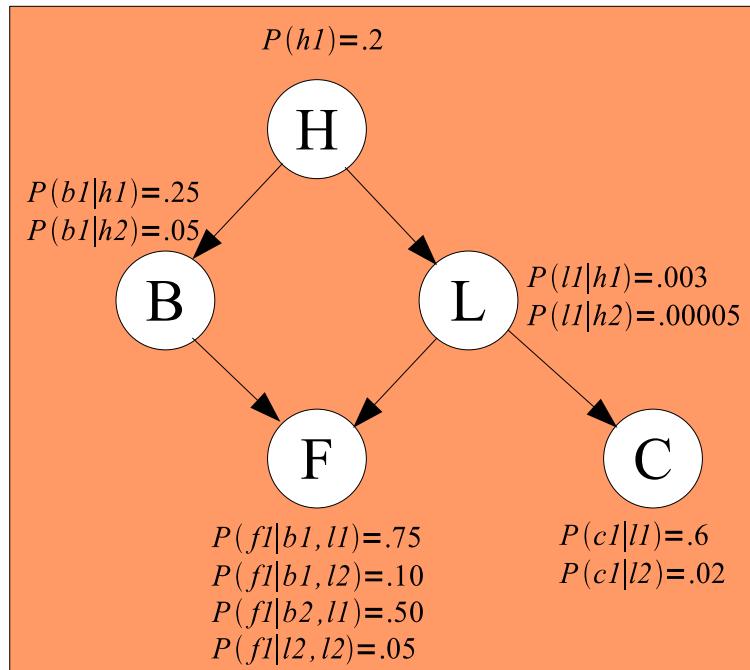
$$P(x|\mathbf{a}) = \alpha\lambda(x)\pi(x) ,$$

where α is a normalizing constant.

Conditioning (1)

Goal: Handling multiply connected BNs using the algorithm for singly connected BNs

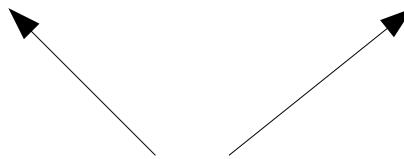
=> construct 2 BNs, one containing the CPD P' of P given $H=h1$ and another containing the CPD P'' of P given $H=h2$.



Conditioning (2)

- Suppose F is instantiated to $f1$.
- Suppose we are interested in $P(b1|f1)$.

$$\Rightarrow P(b1|f1) = \sum_{h=h1}^{h2} P(b1, h|f1) \\ = P(b1|h1, f1)P(h1|f1) + P(b1|h2, f1)P(h2|f1)$$



Loop-cutset := nodes on which we condition to break loops

Can be determined via Pearl's Algorithm from (a) and (b), respectively

$$P(h1|f1) = \alpha [P(f1|h1)P(h1)]$$
$$P(h2|f1) = \alpha [P(f1|h2)P(h2)]$$

Directly from original DAG (H is root)

Can be determined via Pearl's Algorithm from (a) and (b), respectively

Conditioning Algorithm

- Determine loop-cutset C
 - Heuristics are used (e.g., use root nodes if possible) because
 - Determining minimal loop-cutset is NP-hard.
- Let E be a set of instantiated nodes with instantiations $e = \{e_1, \dots, e_k\}$ (Evidence)
- For each X in $V - \{E \cup C\}$ we have

$$P(x_i|e) = \sum_c P(x_i, c|e) = \sum_c P(x_i|c, e) P(c|e)$$

Bayes rule

$$P(c|e) = \alpha P(e|c) P(c) \quad \text{where} \quad \alpha = \frac{1}{P(e)}$$

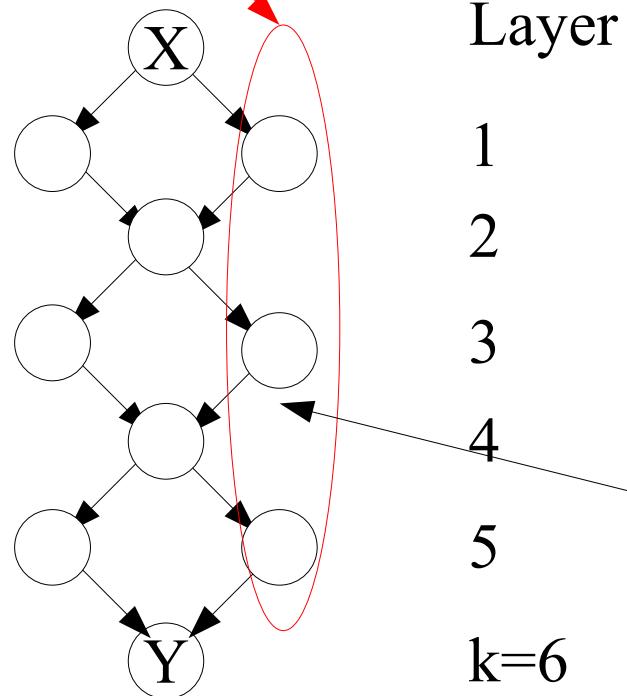
$$P(e|c) = P(e_k|e_{k-1}, e_{k-2}, \dots, e_1, c) P(e_{k-1}|e_{k-2}, \dots, e_1, c) \dots P(e_1|c) \quad \text{Chain rule}$$

Can be determined via Pearl's Algorithm using the created singly connected DAGs

Readily available if all nodes in C are root nodes;
otherwise special algorithm needed (see Suermond & Cooper)

Problematic Case Example

Conditioning



The problem of inference in BNs has shown to be NP complete

Assumption: Binary nodes

Noisy OR-Gate Model – Motivation

- A BN requires the CPs (conditional probabilities) of each variable given all combinations of values of its parents.
- Assume only binary nodes are considered:
 - For each node with p parents: 2^p CPs are needed.
 - For large p , storage and computational requirements for inference become infeasible.
 - Often the CPs based on several parent values are ordinarily not very accessible.
 - **Example:**
 - It is much easier to ascertain $P(F|L)$ and $P(F|B)$ than to ascertain $P(F|L,B)$.

Noisy OR-Gate Model – Assumptions

Causal inhibition

- There is some mechanism that inhibits its cause from bringing about its effect and
- The presence of the cause results in the presence of the effect iff this mechanism is disabled (turned off).

Exception independence

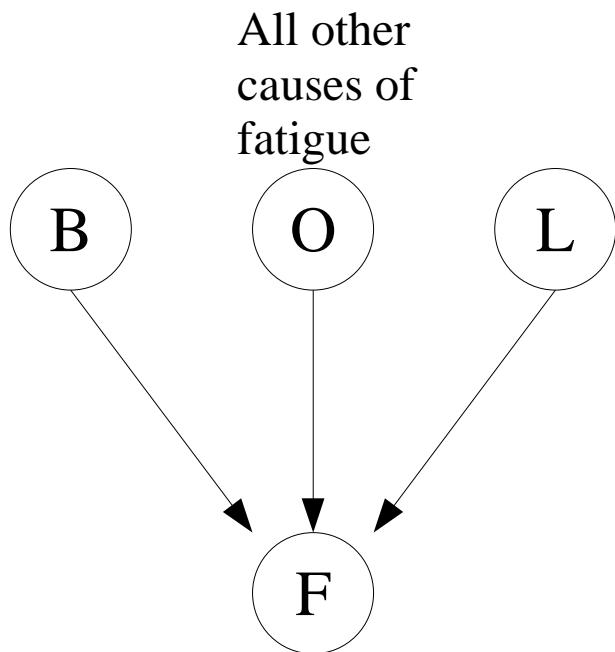
- Mechanism inhibiting a cause is independent of the mechanism that inhibits another cause.

Accountability

- Effects can happen only if at least one of its causes is present and is not being inhibited.
⇒ All causes not explicitly stated must be lumped into one unknown cause (Node “0” for “Other causes”).

Binary Variables

- Value “1” = feature present
- Value “2” = feature absent



Causal inhibition

- B will only result in F iff the mechanism that inhibits this from happening is not present

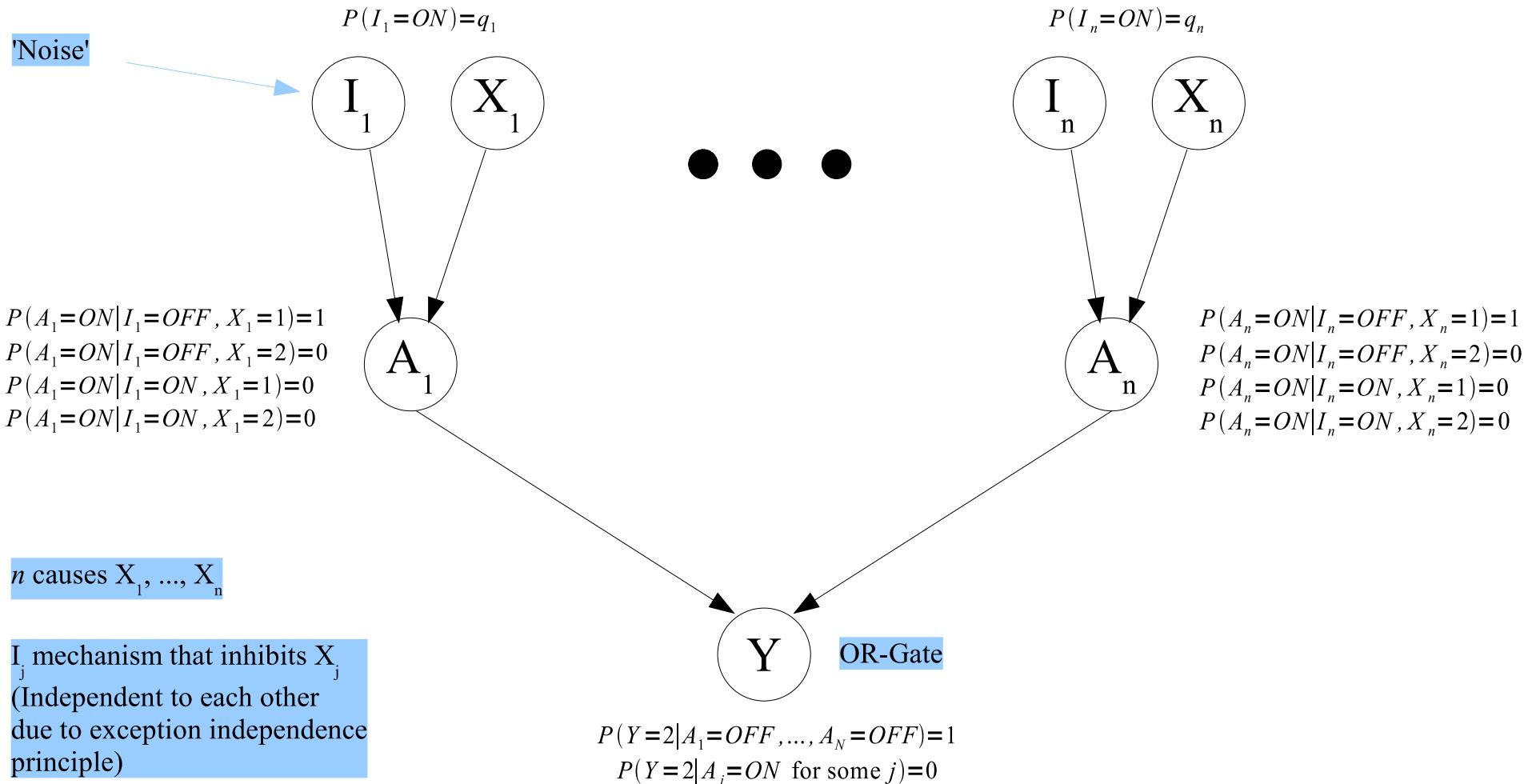
Exception independence

- Mechanism inhibiting B resulting in F is independent of the mechanism that inhibits L or O .

Accountability

- F cannot be present unless at least one of B , L , or O is present

Noisy OR-Gate Model



Efficient Problem Specification

Theorem 3.3: Suppose we have a Bayesian network representing the Noisy OR-gate model (see previous slide). Let

$$W = \{X_1, \dots, X_n\},$$

and let

$$w = \{x_1, \dots, x_n\}$$

be a set of values of the variables in W . Furthermore, let S be a set of indices such that $j \in S$ iff $X_j = 1$. That is,

$$S = \{j \text{ such that } X_j = 1\}.$$

Then

$$P(Y = 2 | W = w) = \prod_{j \in S} q_j.$$

This is what we
determine in practice

The value $p_j = 1 - q_j$ is called the **causal strength** of X_j for Y .

Theorem 3.3 implies: $p_j = P(Y = 1 | X_j = 1, X_i = 2 \forall i \neq j)$

⇒ Value is relatively accessible: E.g., DB of patients with (B & no other disease) or (L & no other disease)

Efficient Inference (1)

Assume the variables satisfy the noisy OR-Gate model, Y has n parents X_1, \dots, X_n . Let p_j be the causal strength from X_j to Y , $q_j = 1 - p_j$.

x_j^+ denotes that X_j is present, and x_j^- that X_j is not present, then

$$\begin{aligned}\lambda_Y(x_j^+) &= \lambda(y^-)q_jP_j + \lambda(y^+)(1 - q_jP_j) \\ \lambda_Y(x_j^-) &= \lambda(y^-)P_j + \lambda(y^+)(1 - P_j)\end{aligned}$$

where

$$P_j = \prod_{i \neq j} [1 - p_i \pi_Y(x_i^+)]$$

instead of

$$\lambda_Y(x_j) = \sum_y \left[\sum_{x_1, \dots, x_{j-1}, x_{j+1}, \dots, x_n} \left(P(y|x_1, \dots, x_n) \prod_{i \neq j} \pi_Y(x_i) \right) \right] \lambda(y)$$

and

Efficient Inference (2)

$$\pi(y^+) = 1 - \prod_{j=1}^n [1 - p_j \pi_Y(x_j^+)]$$
$$\pi(y^-) = \prod_{j=1}^n [1 - p_j \pi_Y(x_j^+)] = 1 - \pi(y^+)$$

instead of

$$\pi(y) = \sum_{x_1, \dots, x_n} \left(P(y|x_1, \dots, x_n) \prod_{j=1}^n \pi_Y(x_j) \right)$$

Example: Object Detection

Object detection by part detection

- Each of the parts indicated a face (left eye, right eye, nose, mouth, etc.)
- Each of the parts can be blocked individually and doesn't influence the other features (e.g., by spatial occlusion)
- All parts of a face must be captured (accountability)

SS 2014 – Bayesian Networks

Continuous Variable Inference

University of Augsburg

Multimedia Computing and Computer Vision,

Prof. Dr. Rainer Lienhart

Rainer.Lienhart@informatik.uni-augsburg.de

www.multimedia-computing.org

Reference

Richard E. Neapolitan. **Learning Bayesian Networks.** *Prentice Hall Series in Artificial Intelligence*, ISBN 0-13-012534-2.

Don't forget. Reading the book chapters 1 – 6 is mandatory.

Chapter on ***More Inference Algorithm***
(chapter 4)

Figures and text are taken from that book

The Normal Distribution (1)

Definition 4.1: The **normal density function with parameters μ and σ** , where $-\infty < \mu < \infty$ and $\sigma > 0$, is

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \quad -\infty < x < \infty$$

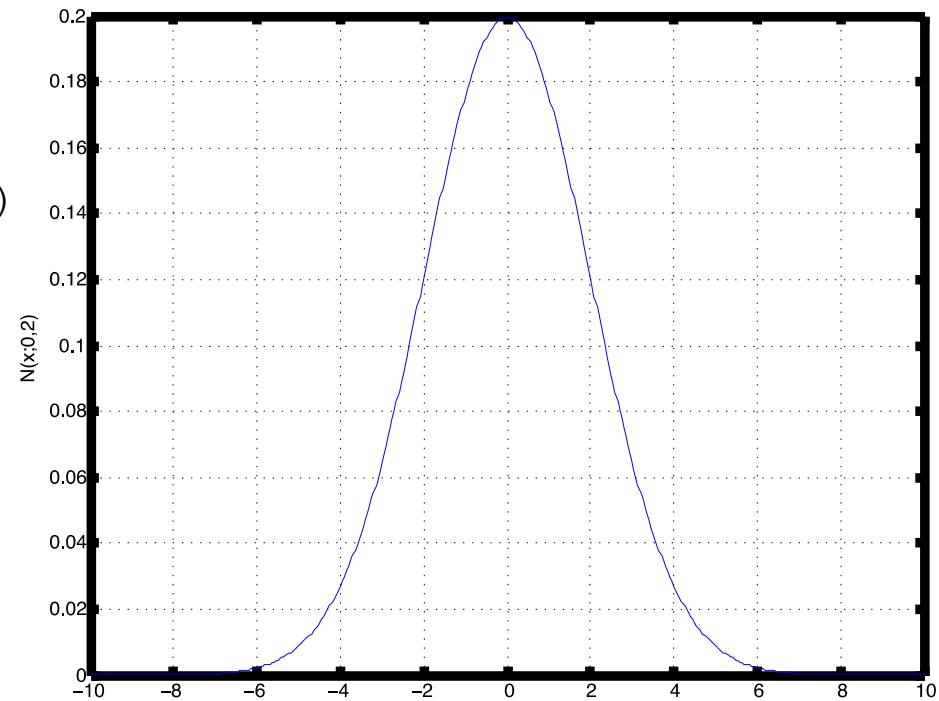
and is denoted $N(x; \mu, \sigma^2)$. A random variable X that has this density function is said to have a **normal distribution** with $E(X) = \mu$, $Var(X) = \sigma^2$.

Standard normal density function: $N(x; 0, 1)$

Octave/Matlab Script

```
function [ret] = myGaussian (x, my, sigma)
    ret = exp(-(x-my).* (x-my) / (2*
sigma*sigma)) / (sqrt(2*pi)*sigma);
end
```

```
x = -10:.05:10;
y = myGaussian(x,0,2);
plot(x,y); grid on
xlabel('x'); ylabel('N(x;0,2)')
```

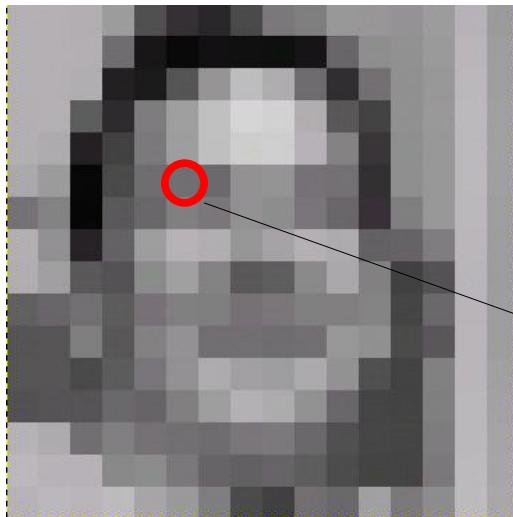


Motivation

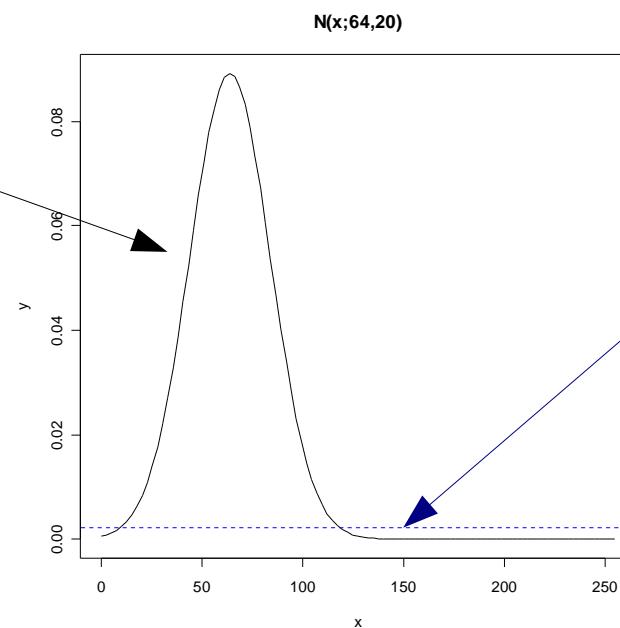
Why continuous random variables are needed?

- Example from image processing domain

Example of Object



Example of Non-Object



The Normal Distribution (2)

Theorem 4.1: The following equalities hold for the normal density function:

$$\begin{aligned}N(x; \mu, \sigma^2) &= N(\mu; x, \sigma^2) \\N(ax; \mu, \sigma^2) &= \frac{1}{a} N\left(x; \frac{\mu}{a}, \frac{\sigma^2}{a^2}\right) \\N(x; \mu_1, \sigma_1^2)N(x; \mu_2, \sigma_2^2) &= kN\left(x; \frac{\sigma_2^2 \mu_1 + \sigma_1^2 \mu_2}{\sigma_1^2 + \sigma_2^2}, \frac{\sigma_1^2 \sigma_2^2}{\sigma_1^2 + \sigma_2^2}\right) \\&= kN(x; \mu, \sigma^2)\end{aligned}$$

with k independent of x and

$$\begin{aligned}\sigma^2 &= \left[\sum_{i=1}^2 \frac{1}{\sigma_i^2} \right]^{-1} = \left[\frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2} \right]^{-1} = \left[\frac{\sigma_2^2 + \sigma_1^2}{\sigma_1^2 \sigma_2^2} \right]^{-1} = \frac{\sigma_1^2 \sigma_2^2}{\sigma_2^2 + \sigma_1^2} \\\mu &= \sigma^2 \sum_{i=1}^2 \frac{\mu_i}{\sigma_i^2} = \sigma^2 \left(\frac{\mu_1}{\sigma_1^2} + \frac{\mu_2}{\sigma_2^2} \right) = \sigma^2 \left(\frac{\mu_1 \sigma_2^2 + \mu_2 \sigma_1^2}{\sigma_1^2 \sigma_2^2} \right) = \frac{\mu_1 \sigma_2^2 + \mu_2 \sigma_1^2}{\sigma_2^2 + \sigma_1^2} \\\int_x N(x; \mu_1, \sigma_1^2)N(x; \mu_2, \sigma_2^2)dx &= N(y; \mu_1, \sigma_1^2 + \sigma_2^2)\end{aligned}$$

Example 4.1 and Example 4.2

See book, pages 185 to 187.

Gaussian Bayesian Network

Gaussian Bayesian Network = The value of each variable X is a linear function of the values of its parents

$$x = w_X + \sum_{Z \in PA_X} b_{XZ} z$$

with

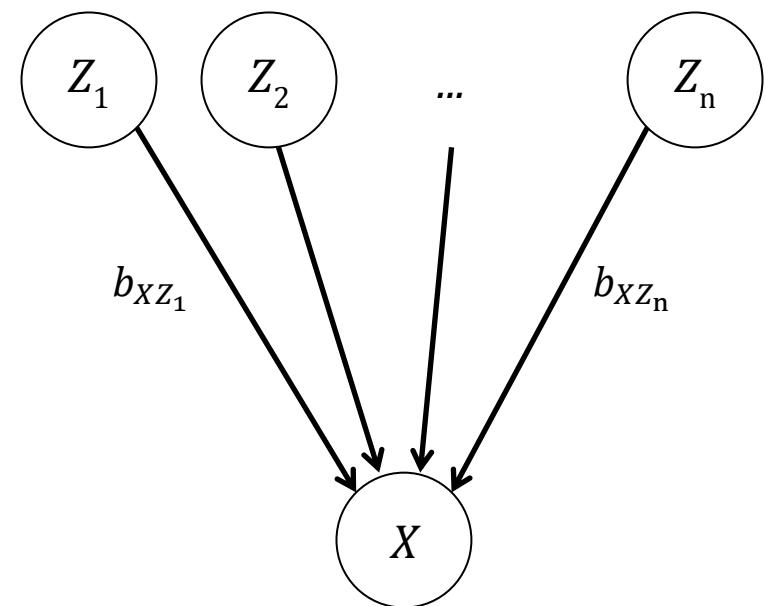
PA_X = set of parents of X .

W_X = uncertainty in X 's value given values of X 's parents with density $N(w_X; 0, \sigma_{W_X}^2)$; independent of each $Z \in PA_X$.

In other words, the value of each variable X is a weighted sum of its parents' values plus additive white noise.

Thus, the conditional density function of node X is

$$p(x|pa_X) = N(x; \sum_{Z \in PA_X} b_{XZ} z, \sigma_{W_X}^2)$$



For each root X , we specify its density function $N(x; \mu_X, \sigma_X^2)$ using either of the following both options:

- $N(x; \mu_X, 0)$ if the value is known (complete certainty)
- $N(x; 0, \infty)$ if the value is unknown (complete uncertainty)

Inference in singly connected networks.

IMPORTANT: To avoid clutter, σ is used to represent the variance, and not the standard deviation!

Discrete:

Define λ messages:

For each child Y of X , for all values of x ,

$$\lambda_Y(x) \equiv \sum_y \left[\sum_{w_1, w_2, \dots, w_k} \left(P(y|x, w_1, w_2, \dots, w_k) \prod_{i=1}^k \pi_Y(w_i) \right) \right] \lambda(y).$$

where W_1, W_2, \dots, W_k are the other parents of Y .

Gaussian:

Define λ messages:

For each child Y of X , for all values of x ,

$$\begin{aligned} \sigma_{YX}^\lambda &= \frac{1}{b_{YX}^2} \left[\sigma_Y^\lambda + \sigma_{W_Y} + \sum_{Z \in \text{PA}_Y - \{X\}} b_{YZ}^2 \sigma_{YZ}^\pi \right] \\ \mu_{YX}^\lambda &= \frac{1}{b_{YX}} \left[\mu_Y^\lambda - \sum_{Z \in \text{PA}_Y - \{X\}} b_{YZ} \mu_{YZ}^\pi \right] \end{aligned}$$

GBN - Algorithm (2)

Discrete

Define λ values:

If $X \in \mathbf{A}$ and X 's value is \hat{x} ,

$$\lambda(\hat{x}) \equiv 1$$

$$\lambda(x) \equiv 0 \quad \text{for } x \neq \hat{x}.$$

If $X \notin \mathbf{A}$ and X is a leaf, for all values of x ,

$$\lambda(x) \equiv 1.$$

If $X \notin \mathbf{A}$ and X is a nonleaf, for all values of x ,

$$\lambda(x) \equiv \prod_{U \in \text{CH}_X} \lambda_U(x),$$

where CH_X is the set of all children of X .

Gaussian

If $X \in \mathbf{A}$ and X 's value is \hat{x} ,

$$\sigma_X^\lambda = \sigma_X = 0$$

$$\mu_X^\lambda = \mu_X = \hat{x}$$

If $X \notin \mathbf{A}$ and X is a leaf, for all values of x ,

$$N(x; 0, \infty)$$

If $X \notin \mathbf{A}$ and X is a nonleaf, for all values of x ,

$$\sigma_X^\lambda = \left[\sum_{U \in \text{CH}_X} \frac{1}{\sigma_{UX}^\lambda} \right]^{-1}$$
$$\mu_X^\lambda = \sigma_X^\lambda \sum_{U \in \text{CH}_X} \frac{\mu_{UX}^\lambda}{\sigma_{UX}^\lambda}$$

Discrete

Define π messages:

Let Z be a parent of X . Then for all values of z ,

$$\pi_X(z) \equiv \pi(z) \prod_{U \in \text{CH}_Z - \{X\}} \lambda_U(z).$$

Gaussian

Define π messages:

Let Z be a parent of X . Then for all values of z ,

$$\sigma_X^\lambda = \left[\frac{1}{\sigma_Z^\pi} + \sum_{U \in \text{CH}_Z - \{X\}} \frac{1}{\sigma_{UZ}^\lambda} \right]^{-1}$$

$$\mu_X^\lambda = \sigma_X^\lambda \left[\frac{\mu_Z^\pi}{\sigma_Z^\pi} + \sum_{U \in \text{CH}_Z - \{X\}} \frac{\mu_{UZ}^\lambda}{\sigma_{UZ}^\lambda} \right]$$

GBN - Algorithm (4)

Discrete

Define π values:

If $X \in \mathbf{A}$ and X 's value is \hat{x} ,

$$\pi(\hat{x}) \equiv 1$$

$$\pi(x) \equiv 0 \text{ for } x \neq \hat{x}.$$

If $X \notin \mathbf{A}$ and X is the root, for all values of x ,

$$\pi(x) \equiv P(x)$$

If $X \notin \mathbf{A}$, X is not a root, and Z_1, Z_2, \dots, Z_j are the parents of X , for all values of x ,

$$\pi(x) = \sum_{z_1, z_2, \dots, z_j} P(x|z_1, z_2, \dots, z_j) \prod_{i=1}^j \pi_{Z_i}(z_i).$$

Gaussian

If $X \in \mathbf{A}$ and X 's value is \hat{x} ,

$$\sigma_X^\pi = \sigma_x = 0$$

$$\mu_X^\pi = \mu_x = \hat{x}$$

If $X \notin \mathbf{A}$ and X is a root, for all values of x ,

$$N(x; \mu_X, \sigma_X)$$

If $X \notin \mathbf{A}$ and X is not a root, for all values of x ,

$$\sigma_X^\pi = \sigma_{W_X} + \sum_{Z \in \text{PA}_X} b_{XZ}^2 \sigma_{XZ}^\pi$$

$$\mu_X^\pi = \sum_{Z \in \text{PA}_X} b_{XZ} \mu_{XZ}^\pi$$

Discrete

Given the definitions above, for each variable X , we have for all values of x ,

$$P(x|\mathbf{a}) = \alpha\lambda(x)\pi(x),$$

where α is a normalizing constant.

Gaussian

The variance and expectation for X are as follows:

$$\sigma_X = \frac{\sigma_X^\pi \sigma_X^\lambda}{\sigma_X^\pi + \sigma_X^\lambda}$$

$$\mu_X = \frac{\sigma_X^\pi \mu_X^\lambda + \sigma_X^\lambda \mu_X^\pi}{\sigma_X^\pi + \sigma_X^\lambda}$$

Algorithm 4.1

Algorithm 4.1 im Vergleich zu 3.2?

GBN – Calculating with Infinite Variances

The calculations with ∞ are done by taking limits, and every specified infinity represents the same variable approaching ∞ .

Example:

If $\sigma_P^\pi = \infty, \mu_P^\pi = 0, \sigma_P^\lambda = \infty, \mu_P^\lambda = 8000$, then

$$\frac{\sigma_P^\pi \mu_P^\lambda + \sigma_P^\lambda \mu_P^\pi}{\sigma_P^\pi + \sigma_P^\lambda} = \lim_{t \rightarrow \infty} \frac{t \times 8000 + t \times 0}{t + t} = \lim_{t \rightarrow \infty} \frac{t}{t} \times \frac{8000 + 0}{2} = \lim_{t \rightarrow \infty} \frac{8000}{2} = 4000$$

Note, since λ and π messages/values are used in other computations, we assign variables values that are multiples of infinity when it is indicated.

Example:

If

$$\sigma_{DP}^\lambda = 0 + 300^2 + \infty + \infty = 2\infty$$

we would use $2t$ in an expression containing σ_{DP}^λ .

SS 2014 – Bayesian Networks

Approximate Inference

*University of Augsburg
Multimedia Computing and Computer Vision,
Prof. Dr. Rainer Lienhart
Rainer.Lienhart@informatik.uni-augsburg.de
www.multimedia-computing.org*

Reference

Richard E. Neapolitan. **Learning Bayesian Networks.** *Prentice Hall Series in Artificial Intelligence*, ISBN 0-13-012534-2.

Don't forget. Reading the book chapters 1 – 6 is mandatory.

Chapter on ***More Inference Algorithm***
(chapter 4)

Figures and text are taken from that book

Motivation

Inference in BNs is NP-hard.

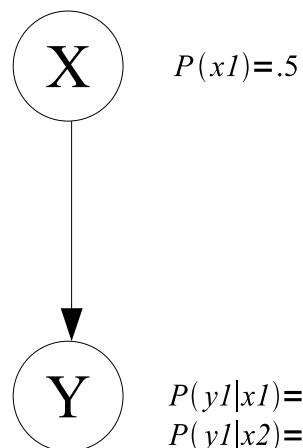
- ⇒ For some networks none of our exact inference algorithms will be efficient
- ⇒ Use approximation

Approach: ***Stochastic Simulation***

- ⇒ Randomly create data items according to probability distribution P in DAG
- ⇒ Approximate CPD of interest using these samples

Logic Sampling - Example (1)

Problem:



$$\hat{P}(y1) = ?$$

Sampling Data:

Case	X	Y
1	X2	Y1
2	X1	Y1
3	X1	Y2
4	X2	Y1
5	X1	Y2
6	X2	Y1
7	X2	Y2

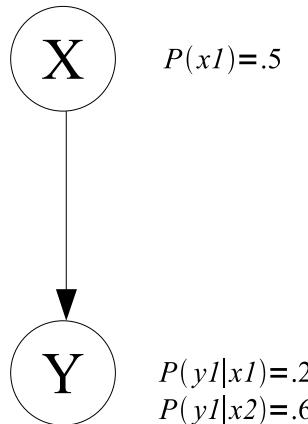
$$\hat{P}(y1) = \frac{k}{m} = \frac{4}{7}$$

Sampling Algorithm:

```
k=0;  
for (i=0 ; i<m ; i++) {  
    generate a value x of X  
    using P(x)  
    generate a value y of Y  
    using P(y|x)  
    if (y == y1)  
        k++  
}  
hat P(y1) = k/m
```

Logic Sampling - Example (2)

Problem:



Sampling Data:

Case	X	Y
1	X2	Y1
2	X1	Y1
3	X1	Y2
4	X2	Y1
5	X1	Y2
6	X2	Y1
7	X2	Y2

Sampling Algorithm:

```

k=0;
for (i=0 ; i<m ; i++) {
  repeat
    generate a value x of X
    using P(x)
    generate a value y of Y
    using P(y|x)
  until (y == y1);
  if (x == x1)
    k++
}
hat P(x1|y1) = k/m
  
```

$$\hat{P}(x1|y1) = ?$$

$$\hat{P}(x1|y1) = \frac{k}{m} = \frac{1}{4}$$

Why regenerate a new value for X each time a y2 is generated for Y?
Because we would otherwise just get P(x)

Logic Sampling - Algorithm (1)

Ancestral Ordering := an ordering of the nodes such that, if Z is a descendant of Y , then Z follows Y in the ordering.

Algorithm 4.2 Approximate Inference Using Logic Sampling

Problem: Given a Bayesian network, determine the probabilities of the values of each node conditional on specified values of the nodes in some subset.

Inputs: Bayesian network (\mathbb{G}, P) , where $\mathbb{G} = (\mathbb{V}, \mathbb{E})$, and a set of values a of a subset $A \subseteq \mathbb{V}$.

Outputs: Estimates of the conditional probabilities of the values of each node in $\mathbb{V} - A$ given $A = a$.

Logic Sampling - Algorithm (2)

```
void logic_sampling (Bayesian-network& (G, P) where G = (V, E),
                    set-of-variables A,
                    set-of-variable-values a,
                    estimates& P(x_j|a))
{
    order the n nodes in V in an ancestral ordering;
    for (each X_j ∈ V - A)
        for (k = 1; k <= # of values in X_j's space; k++)           // x_{jk} is
            set # of occurrences of x_{jk} to 0;                      // the kth
        for (i = 1; i <= m; i++) {                                     // value in
            j = 1;                                                       // X_j's space.
            while (j <= n) {
                generate a value x̃_j for X_j using
                P(x_j|p̃a_j) where p̃a_j is the
                set of values generated for X_j's parents;
                if (X_j ∈ A && x̃_j ≠ the value of X_j ∈ a)
                    j = 1;
                else
                    j++;
            }
        }
}
```

Logic Sampling - Algorithm (3)

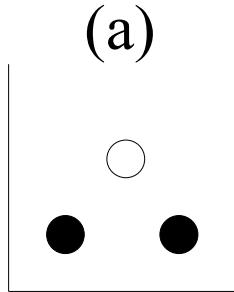
```
for (each  $X_j \in V - A$ )
    for ( $k = 1; k \leq \#$  of values in  $X_j$ 's space;  $k++$ )
        if ( $x_{jk} == \tilde{x}_j$ )
            add 1 to # of occurrences of  $x_{jk}$ ;
    }
for (each  $X_j \in V - A$ )
    for ( $k = 1; k \leq \#$  of values in  $X_j$ 's space;  $k++$ )
         $\hat{P}(x_{jk}|a) = \frac{\# \text{ of occurrences of } x_{jk}}{m};$ 
}
```

Likelihood Weighting (1)

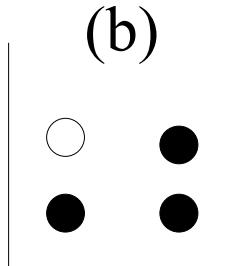
- Logic sampling rejects many randomly generated cases that do not have the evidence variables **A** instantiated to **a**.
 - If the probability of evidence is low, we will reject many cases: $1 - P(\mathbf{a})$.
- ⇒ **Likelihood Weighting** circumvents this problem.

Likelihood Weighting (2)

That's what we want to estimate



$$P(\text{black}) = 2/3$$



$$P'(\text{black}) = 3/4$$

Define

$$\text{score}(\text{black}) = \frac{P(\text{black})}{P'(\text{black})}$$

Now use (b) to estimate (a):

$$\begin{aligned}\lim_{m \rightarrow \infty} \frac{k \times \text{score}(\text{black})}{m} &= \lim_{m \rightarrow \infty} \frac{k \times \frac{P(\text{black})}{P'(\text{black})}}{m} \\ &= \frac{P(\text{black})}{P'(\text{black})} \lim_{m \rightarrow \infty} \frac{k}{m} \\ &= \frac{P(\text{black})}{P'(\text{black})} P'(\text{black}) = P(\text{black})\end{aligned}$$

using this distribution

Likelihood Weighting (3)

Note: For any finite sample, we would not necessarily have

$$\frac{k \times \text{score}(\text{black})}{m} + \frac{(m - k) \times \text{score}(\text{white})}{m} = 1.$$

We can expect this only in the case of the limit. Thus for any sample we need to normalize.

Example:

See example 4.14 on page 213.

Let (G, P) with $G = (V, E)$ be a BN, $V = \{X_1, X_2, \dots, X_n\}$, $A \subseteq V$, $W = V - A$, a and w be sets of values of the variables A , respectively and W and $v = w \cup a$. Then

$$\begin{aligned} P(w|a) &= \frac{P(w, a)}{P(a)} = \frac{P(v)}{P(a)} \\ &= \alpha P(v) \\ &= \alpha P(x_n|pa_n)P(x_{n-1}|pa_{n-1}) \dots P(x_2|pa_2)P(x_1|pa_1) \end{aligned}$$

with α being a normalization constant.

Likelihood Weighting (4)

We want to create samples according to $P(w|a) = \alpha P(x_n|pa_n)P(x_{n-1}|pa_{n-1})\dots P(x_2|pa_2)P(x_1|pa_1)$.

Define $P'(w) = \prod_{X_i \in W} P(x_i|pa_i)$ and

$$\begin{aligned} score(w) &= \frac{P(w|a)}{P'(w)} \\ &= \frac{\alpha P(x_n|pa_n)P(x_{n-1}|pa_{n-1})\dots P(x_2|pa_2)P(x_1|pa_1)}{\prod_{X_i \in W} P(x_i|pa_i)} \\ &= \alpha \prod_{X_i \in A} P(x_i|pa_i) \end{aligned}$$

Two requirements:

1. Must be able to create samples according to P'
2. Must be able to calculate $score = P/P'$

Thus, if we estimate $\hat{P}(x_{jk}|a)$ based on distribution $P'(w)$, then with knowing $score(w)$ we can derive $P(x_{jk}|a)$ up to a constant factor.

Thus by normalizing all probability estimates from the scores, we get the finally correct estimated for $P(x_{jk}|a)$.

Likelihood Weighting – Algorithm (1)

Algorithm 4.3 Approximate Inference Using Likelihood Weighting

Problem: Given a Bayesian network, determine the probabilities of the values of each node conditional on specified values of the nodes in some subset.

Inputs: Bayesian network (\mathbb{G}, P) , where $\mathbb{G} = (\mathcal{V}, \mathcal{E})$, and a set of values a of a subset $A \subseteq \mathcal{V}$.

Outputs: Estimates of the conditional probabilities of the values of each node in $\mathcal{V} - A$ given $A = a$.

```
void like_weight (Bayesian-network& ( $\mathbb{G}, P$ ) where  $\mathbb{G} = (\mathcal{V}, \mathcal{E})$ ,  
set-of-variables A,  
set-of-variable-values a,  
estimates&  $\hat{P}(x_j|a)$ )
```

Likelihood Weighting – Algorithm (2)

```
{  
    order the  $n$  nodes in  $V$  in an ancestral order;  
    for (each  $X_j \in V - A$ )  
        for ( $k = 1; k \leq \#$  of values in  $X_j$ 's space;  $k++$ ) //  $x_{jk}$  is  
             $\hat{P}(x_{jk} | a) = 0;$  // the  $k$ th  
        for (each  $X_j \in A$ ) // value in  
            set  $\tilde{x}_j$  to the value of  $X_j$  in  $a;$  //  $X_j$ 's space.  
        for ( $i = 1; i \leq m; i++$ ) {  
            for ( $j = 1; j \leq n; j++$ ) {  
                if ( $X_j \notin A$ )  
                    generate a value  $\tilde{x}_j$  for  $X_j$  using // Use all  
                     $P(x_{jk} | \tilde{pa}_j)$  where  $\tilde{pa}_j$  is the // values of  $k.$   
                    set of values generated for  $X_j$ 's parents;  
            }  
        }
```

Likelihood Weighting – Algorithm (3)

```
score =  $\prod_{X_j \in A} P(\tilde{x}_j | \tilde{\text{pa}}_j);$ 
for (each  $X_j \in V - A$ );
    for ( $k = 1; k <= \#$  of values in  $X_j$ 's space;  $k + +$ )
        if ( $x_{jk} == \tilde{x}_j$ )
             $\hat{P}(x_{jk} | \text{a}) = \hat{P}(x_{jk} | \text{a}) + score;$ 
    }
    for (each  $X_j \in V - A$ )
        for ( $k = 1; k <= \#$  of values in  $X_j$ 's space;  $k + +$ )
            normalize  $\hat{P}(x_{jk} | \text{a});$ 
}
```

Abductive Inference

Abductive inference := determining the most probable explanation for a set of findings.

Definition 4.2: Let (G, P) where $G = (V, E)$ be a BN, let $M \subseteq V, D \subseteq V$, and $M \cap D = \emptyset$. M is called the **manifestation set** and D is called the **explanation set**. Let m be a set of values of the variables in M . Then

$$d_{max} = \underset{d \in D}{argmax} P(d|m)$$

is called a **most probable explanation (MPE)** for m . The process of determining such a set is called **abductive inference**.

Brute force solution (chain rule) with $D = \{D_1, \dots, D_k\}, M = \{M_1, \dots, M_j\}, d = \{d_1, \dots, d_k\}$;

$$P(d|m) = P(d_1, \dots, d_k | m_1, \dots, m_j)$$

- ⇒ All these probabilities can be computed with our inference algorithms
- ⇒ But with exponential time complexity

- ⇒ Best-first search (best = highest bound) with branch-and-bound pruning is usually faster
- ⇒ For more details consult book.

SS 2014 – Bayesian Networks

Influence Diagrams

*University of Augsburg
Multimedia Computing and Computer Vision,
Prof. Dr. Rainer Lienhart
Rainer.Lienhart@informatik.uni-augsburg.de
www.multimedia-computing.org*

Reference

Richard E. Neapolitan. **Learning Bayesian Networks.** *Prentice Hall Series in Artificial Intelligence*, ISBN 0-13-012534-2.

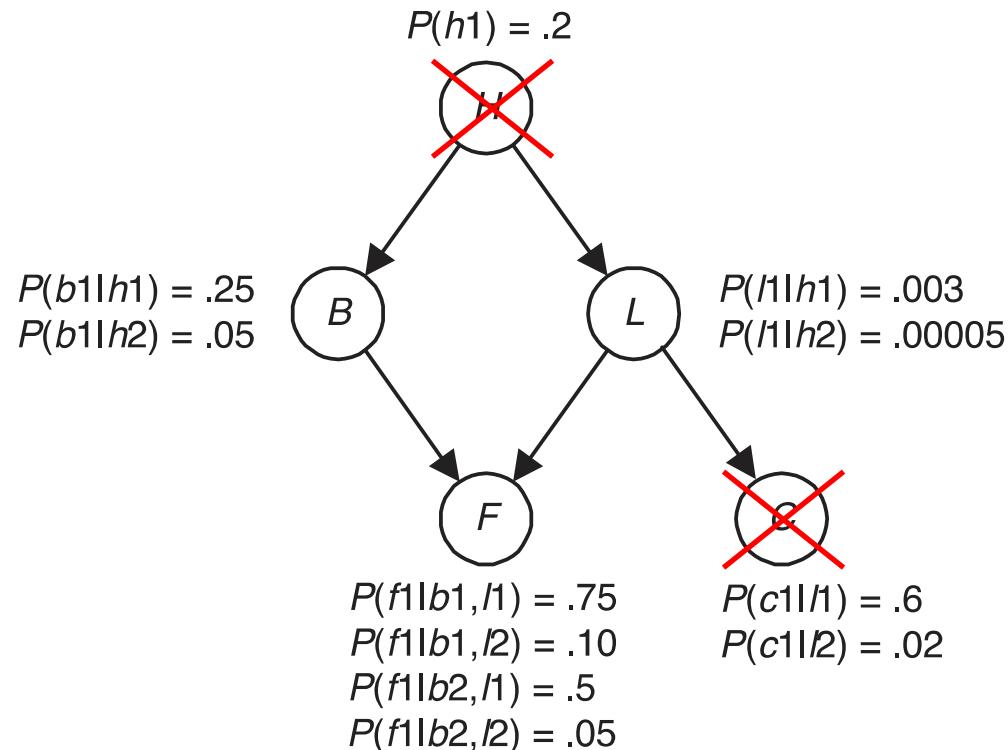
Don't forget. Reading the book chapters 1 – 6 is mandatory.

Chapter on ***Influence Diagrams***
(chapter 5)

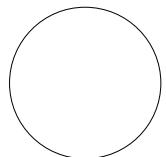
Figures and text are taken from that book

Motivation

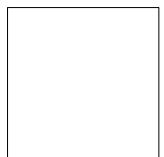
- Info obtained by inference in a BN can be used to arrive at a decision even though the BN itself does not recommend a decision.
- BN will be extended here to recommend a decision.



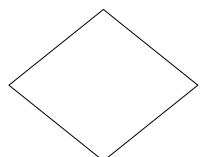
3 Kinds of Nodes



→ Chance node →representing random variables

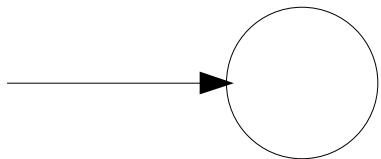


→ Decision node →representing decisions

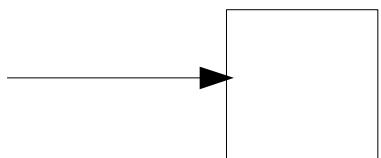


→ Utility node →Random variable whose possible values are the utilities of the outcomes

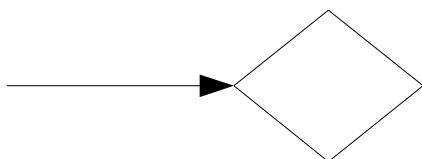
Meaning of Edges



- Value of node is probabilistically dependent on the value of parent



- Value of parent is known at time of decision making --> sequence



- Value of node is deterministically dependent on the value of parent

Influence Diagram = Augmented BN

Influence Diagram := BN + (decision & utility nodes)

- ⇒ Chance nodes satisfy the Markov condition with the probability distribution
- ⇒ Decision nodes **must** be ordered based on the order in which decisions are made using edges
 - e.g., order: $D_1, D_2, D_3 \implies$ in the graph: $D_1 \rightarrow D_2 \rightarrow D_3$

Example 5.13

You have the opportunity to buy a spiffy car for \$10,000 and sell it to a prospect buyer for \$11,000 iff car is in excellent mechanical condition (e.m.c.) .

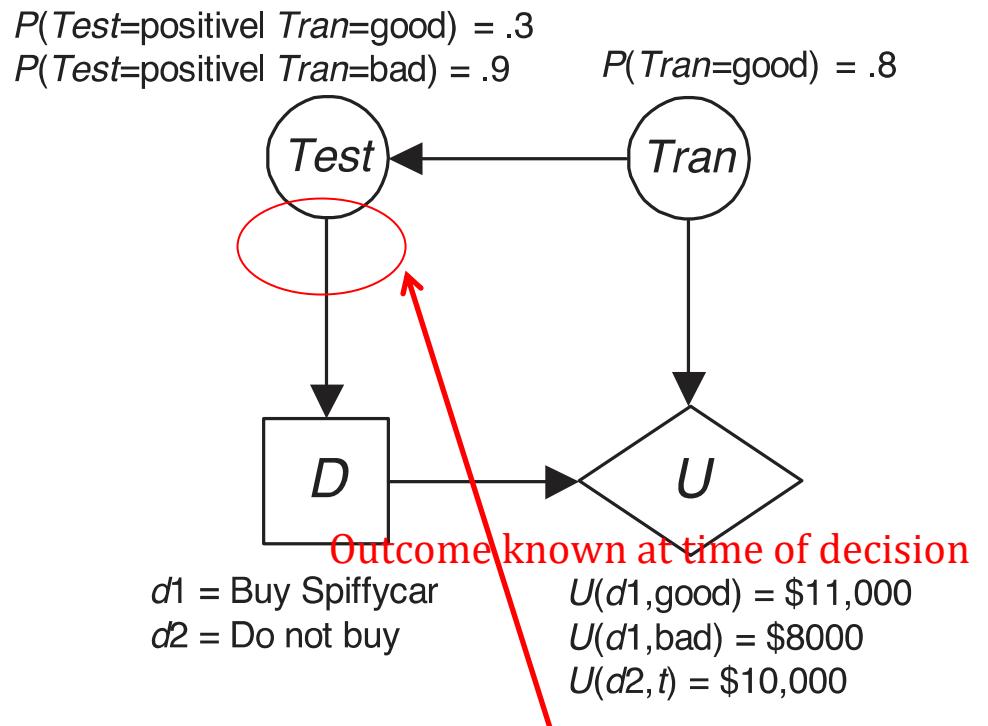
Everything is in e.m.c. except you are not sure about the condition of the transmission. If broken, repair will cost \$3000 before you can sell the car.

A friend can run a check of the transmission with the following success rate:

$$P(\text{Test=positive} \mid \text{Tran=good}) = .3$$

$$P(\text{Test=positive} \mid \text{Tran=bad}) = .9$$

$$P(\text{Tran = good}) = .8$$

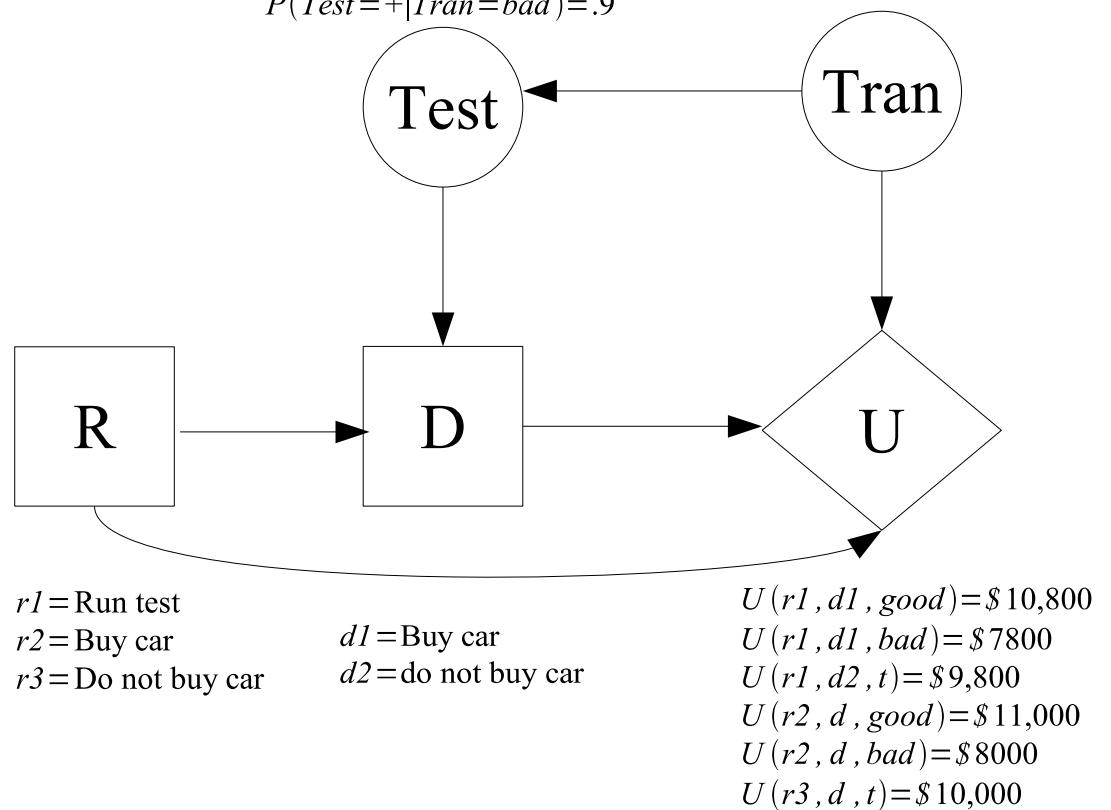


Example 5.14

Same situation as in previous example except that this time the transmission test cost \$200. So additionally, you have to decide "R" whether to perform the test or not before deciding to buy the car

No arrow from "R" to "Test" because test outcome does not depend on "R"

$$P(\text{Test} = + | \text{Tran} = \text{good}) = .3$$
$$P(\text{Test} = + | \text{Tran} = \text{bad}) = .9$$
$$P(\text{Tran} = \text{good}) = .8$$



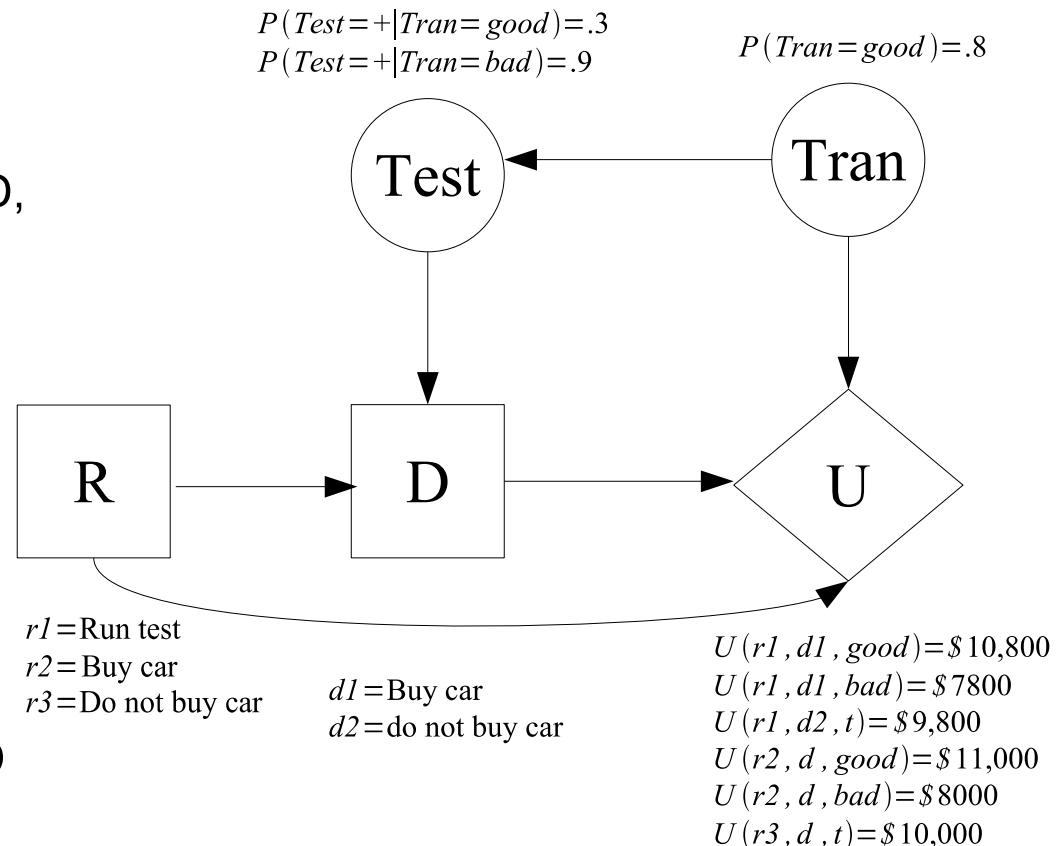
Solving Example 5.14 (1)

Goal: Determine decision that maximized the expected value (E) of the utility (U). We call that value EU.

Note: Since there is an edge from R to D, decision R is made to be first!

$$\begin{aligned}
 EU(r1) &= E(U|r1) \\
 &= \sum_{d, Tran} P(d, Tran|r1)U(r1, d, Tran) \\
 &= P(d1, good|r1)U(r1, d1, good) \\
 &\quad + P(d1, bad|r1)U(r1, d1, bad) \\
 &\quad + P(d2, good|r1)U(r1, d2, good) \\
 &\quad + P(d2, bad|r1)U(r1, d2, bad)
 \end{aligned}$$

Further note: D and Tran are not dependent on R; however, no decision D is made for certain values of decision R



Solving Example 5.14 (2)

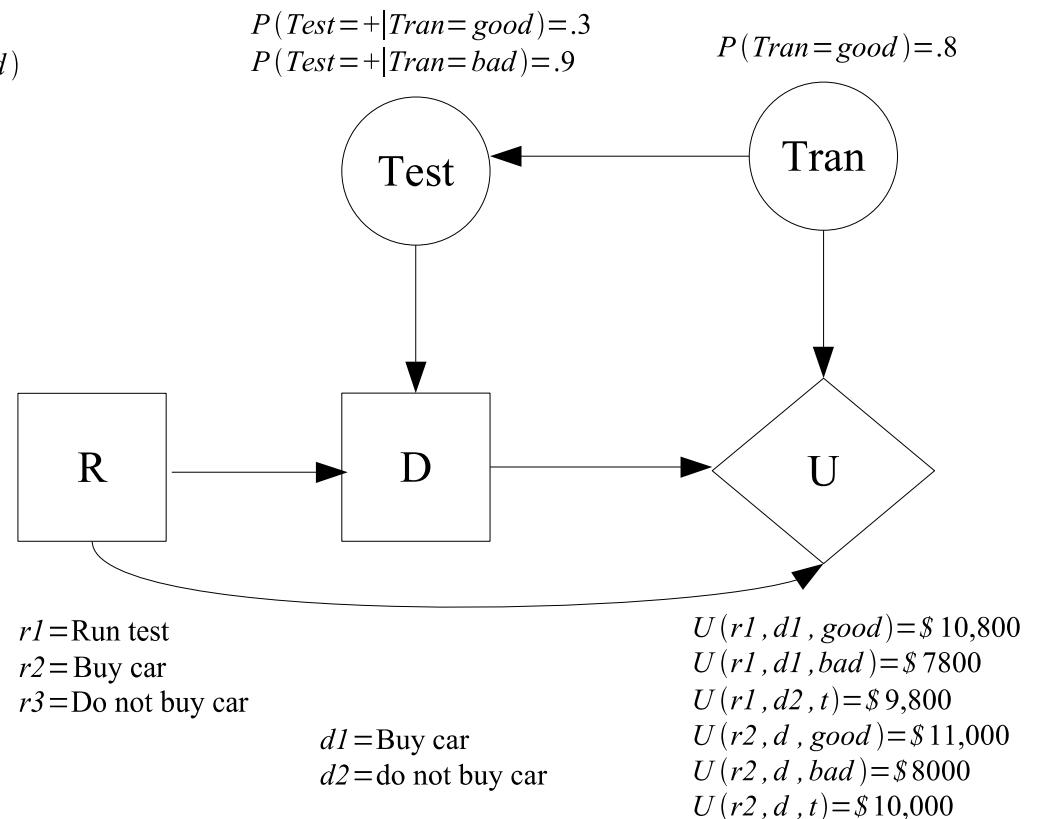
$$\begin{aligned}
 P(d1, good | r1) &= P(d1, good) = P(d1|good)P(good) \\
 &= \left[\sum_{test=-}^+ P(d1, test | good) \right] P(good) \\
 &= \left[\sum_{test=-}^+ P(d1 | test, good) P(test | good) \right] P(good) \\
 &= \left[\sum_{test=-}^+ P(d1 | test) P(test | good) \right] P(good) \\
 &= [(1)P(-|good)+(0)P(+|good)]P(good) \\
 &= P(-|good)P(good) \\
 &= (.7)(.8)=.56
 \end{aligned}$$

$$P(d1, bad | r1) = .02$$

$$P(d2, good | r1) = .24$$

$$P(d2, bad | r1) = .18$$

$$\begin{aligned}
 EU(r1) &= E(U|r1) \\
 &= P(d1, good | r1)U(r1, d1, good) \\
 &\quad + P(d1, bad | r1)U(r1, d1, bad) \\
 &\quad + P(d2, good | r1)U(r1, d2, good) \\
 &\quad + P(d2, bad | r1)U(r1, d2, bad) \\
 &= (.56)(\$10,800) + (.02)(\$7800) \\
 &\quad + (.24)(\$9800) + (.18)(\$9800) \\
 &= \$10320 \\
 \text{EU}(r2) &= \$10,400 \\
 EU(r3) &= \$10,000
 \end{aligned}$$



Solving Influence Diagrams

Assume

- n decision nodes D_1, D_2, \dots, D_n
- Indexes are in the order decisions must be made

```
Determine_EU(i, d1, ..., di-1 /* IN */, maxDi /* OUT */
*/
{
    if (i==n)
        maxEUd1, ..., dn = -∞
        For all dn possible at Dn
            EUd1, ..., dn = EU(d1, ..., dn) = E(U |
d1, ..., dn)
            if (EUd1, ..., dn > maxEUd1, ..., dn)
                maxEUd1, ..., dn = EUd1, ..., dn
                maxDn = { dn }
        else
            maxEUd1, ..., di = -∞
            For all di possible at Di
                maxDi+1 = ∅
```

```
        EUd1, ..., di = Determine_EU(i+1, d1, ..., di, maxDi+1)
        if (EUd1, ..., di > maxEUd1, ..., di)
            maxEUd1, ..., di = EUd1, ..., di
            maxDi = maxDi+1 union { di }

        return maxEUd1, ..., di // to maxDi by reference
    }

main()
{
    maxD1 = ∅
    Call Determine_EU( 1, ∅, maxD1)
}
```

SS 2014 – Bayesian Networks

Dynamic Networks

*University of Augsburg
Multimedia Computing and Computer Vision,
Prof. Dr. Rainer Lienhart
Rainer.Lienhart@informatik.uni-augsburg.de
www.multimedia-computing.org*

Reference

Richard E. Neapolitan. **Learning Bayesian Networks.** *Prentice Hall Series in Artificial Intelligence*, ISBN 0-13-012534-2.

Don't forget. Reading the book chapters 1 – 6 is mandatory.

Chapter on ***Influence Diagrams***
(chapter 5)

Figures and text are taken from that book

Modeling Temporal Relationships & Notations

So far:

- BN represents probabilistic relationships among a set of random variables at some point in time

Now:

- BN relates the value of some variable to its values and the values of other variables at previous points in time
 - ⇒ Called **dynamic Bayesian Network (DBN)**
 - ⇒ **Goal:** Modeling temporal processes

Notations:

Given random variables X_1, \dots, X_n , the column vector

$$\mathbf{X} = (X_1, \dots, X_n)^T = \begin{pmatrix} X_1 \\ \vdots \\ X_n \end{pmatrix}$$

is called a **random vector**. A **random matrix** is defined in the same manner.

We will also use \mathbf{X} to denote the set of random variables $\mathbf{X} = \{X_1, \dots, X_n\}$. Similarly, we use \mathbf{x} to denote both a vector of values of \mathbf{X} and the set of values that constitute \mathbf{x} .

Notations (2)

$P(\mathbf{x})$ denotes the joint probability distribution $P(x_1, \dots, x_n)$. Random vectors are called **independent** if the sets of variables that constitute them are independent. A similar definition holds for conditional independence.

Assumption: Changes in temporal process occur only between discrete time points $t \in \{0, 1, \dots, T\}$

Define:

- $\mathbf{X} = \{X_1, \dots, X_n\}$ set of features whose values change over time
- $X_i[t]$ = random variable X_i at time $t \in \{0, 1, \dots, T\}$
- $\mathbf{X}[t] = (X_1[t], \dots, X_n[t])^T$ random vector \mathbf{X} at time $t \in \{0, 1, \dots, T\}$

Assumption: For all t , each $X_i[t]$ has the same space (depending only on i); called the **space of \mathbf{X}_i** .

Simple Dynamic BN (1)

Simple Dynamic BN := BN consisting of the $T+1$ random vectors $\mathbf{X}[t]$ plus the following specifications:

1. An **initial BN** consisting of (a) an initial DAG G_0 containing the variables in $\mathbf{X}[0]$ and (b) an initial probability distribution P_0 of these variables.
2. A **transition BN** that is a template consisting of (a) a transition DAG G_{\rightarrow} containing the variables in $\mathbf{X}[t] \cup \mathbf{X}[t + 1]$; and (b) a transition probability distribution P_{\rightarrow} that assigns a conditional probability to every value of $\mathbf{X}[t + 1]$ given every value of $\mathbf{X}[t]$:

$$P_{\rightarrow}(\mathbf{x}[t + 1] | \mathbf{x}[t])$$

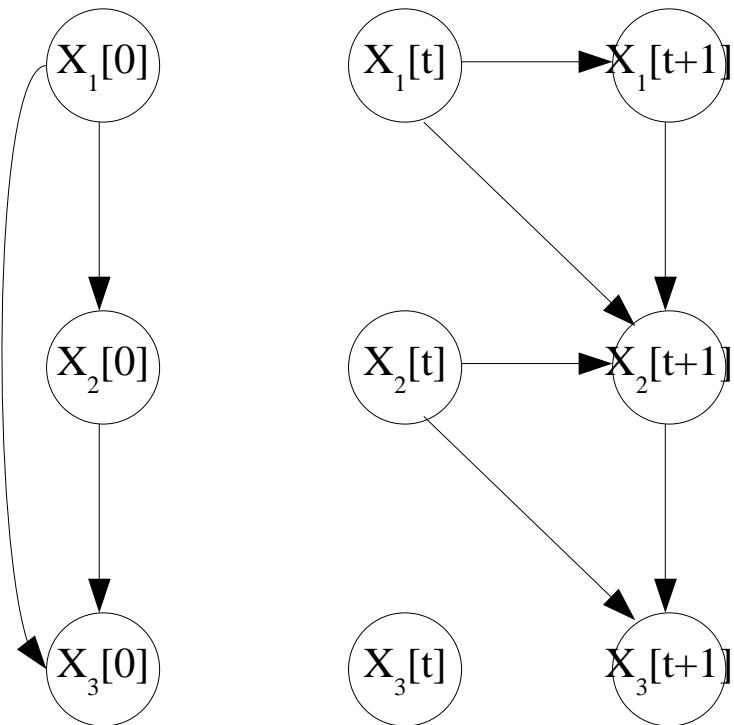
3. The dynamic BN containing the variables that constitute the T random vectors consists of (a) the DAG composed of the DAG G_0 and for $t \in \{0, 1, \dots, T - 1\}$ the DAG G_{\rightarrow} evaluated at t ; and (b) the following JPD:

$$\begin{aligned} P(\mathbf{x}[0], \dots, \mathbf{x}[T]) &= P_0(\mathbf{x}[0]) \prod_{t=0}^{T-1} P_{\rightarrow}(\mathbf{x}[t + 1] | \mathbf{x}[t]) \\ &= P_0(\mathbf{x}[0]) \prod_{t=0}^{T-1} \prod_{i=0}^n P_{\rightarrow}(x_i[t + 1] | \text{pa}_i[t + 1]) \end{aligned}$$

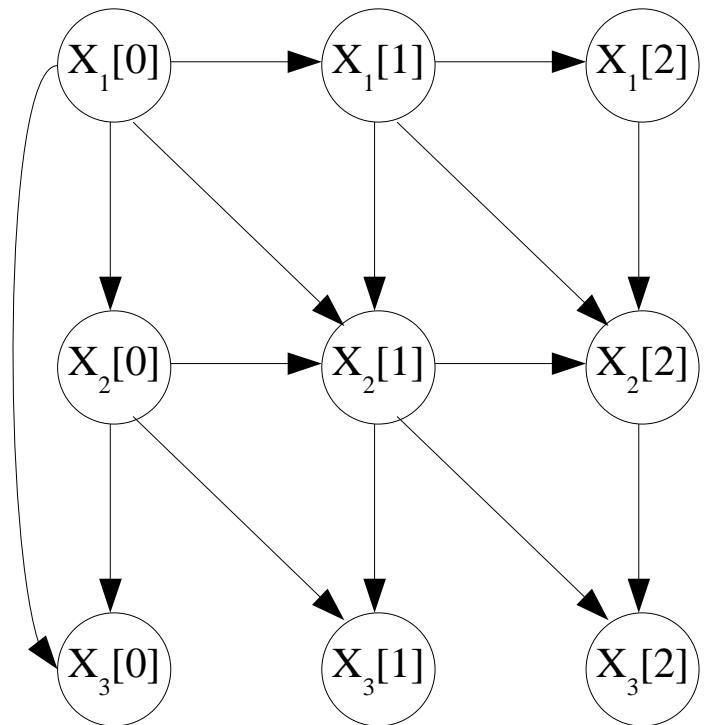
Note that $\text{pa}_i[t + 1]$ denotes the values of the parents of $X_i[t + 1]$, which can be in $\mathbf{X}[t]$ and $\mathbf{X}[t + 1]$.

Simple Dynamic BN (1)

DBN Spec



DBN Example with T=2



Implicit assumptions so far:

- All information needed to predict a world state at time t is contained in the description of the world at time $t - 1$.
⇒ **Markov property**: $P(x[t + 1] | x[t], \dots, x[0]) = P([t + 1] | x[t])$
- Process is stationary, i.e., $P(x[t + 1] | x[t])$ is the same for all t .

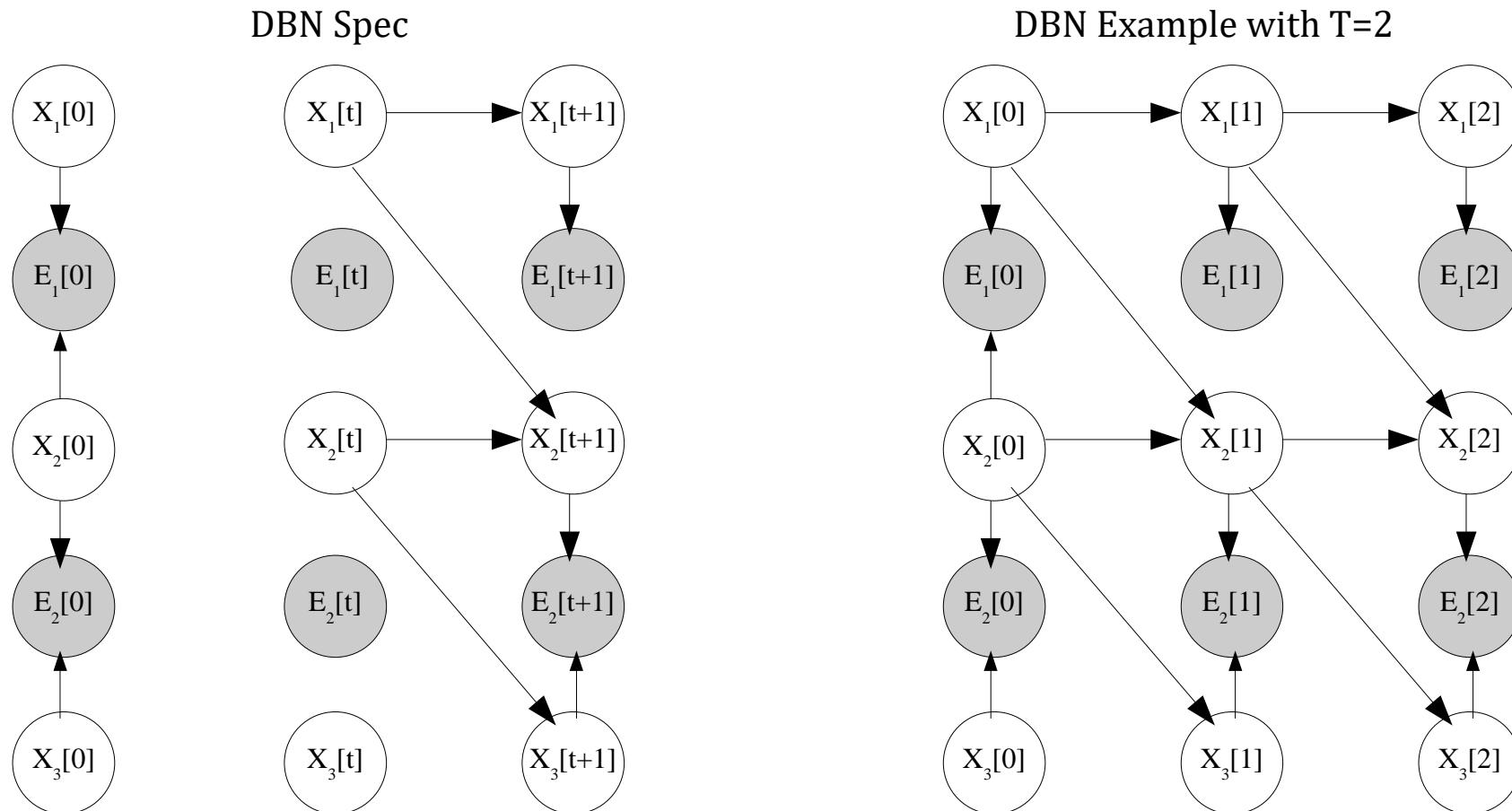
These assumptions

- are not necessary, but reduce the complexity of representation and evaluation of DBNs, and
- are often reasonable.

Inference in DBN can be done using the standard Pearl algorithm

Special DBN Subclass – Bayes Filter (1)

Networks in different time steps are connected only thorough non-evidence variables. The variables labeled with an E are the evidence variables and are instantiated in each time step. For instance, the E are the random variable for sensor measurements.



Special DBN Subclass – Bayes Filter (2)

Let be:

- $e[t]$ = set of values of the evidence variables at time step t
- $f[t]$ = set of values of the evidence values up to and including time step t , i.e., $f[t] = (e[0], \dots, e[t])$

Assume we know $P(x[t] | f[t])$

Then

$$\begin{aligned} \underbrace{P(x[t+1] | f[t])}_{P(x[t+1])} &= \sum_{x[t]} P(x[t+1] | x[t], f[t]) P(x[t] | f[t]) \\ &= \sum_{x[t]} P(x[t+1] | x[t]) P(x[t] | f[t]) \end{aligned}$$

and

$$\begin{aligned} P(x[t+1] | f[t+1]) &= P(x[t+1] | f[t], e[t+1]) \\ &= \alpha P(e[t+1] | x[t+1], f[t]) P(x[t+1] | f[t]) \\ &= \alpha P(e[t+1] | x[t+1]) P(x[t+1] | f[t]) \end{aligned}$$

$P(e[t+1] | x[t+1])$ can be computed using an inference algorithm.

Special DBN Subclass – Bayes Filter (3)

Algorithm:

- Initialization: $P(x[0] | f[0]) = P(x[0] | e[0])$
- For $t=0$ to $T-1$

$$P(x[t+1] | f[t]) = \sum_{x[t]} P(x[t+1] | x[t]) P(x[t] | f[t])$$

$$P(x[t+1] | f[t+1]) = \alpha P(e[t+1] | x[t+1]) P(x[t+1] | f[t])$$

Note: We need to keep only the parameters and network structure of two time steps at any time for the computation.

Example for that kind of DBN structure is given in the next slides.

Also: Our lecture on “*Probabilistic Robotics*” is completely based on this model. This lecture fills the probability distributions with life. There, $P(x[t+1] | f[t])$ is called the **control update** and $P(x[t+1] | f[t+1])$ the **measurement update**.

Example: Mobile Target Localization (1)

DBN developed by K. Basye, T. Dean, J. Kirman, M. Lejter. **A decision-theoretic approach to planning, perception, and control.** *IEEE Expert*, Volume 7, Issue 4, Aug. 1992 Page(s):58 - 65

Goal:

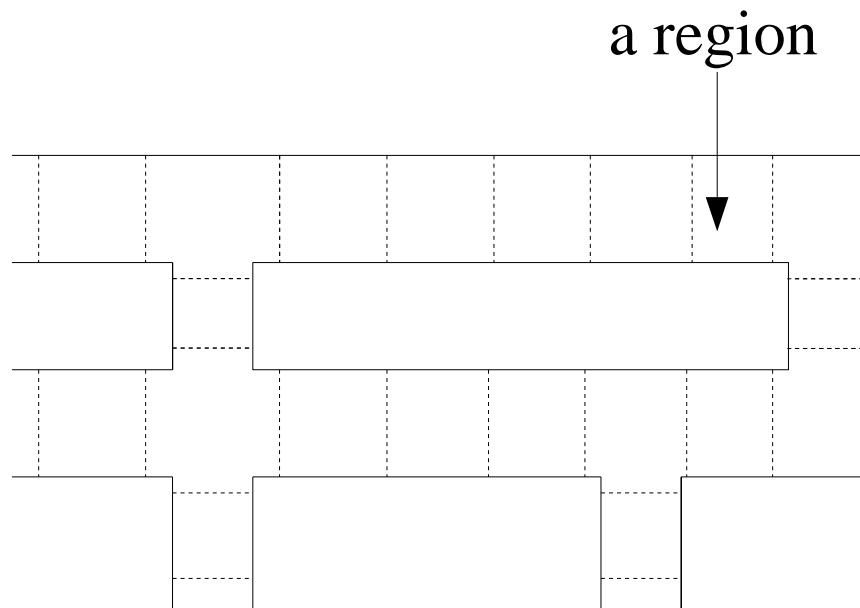
- Track a target while maintaining knowledge of one's own location inside a given world

Model:

- Robot is supplied with map of world which is divided into corridors and junctions.
- **State space** of
 - *Location of the target*: set of all regions
 - *Location of robot*: set of all regions augmented with four directions in which the robot can face

Random variables:

- L_R and L_A (= location of robot and target, respectively)



Tessellation of corridor layout

Example: Mobile Target Localization (2)

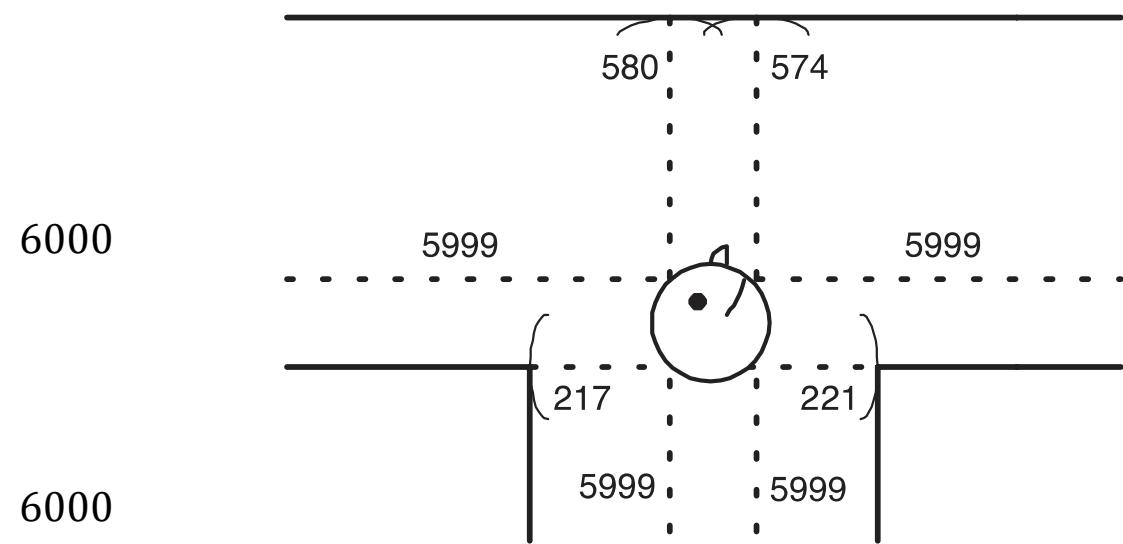


Figure 5.26: Sonar readings upon entering a T-junction.

- Target and robot are mobile
- Robot has 8 sonar sensors (2 in each direction) to measure distance from wall between 30 and 6000 mm. 6000mm means 6000 or more (see figure for example).
- Need probabilistic mapping from raw sensor data to abstract **sensor space** consisting of corridor, T-junction, L-junction, dead end, open space, and crossing.
- Robot has one forward looking camera with which it can identify the target if present. The size of the target in the camera image is used to infer coarsely the distance of the target

Example: Mobile Target Localization (3)

Variable	What the Variable Represents
L_R	Location of the robot
L_A	Location of the target
E_R	Sensor reading regarding location of robot
E_A	Camera reading regarding location of target relative to robot

Actions available to the robot and target:

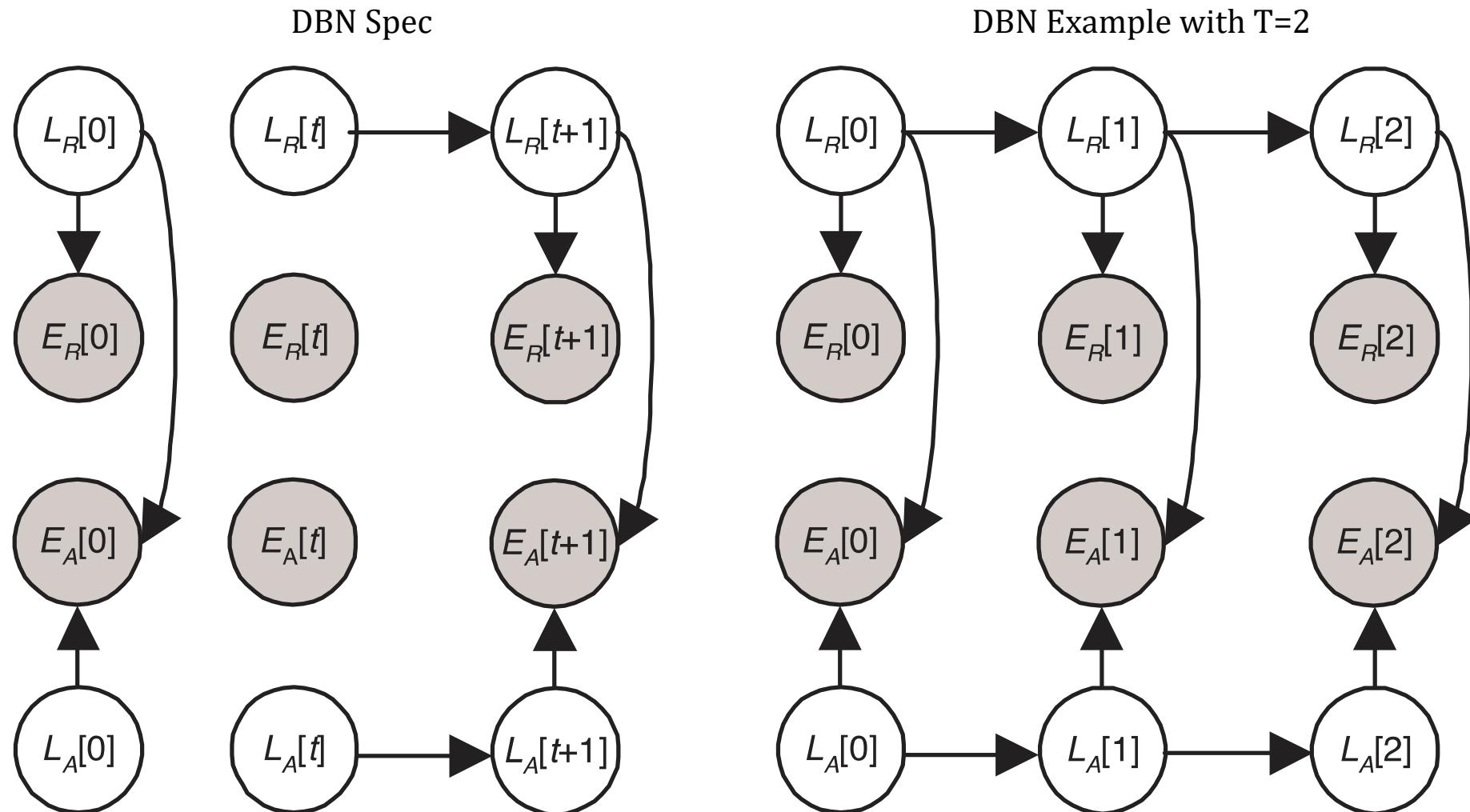
- Travel down the corridor the length of one region
- Turn left/right around the corner
- Turn around, ...

In a DBN these actions are performed in some preprogrammed probabilistic way, which is not related to the sensory data: the location of a robot at time $t+1$ is a probabilistic function of its location at time t .

$P(e_R|l_R)$ can be obtained by repeatedly putting the robot in positions l_R and seeing how often the reading e_R is obtained.

$P(e_A|l_R, l_A)$ can be obtained by repeatedly putting the robot and the target in positions l_R and l_A , respectively, and seeing how often the reading e_A is obtained.

Example: Mobile Target Localization (4)



Example: Mobile Target Localization (5)

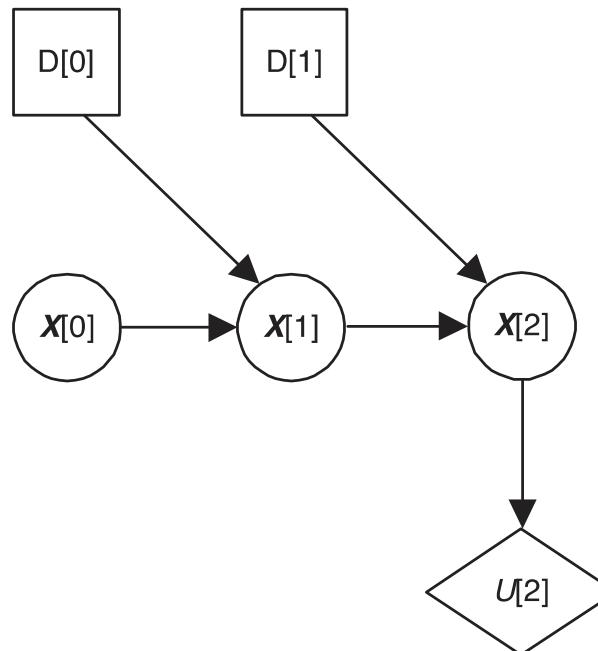
DBN: Robot can sometimes view the target, but does not make any effort to track. It just moves probabilistically around.

Needs ***influence diagram*** to create a robot tracking a target (= probabilistic inference with decision making)

Dynamic Influence Diagram

$\text{DID} := \text{DBN} + \text{Decision nodes } D = \{D[0], \dots, D[T - 1]\} + \text{Utility node}$

- $D[t]$ influences system state at time $t+1$.
- Determine decision at time t that maximizes expected utility at some point T in the future.



- Chance node at each time step in the figure represent the entire DAG at that time step, and so the edges represent sets of edges

Example: Mobile Target Localization (6)

Goal:

- Robot should track target by deciding on its move at time t based on its observed evidence at time t .
⇒ Robot makes decision $D[t]$ at time t about its next movement based on observed evidence in time t that maximizes some expected utility function.

Assumptions:

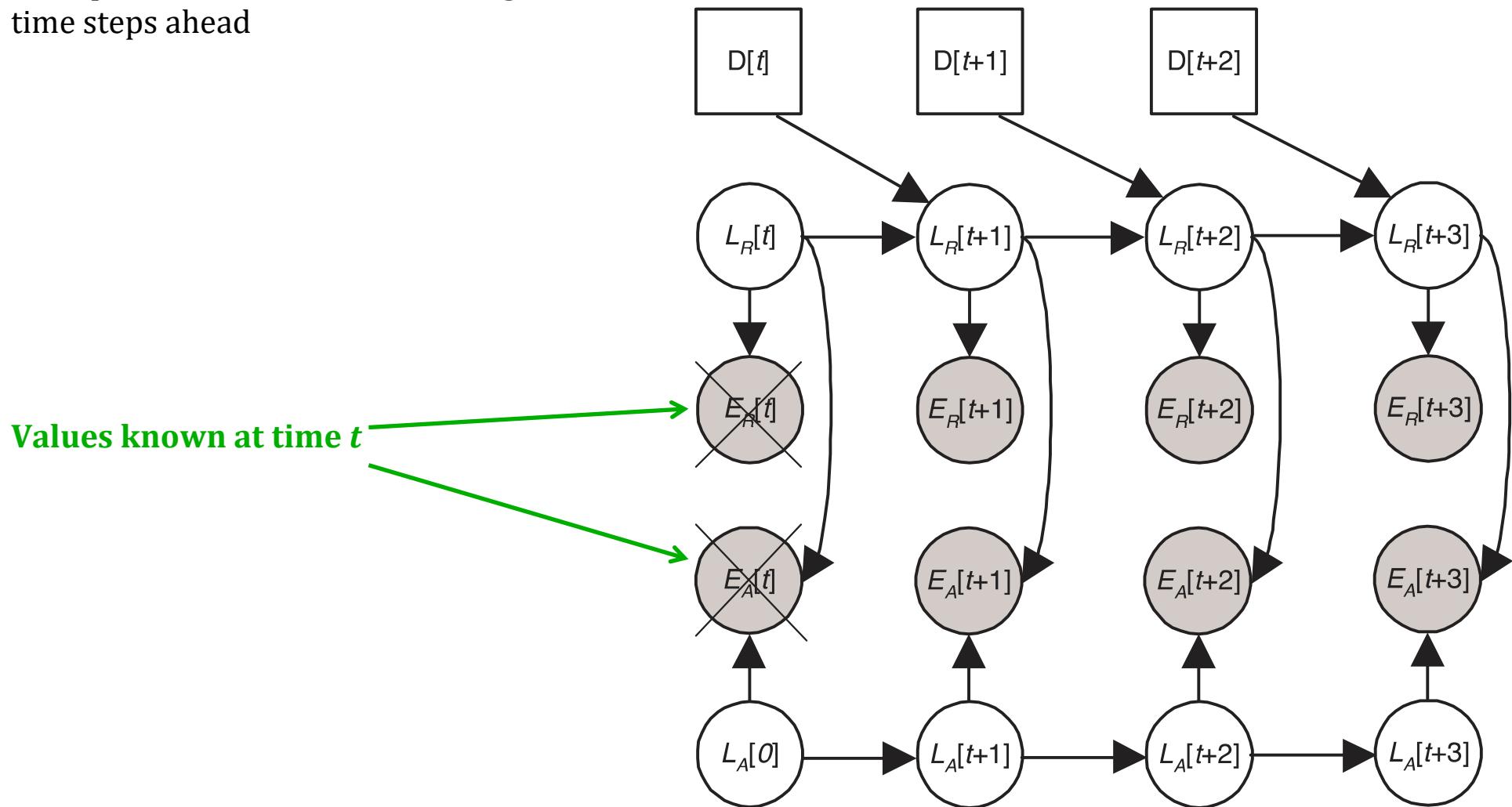
- There are errors in the robots movements. Thus,

$$P(L_R[t + 1] | L_R[t], D[t])$$

- Can be estimated by repeatedly placing the robot in a location $L_R[t]$, perform an action $D[t]$, and then observe its new location $L_R[t + 1]$.

Example: Mobile Target Localization (7)

Example where the robot is looking three time steps ahead



Define Utility Function (1)

Determine decision at time t by looking M time steps into the future.

Thus let be

1. $d_M = \{d[t], d[t+1], \dots, d[t+M-1]\}$
= set of values of the next M decisions including the current one.
2. $f_M = \{e_R[t+1], e_A[t+1], e_R[t+2], e_A[t+2], \dots, e_R[t+M], e_A[t+M]\}$
= set of values of the evidence variables observed after the decisions are made.
3. d_k, f_k = first k decisions and evidence pairs in the above sets d_M and f_M , respectively, ($1 \leq k \leq M$)

Define

$$U_k(f_k, d_k) = - \min_u \underbrace{\sum_v dist(u, v) P'(L_A[t+k] = v | f_k, d_k)}_{\text{expected value of the distance between the target and a given location } u} \underbrace{\text{The smaller this value is the more likely it is the target is close to } u}_{}$$

where

$v, u \in \text{space of } L_A$ (set of target locations).

The location \check{u} , which has the minimum expected value, is our best guess as where the target is if we make decisions d_k and obtain observations f_k . So the utility of the decision and the observations is the expected value for \check{u} .

Define Utility Function (2)

Utility at time steps k :

$$EU_k(d_k) = \sum_{f_k} U_k(f_k, d_k) P'(f_k | d_k)$$

All time steps $t + 1$ to $t + M$ are taken into account for the overall utility computation by means of a weighted sum of the utilities at each time step:

$$EU(d_M) = \sum_{k=1}^M \gamma_k EU_k(d_k)$$

where γ_k decreases with k to discount the impact of future consequences, i.e.,

$$\gamma_k \geq \gamma_{k+1} \forall k, \gamma_k = 0 \text{ for } k > M.$$

Summary

$t=0$

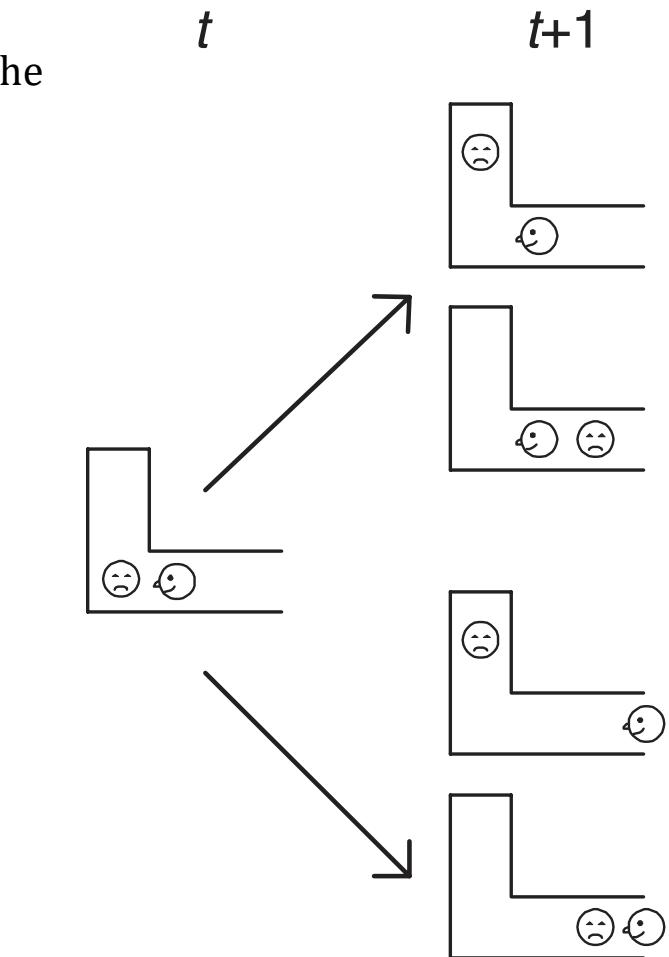
Repeat

- In time step t the robot reads its sensory data from sonar sensors & camera
- In time step t the robot updates its probability distribution based on the evidence from sonar sensors & camera
- The expected utility of a sequence of decisions/actions is evaluated (time horizon M periods ahead)
- This is done for all possible sequences of decisions, and the one that maximizes the expected utility is chosen.
- The first decision/action in that sequences is executed
- $t++$

Emergent Behavior

Emerging behavior := behavior that is not purposefully programmed into the robot, but that emerges as a consequence of the model

Staying close to the target may not be optimal.



SS 2014 – Bayesian Networks

Parameter Learning: Binary Variables

*University of Augsburg
Multimedia Computing and Computer Vision,
Prof. Dr. Rainer Lienhart
Rainer.Lienhart@informatik.uni-augsburg.de
www.multimedia-computing.org*

Reference

Richard E. Neapolitan. **Learning Bayesian Networks.** *Prentice Hall Series in Artificial Intelligence*, ISBN 0-13-012534-2.

Don't forget. Reading the book chapters 1 – 6 is mandatory.

Chapter on ***Parameter Learning: Binary Variables***
(chapter 6)

Figures and text are taken from that book

Definitions

So far:

- DAG in BN was hand-constructed by domain expert
- CP were assessed by
 - Expert (subject probabilities),
 - Learned from data (relative frequencies), or
 - Combination of both (we will learn in this section)

Structure := shape of a DAG (= set E of edges)

Parameters := the conditional probability distributions (CPDs) of the nodes

Parameter Learning := methods for learning the conditional probabilities **from data**

Structure Learning := methods for learning the DAG **from data**

Note: Parameter learning only possible when probabilities are relative frequencies.

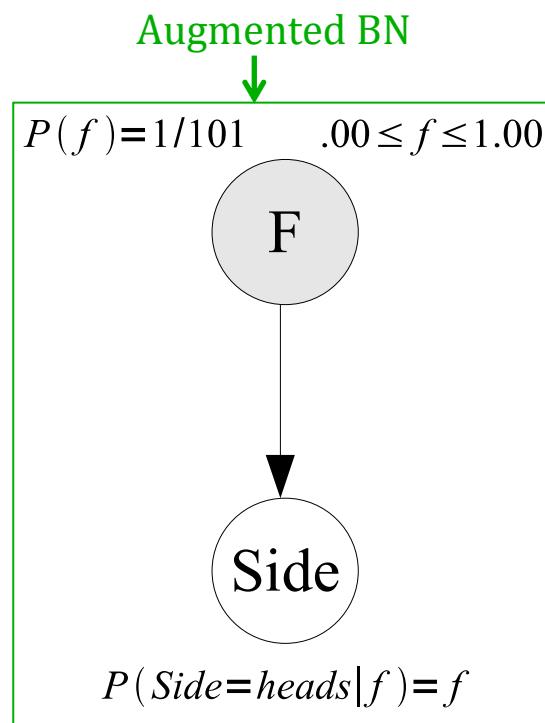
Belief concerning the value of a relative frequency is represented by a subjective probability distribution

Learning a Single Binary Parameter

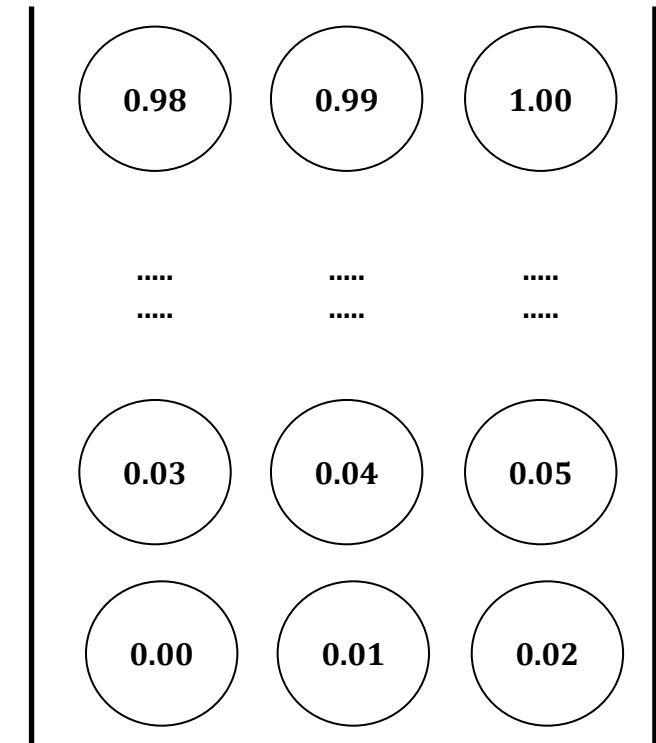
All Relative Frequencies Are Equally Probable (1)

Example:

- An urn containing 101 coins, each with a different propensity for landing heads.
- Belief concerning the value of a relative frequency is represented by a subjective probability distribution
- Assume: All relative frequencies are equally probable (**principle of indifference**)



Node represents our belief about relative frequency



Suppose a coin is picked at random. Then

$$P(\text{Side} = \text{heads} | f) = f$$

if $\text{Side} = \{\text{head}, \text{tail}\}$ and $F = \{(0.00), (0.01), \dots, (0.99), (1.00)\}$.

All Relative Frequencies Are Equally Probable (2)

$$\begin{aligned} P(\text{Side} = \text{heads}) &= \sum_{f=.00}^{1.00} P(\text{Side} = \text{heads}|f)P(f) \\ &= \sum_{f=.00}^{1.00} f \left(\frac{1}{101} \right) \\ &= \left(\frac{1}{101} \right) \sum_{f=.00}^{1.00} f \\ &= \left(\frac{1}{101 \times 100} \right) \sum_{f=0}^{100} f \\ &= \left(\frac{1}{101 \times 100} \right) \left(\frac{100 \times 101}{2} \right) = \frac{1}{2} \end{aligned}$$

= Subjective probability of coin landing heads on first toss.

Continuous Case

- For every real number f between 0 and 1 there is exactly one coin with propensity of f for landing heads.
- Pick coin at random
- Our belief about relative frequency (uniform distribution, principle of indifference):

$$\rho(f) = 1$$
$$\int_0^1 \rho(f) df = 1$$

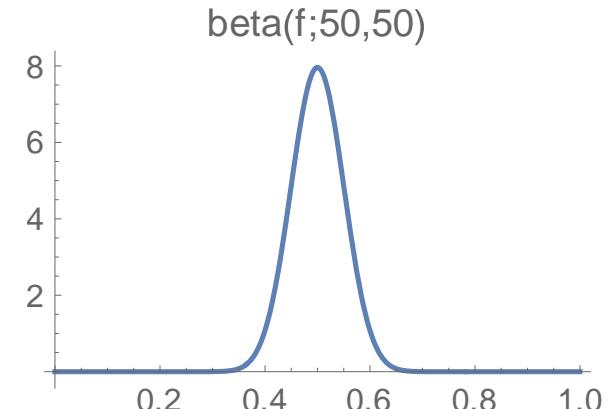
From this it follows

$$\begin{aligned} P(\text{Side} = \text{heads}) &= \int_0^1 P(\text{Side} = \text{heads}, f) df \\ &= \int_0^1 P(\text{Side} = \text{heads}|f) \rho(f) df \\ &= \int_0^1 f \cdot (1) df = \frac{1}{2} \end{aligned}$$

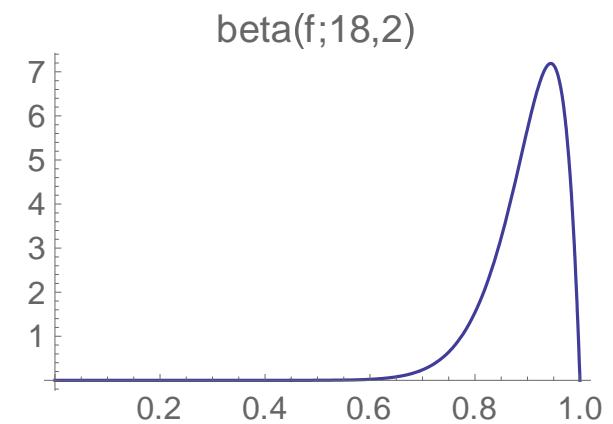
= Subjective probability of coin landing heads on first toss.

All Relative Frequencies Are Not Equally Probable

- In many cases we do not feel that all numbers in $[0,1]$ are equally likely to be the value of a relative frequency
- Example:
 - A coin from someone's pocket
 - Any thumbtack \Rightarrow We feel that it should be about .5



- Relative frequency of individuals in a sample of US citizen that brushed their teeth this morning
- \Rightarrow We feel that it should be about .9



Gamma & Beta Function (1)

The **beta density function** provides us with a natural way for quantifying prior beliefs about relative frequencies and updating these beliefs in the light of evidence.

Gamma function (generalization of factorial function $\Gamma(x + 1) = x!$ for $x \in \mathbb{N}$):

$$\Gamma(x) = \int_0^{\infty} t^{x-1} e^{-t} dt$$

The integral converges iff $x > 0$.

Note that

$$\frac{\Gamma(x+1)}{\Gamma(x)} = x \iff \Gamma(x+1) = x \cdot \Gamma(x)$$

Definition 6.1: The **beta density function** with parameters $a, b \in \mathbb{R}^+ \setminus \{0\}$, $N = a + b$ is

$$\text{beta}(f; a, b) = \rho(f) = \frac{\Gamma(N)}{\Gamma(a)\Gamma(b)} f^{a-1} (1-f)^{(b-1)} \quad 0 \leq f \leq 1.$$

A random variable F that has this density function is said to have a **beta distribution**.

Gamma & Beta Function (2)

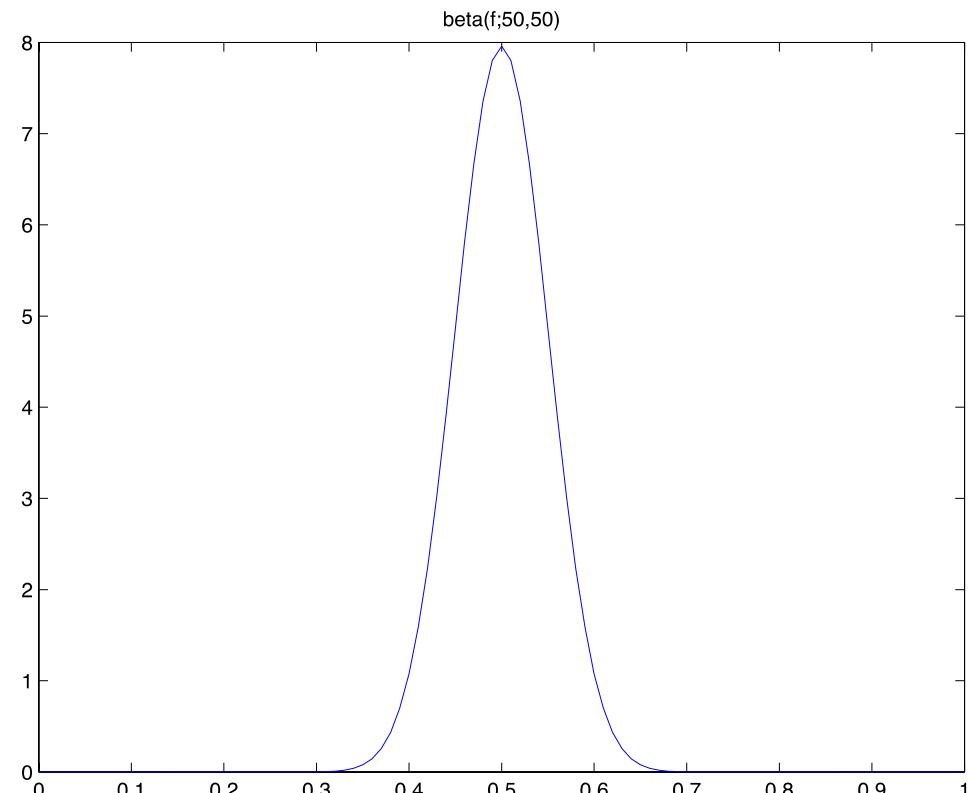
Matlab/Octave:

Filename: *myBeta.m*

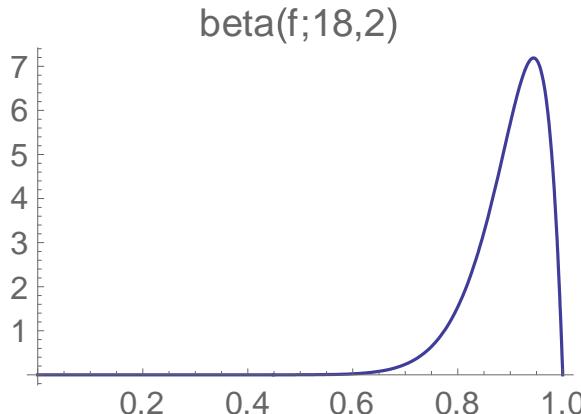
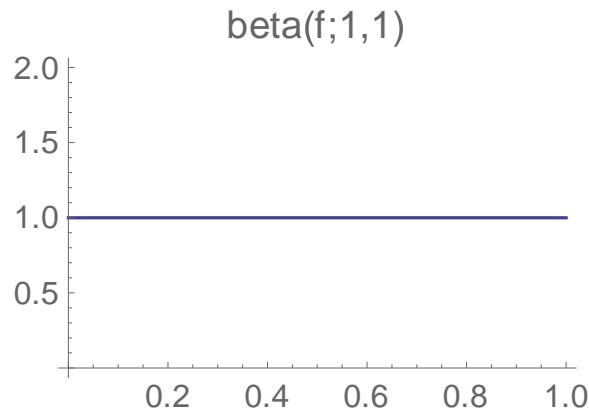
```
function [ret] = myBeta (f, a, b)
    ret = gamma (a+b) ./ gamma (a) ./ gamma (b) .* power (f,a-1) .* power (1-f,b-1);
end
```

Main Window:

```
f=0:0.01:1;
plot(f,myBeta(f,50,50));
title('beta(f;50,50)');
```



Further Examples of $\text{beta}(f; a, b)$



Use beta density function to specify prior beliefs, i.e., your belief that not all numbers in $[0,1]$ are equally likely to be the value of a relative frequency.

Interpretation: If $a, b \in \mathbb{N}$, then the probability experience of an assessor is equivalent to having seen the first outcome a times in $a+b$ trials.

The larger the values of a and b , the more the mass is concentrated around $a/(a + b) = a/N$.

Gamma & Beta Function (2)

Lemma 6.2: For $a, b \in \mathbb{R}^+ \setminus \{0\}$, $N = a + b$ we have

$$\int_0^1 f^a (1-f)^b df = \frac{\Gamma(a+1)\Gamma(b+1)}{\Gamma(a+b+2)}$$

$$\Leftrightarrow \int_0^1 f^{a-1} (1-f)^{b-1} df = \frac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)} = \frac{\Gamma(a)\Gamma(b)}{\Gamma(N)}$$

Lemma 6.3: If F has a beta distribution with parameters $a, b \in \mathbb{R}^+ \setminus \{0\}$, $N = a + b$, then

$$E(F) = \frac{a}{N}$$

(In book: proof on page 292)

Gamma & Beta Function (2)

Lemma 6.3: If F has a beta distribution with parameters $a, b \in \mathbb{R}^+ \setminus \{0\}$, $N = a + b$, then

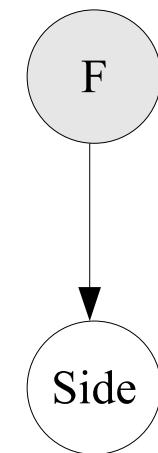
$$E(F) = \frac{a}{N}$$

Representing Belief Concerning a Relative Frequency

Belief about relative frequency



$\rho(f)$



$$P(X=x1|F=f) = f$$

Theorem 6.1: Suppose X is a random variable with two values $x1$ and $x2$, and F is another random variable such that

$$P(X = x1|f) = f$$

Then

$$P(X = x1) = E(F).$$

Corollary 6.1: If the conditions in Theorem 6.1 hold, and F has a beta distribution with parameters $a, b, N = a + b$, then

$$P(X = 1) = \frac{a}{N}$$

Learning a Relative Frequency (1)

Definition 6.2: Suppose

1. We have a set of random variables (or random vectors) $D = \{X^{(1)}, X^{(2)}, \dots, X^{(M)}\}$ such that each $X^{(M)}$ has the same space.
2. There is a random variable F (or a set of random variables F) with the density ρ (or a set of densities ρ) such that the $X^{(h)}$'s are mutually independent conditional on F , and for all values of f of F , all $X^{(h)}$ have the same probability distribution conditional on f .

Then D is called a **sample** of size M with parameter F .

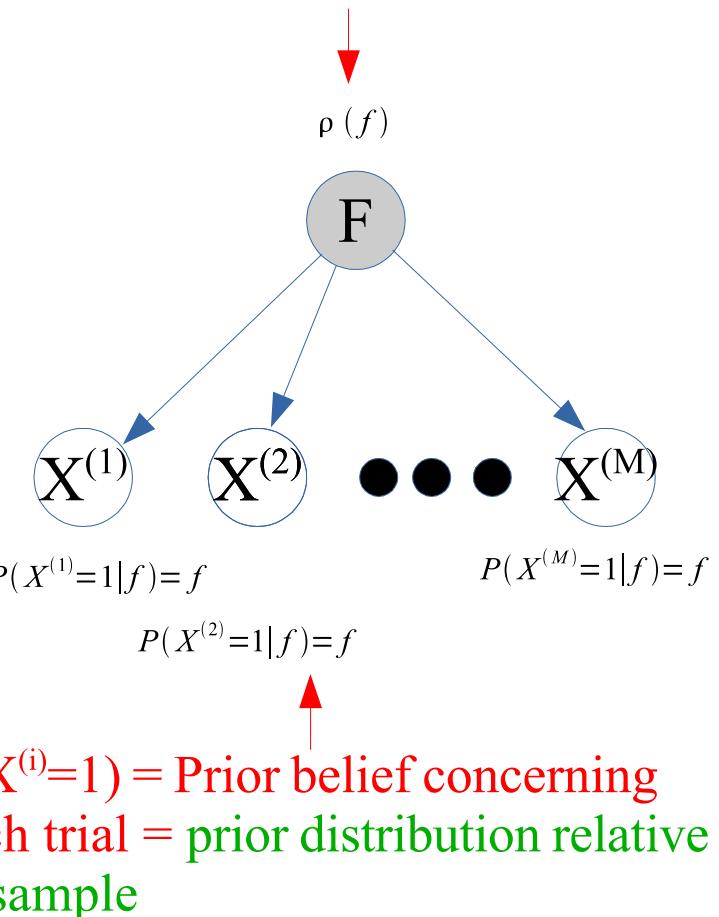
Definition 6.3: Suppose we have a sample of size M such that

1. Each $X^{(h)}$ has space $\{1,2\}$;
2. $F = \{F\}$, F has space $[0,1]$, and for $1 \leq h \leq M$:

$$P(X^{(h)} = 1 | f) = f$$

Then D is called a **binomial sample** of size M with parameter F .

Prior belief concerning unknown parameter(s) = prior density function of the parameter(s)



Learning a Relative Frequency (2)

Lemma 6.4: Suppose F has a beta distribution with parameters $a, b, N = a + b, s$ and t are two integers ≥ 0 , and $M = s + t$. Then

$$E(F^s[1 - F]^t) = \frac{\Gamma(N)}{\Gamma(N + M)} \frac{\Gamma(a + s)\Gamma(b + t)}{\Gamma(a)\Gamma(b)}$$

Proof (p. 297): ...

Lemma 6.5: Suppose F has a beta distribution with parameters $a, b, N = a + b, s$ and t are two integers ≥ 0 , and $M = s + t$. Then

$$\frac{f^s(1 - f)^t \rho(f)}{E(F^s[1 - F]^t)} = \text{beta}(f; a + s, b + t)$$

Proof (p. 298): ...

Learning a Relative Frequency (3)

Theorem 6.2: Suppose

1. D is a binomial sample of size M with parameter F ;
2. we have a set of values $d = \{x^{(1)}, x^{(2)}, \dots, x^{(M)}\}$ of the variables in D (**$d = data\ set$** or just **$data$**)
3. s is the number of variables in d equal to 1; and
4. t is the number of variables in d equal to 2.

Then

$$P(d) = E(F^s[1 - F]^t) \stackrel{\text{if } F \text{ has beta distribution}}{=} \frac{\Gamma(N)}{\Gamma(N + M)} \frac{\Gamma(a + s)\Gamma(b + t)}{\Gamma(a)\Gamma(b)}$$

with parameters $a, b, N = a + b$

Example 6.6 (p. 299)

Learning a Relative Frequency (4)

Theorem 6.3: If the conditions in the Theorem 6.2 hold, then

$$\rho(f|d) = \frac{f^s(1-f)^t \rho(f)}{E(F^s[1-F]^t)}$$

where $\rho(f|d)$ is the density function of F conditional on $D=d$.

(Note that $P(d|f) = f^s(1-f)^t$ and $P(f|d) = \frac{P(d|f)P(f)}{P(d)}$.)

Suppose additionally F has a beta distribution with parameters $a, b, N = a + b$. That is,

$$\rho(f) = \text{beta}(f; a, b)$$

Then

$$\rho(f|d) = \text{beta}(f; a + s, b + t).$$

The density function of F conditioned on data d (i.e., $\rho(f|d)$) is called the **updated density function of the parameters** relative to the sample and the data d .

= our **posterior belief** concerning the unknown parameters.

Learning a Relative Frequency (3)

Theorem 6.2: Suppose

1. D is a binomial sample of size M with parameter F ;
2. we have a set of values $d = \{x^{(1)}, x^{(2)}, \dots, x^{(M)}\}$ of the variables in D (**$d = data\ set$** or just **$data$**)
3. s is the number of variables in d equal to 1; and
4. t is the number of variables in d equal to 2.

Then

$$P(d) = E(F^s[1 - F]^t) \stackrel{\text{if } F \text{ has beta distribution}}{=} \frac{\Gamma(N)}{\Gamma(N + M)} \frac{\Gamma(a + s)\Gamma(b + t)}{\Gamma(a)\Gamma(b)}$$

with parameters $a, b, N = a + b$

Example 6.6 (p. 299)

Theorem 6.3: If the conditions in the Theorem 6.2 hold and F has a beta distribution with parameters $a, b, N = a + b$. That is, $\rho(f) = beta(f; a, b)$, then

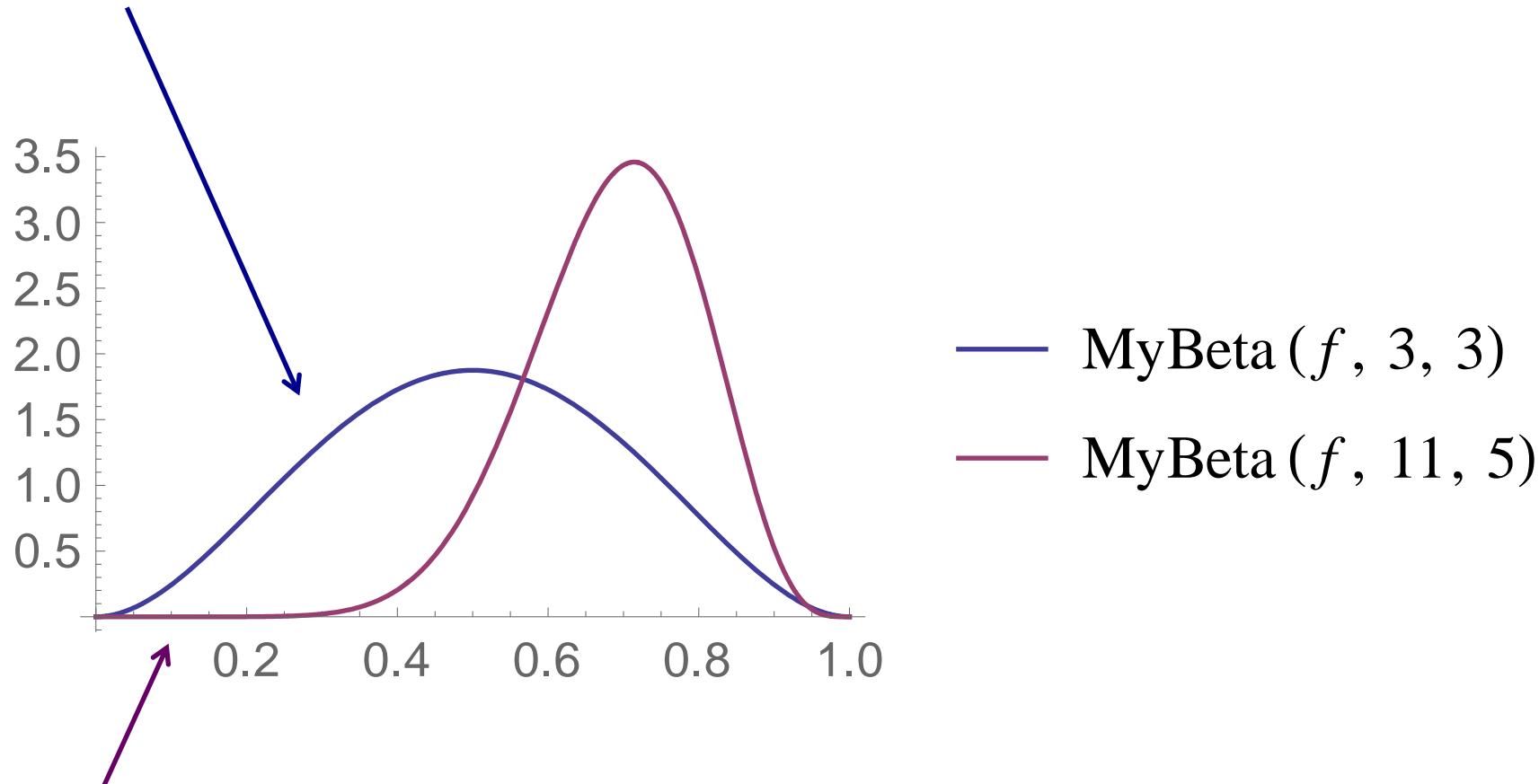
$$\rho(f|d) = beta(f; a + s, b + t).$$

The density function of F conditioned on data d (i.e., $\rho(f|d)$) is called the ***updated density function of the parameters*** relative to the sample and the data d .

= our ***posterior belief*** concerning the unknown parameters.

Learning a Relative Frequency (5)

$\text{beta}(f; 3,3)$:= our **prior belief** concerning the relative frequency of heads in a coin tossing experiment



$\text{beta}(f; 11,5)$:= our **posteriori belief** concerning the relative frequency of heads in a coin tossing game after seeing 8 heads in 10 trials

Learning a Relative Frequency (6)

Theorem 6.4: Suppose the conditions in Theorem 6.2 hold, and we create a binomial sample of size $M + 1$ by adding another variable $X^{(M+1)}$. Then, if D is the binomial sample of size M , the **updated distribution relative to the sample and the data d** is given by

$$P(X^{(M+1)} = 1|d) = E(F|d) \quad \begin{matrix} \text{if } F \text{ has beta distribution} \\ \text{with parameters } a, b, N = a+b \end{matrix} \quad \frac{a+s}{N+M}$$

It represents our posterior belief concerning the next trial.

Proof (p. 301): ...

Note 1:

$$\lim_{M \rightarrow \infty} P(X^{(M+1)} = 1|d) = \lim_{M \rightarrow \infty} \frac{a+s}{N+M} = \frac{s}{M} \quad \begin{matrix} \text{= relative} \\ \text{frequency} \end{matrix}$$

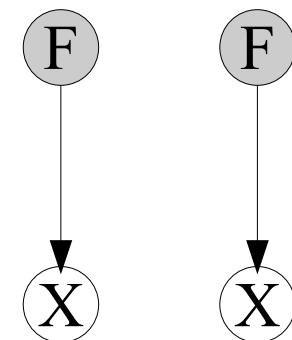
Note 2:

- $E(F|d)$ is our **posterior estimate of the relative frequency with which $X = 1$.**

Posterior belief
after seeing 8
heads in 10 trials

Prior belief

\downarrow \downarrow
 $beta(f; 3,3)$ $beta(f; 11,5)$



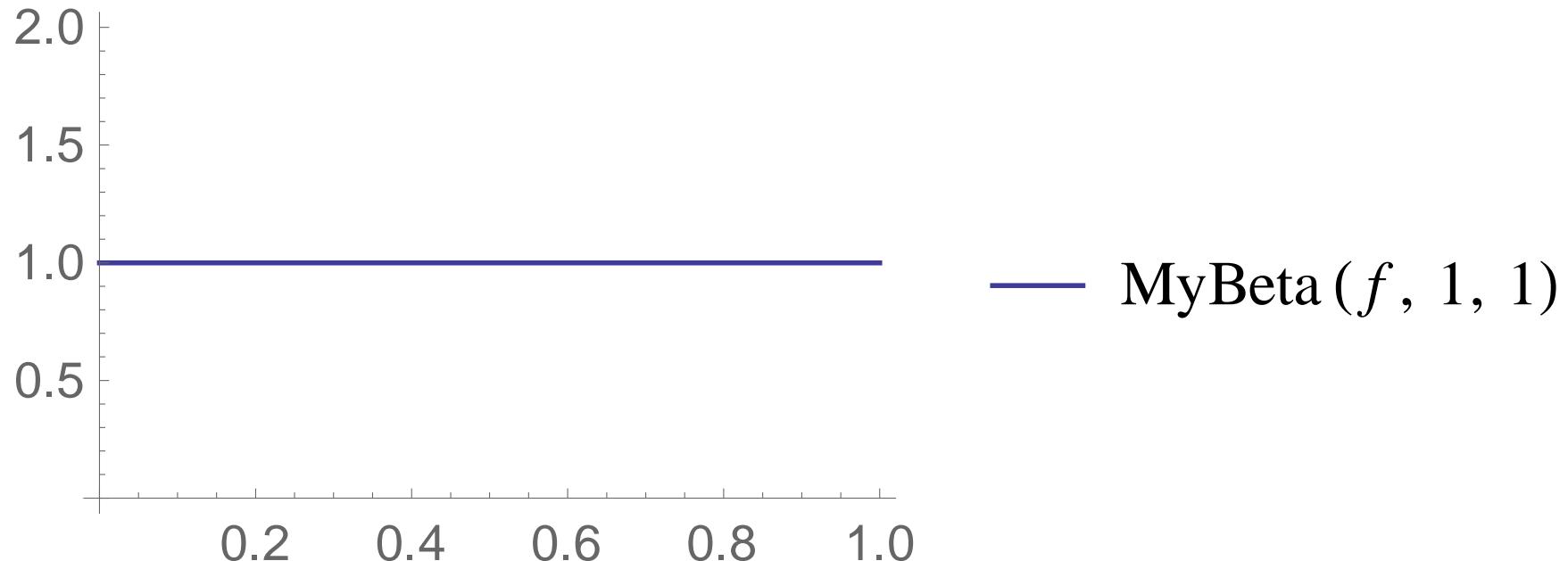
$P(X=1|f)=f \quad P(X=1|f)=f$

Assessing Values of a and b (1)

$a = b = 1$:

All numbers in $[0,1]$ are equally likely to be relative frequency f with which the binary random variable X assumes its value of “1”

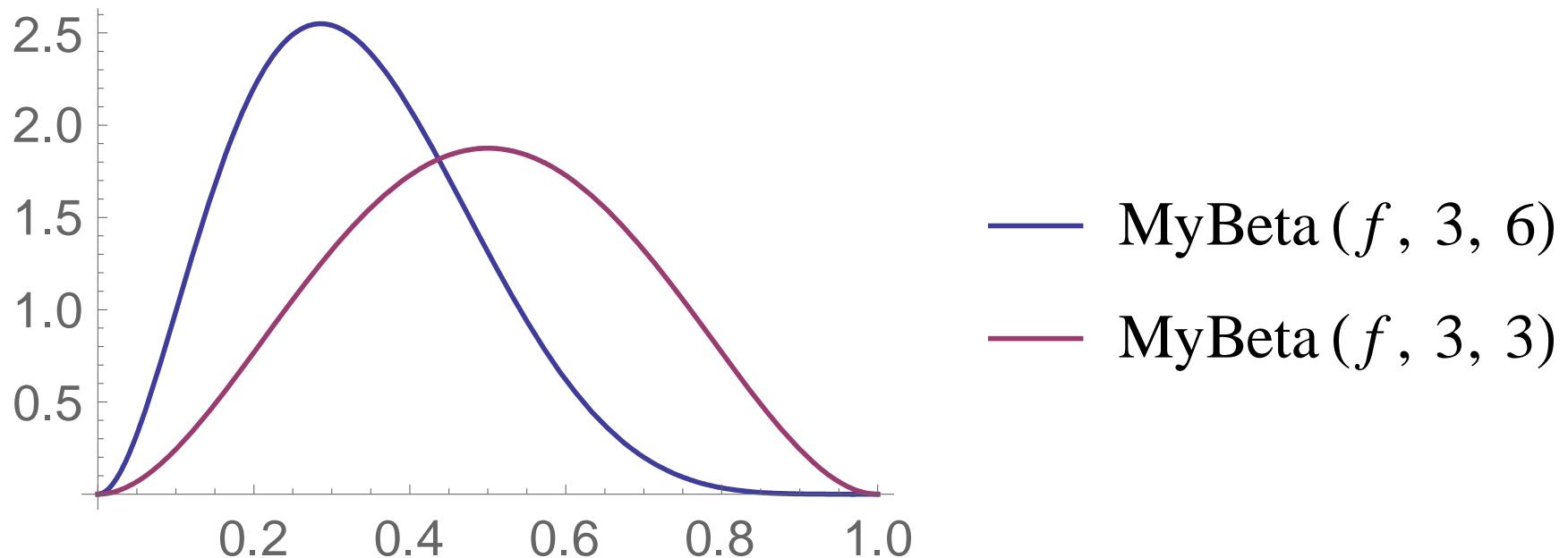
- Use in lack of knowledge or
- For objectivity (no subjective beliefs)



Assessing Values of a and b (2)

$a, b > 1$:

Believe that relative frequency of “1” is most probable around $a/(a + b)$. The larger a and b , the more certain we are.



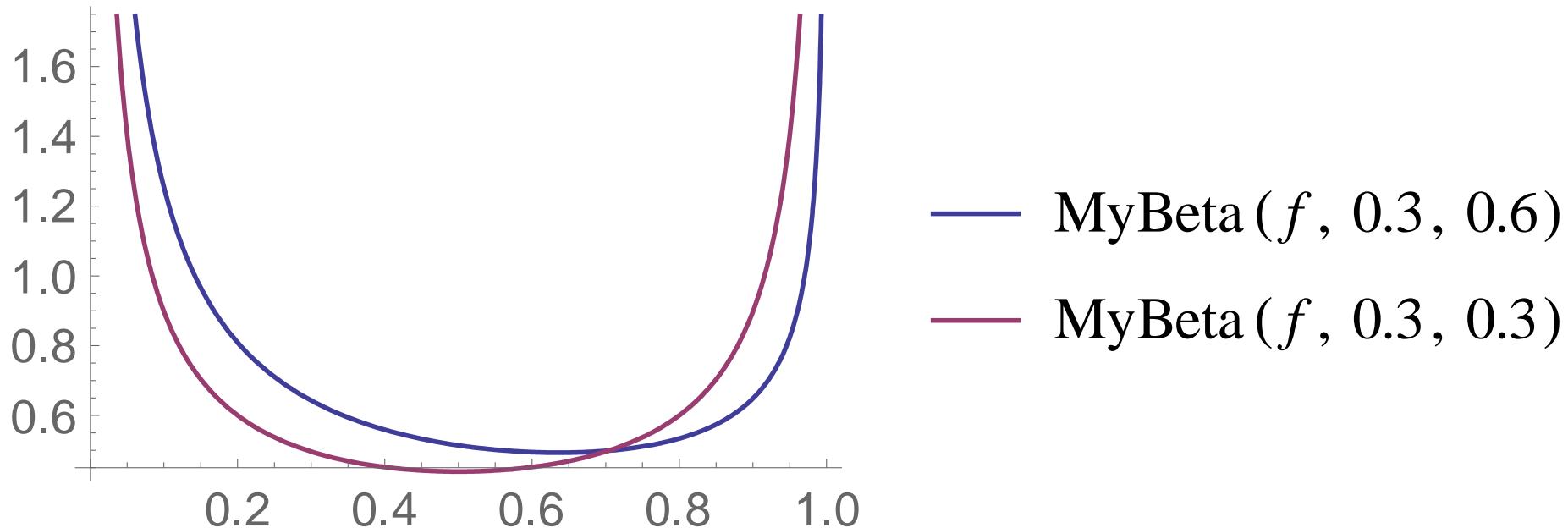
Assessing Values of a and b (3)

$a, b < 1$:

Belief that relative frequency is more likely to be at the extremes (i.e., 0 or 1). A relative frequency is most unlikely around $a/(a + b)$.

$a, b \approx 0$:

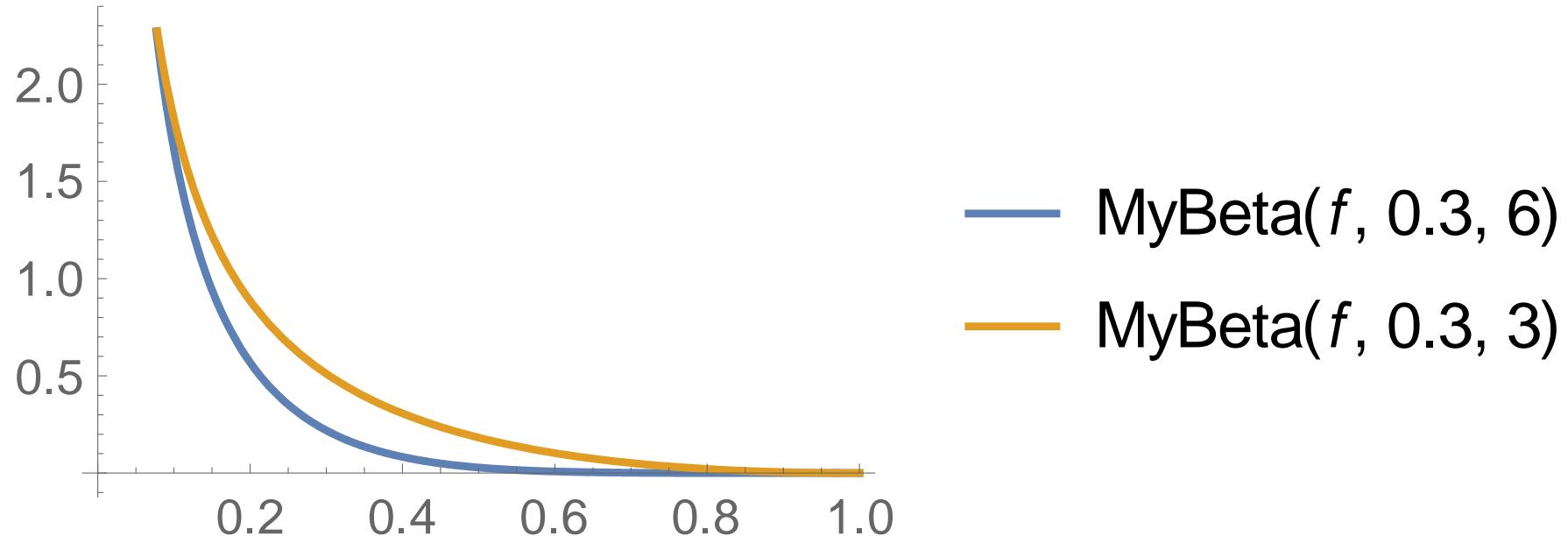
With almost certainty f is 0 or 1.



Assessing Values of a and b (4)

$a < 1, b > 1$:

Belief that relative frequency with which X assumes “1” value is low.



Assessing Values of a and b (5)

Technique 1

- a, b are integers. Performing N trials and seeing a times “1”

Technique 2

- Assess probability for the 1st trial.
- Assess conditional probability for 2nd trial is “1” given the 1st one is a “1”.

$$\Rightarrow \frac{a}{a+b} = \hat{P}(X^{(1)} = 1) \quad \text{and} \quad \frac{a+1}{a+b+1} = \hat{P}(X^{(2)} = 1 | X^{(1)} = 1)$$

⇒ Instructive example in book on page 303 (Example 6.10)

p% Probability Interval for $E(F)$

Solve the following equation for c :

$$\int_{E(F)-c}^{E(F)+c} \rho(f) df = p$$

Case 1:

$(E(F) - c, E(F) + c) \subseteq (0, 1)$:

A p% probability interval is given by $(E(F) - c, E(F) + c)$.

Case 2:

$(E(F) - c, E(F) + c) \not\subseteq (0, 1)$ and $E(F) > .5$:

Solve the following equation for c :

$$\int_{E(F)-c}^1 \rho(f) df = p$$

A p% probability interval is given by $(E(F) - c, 1)$.

Case 3:

$(E(F) - c, E(F) + c) \not\subseteq (0, 1)$ and $E(F) < .5$:

Solve the following equation for c :

$$\int_0^{E(F)+c} \rho(f) df = p$$

A p% probability interval is given by $(0, E(F) + c)$.

Example:

$$a=11, b=5 \implies \rho(f|d) = \text{beta}(f; 11, 5).$$

$$\text{Therefore we have: } E(F|d) = \frac{11}{11+5} = .688$$

The 95% probability interval can be determined by

$$\begin{aligned} \int_{.688-c}^{.688+c} \frac{\Gamma(16)}{\Gamma(11)\Gamma(5)} f^{10} (1-f)^4 df &= .95 \\ \Rightarrow (.668 - .214, .688 + .214) &= (.474, .902) \end{aligned}$$

Approximation (1)

Sometime we only know the expected value and the variance of $\rho(f)$

⇒ Use normal distribution to approximate the unknown original distribution.

- Compute $p\%$ probability interval for $E(F)$ using the normal distribution:
- The normal approximation to a $p\%$ probability interval for F is given by

$$(E(F) - z_p \sigma(F), E(F) + z_p \sigma(F))$$

where

p	z_p
80	1.28
95	1.96
99	2.58

Lemma 6.7: Suppose the random variable F has the $\text{beta}(f; a, b)$ density function. Then

$$E(F^2) = \left(\frac{a+1}{a+b+1} \right) \left(\frac{a}{a+b} \right)$$

Approximation (2)

Example:

$$a=11, b=5$$

$$\Rightarrow E(F) = \frac{11}{16} = .688$$

$$\Rightarrow E(F^2) = \left(\frac{11+1}{11+5+1}\right)\left(\frac{11}{11+5}\right) = \frac{33}{68}$$

$$\Rightarrow Var(F) = E(F^2) - [E(F)]^2 = \frac{33}{68} - \left(\frac{11}{16}\right)^2 = .012638$$

$$\Rightarrow \sigma(F) = .112$$

\Rightarrow The 95% probability interval can be determined as

$$(.688 - (1.96)(.112), .688 + (1.96)(.112)) = (.468, .908)$$

The normal distribution function becomes a better approx. of $beta(f; a, b)$ as a and b become larger and as they become closer to being equal. If a and b are each at least 5, it is usually a reasonable approximation.

Learning All Binary Parameters in a Bayesian Network

Augmented Bayesian Network (1)

Goal: Extending theory for learning from one to all parameters in a BN.

Definition 6.8: An ***augmented Bayesian network*** (\mathbb{G}, F, ρ) is a Bayesian network determined by the following:

1. A DAG $\mathbb{G} = (V, E)$ where $V = \{X_1, \dots, X_n\}$ and each X_i is a random variable.
2. For every i , an auxiliary parent variable F_i of X_i and a density function ρ_i of F_i . Each F_i is a root and has no edge to any variable except X_i . The set of all F_i s is denoted by F . That is

$$F = F_1 \cup F_2 \cup \dots \cup F_n.$$

3. For every i , for all values pa_i of the parents PA_i in V of X_i , and all values f_i of F_i , a probability distribution of X_i conditional on pa_i and f_i .

An augmented BN is simply a BN. **The notation is the only difference!**

⇒ Encodes our beliefs concerning the unknown conditional relative frequencies (parameters) needed for the DAG.

F_i is a set of random variables representing our belief concerning the relative frequencies of the values of X_i given values of the parents of X_i .

Binomial Augmented Bayesian Network

Definition 6.9: A *binomial augmented Bayesian* network $(\mathbb{G}, \mathcal{F}, \rho)$ is an augmented Bayesian network with the following properties:

1. For every i , X_i has space $\{1,2\}$.
2. For every i , there is an ordering $[\text{pa}_{i1}, \text{pa}_{i2}, \dots, \text{pa}_{iq_i}]$ of all instantiations of the parents PA_i in \mathcal{V} of X_i , where q_i is the number of different instantiations of these parents. Furthermore, for every i ,

$$\mathcal{F}_i = \{F_{i1}, F_{i2}, \dots, F_{iq_i}\}$$

where each F_{ij} is a root, has no edge to any variable except X_i , and has density function

$$\rho_{ij}(f_{ij}) \quad 0 \leq f_{ij} \leq 1.$$

3. For every i and j , and all values of $f_i = \{f_{i1}, \dots, f_{ij}, \dots, f_{iq_i}\}$ of \mathcal{F}_i ,

$$P(X_i = 1 | \text{pa}_{ij}, f_{i1}, \dots, f_{ij}, \dots, f_{iq_i}) = f_{ij}.$$

If X_i is a root, PA_i is empty. In this case, $q_i = 1$ and $P(X_i = 1 | f_{i1}) = f_{i1}$.

F_{ij} is a random variable whose probability distribution represents our belief concerning the relative frequency with which X_i is equal to 1 given that the parents of X_i are in their j th instantiation.

Augmented Bayesian Network (2)

1. Since all F_i s are root nodes, they are mutually independent. Therefore, we have **global parameter independence**:

$$\rho(f_1, f_2, \dots, f_n) = \rho_1(f_1)\rho_2(f_2) \cdots \rho_n(f_n)$$

Subscripting both ρ and f creates clutter. Thus we write

$$\rho(f_1, f_2, \dots, f_n) = \rho(f_1)\rho(f_2) \cdots \rho(f_n)$$

It's clear from the subscript on f which density function each ρ represents.

2. Since F_{ij} are all roots in the BN, they are mutually independent. Therefore, we have **local parameter independence** of their members F_{ij}

$$\rho(f_{i1}, f_{i2}, \dots, f_{iq_i}) = \rho(f_{i1})\rho(f_{i2}) \cdots \rho(f_{iq_i}) \quad \text{for } 1 \leq i \leq n$$

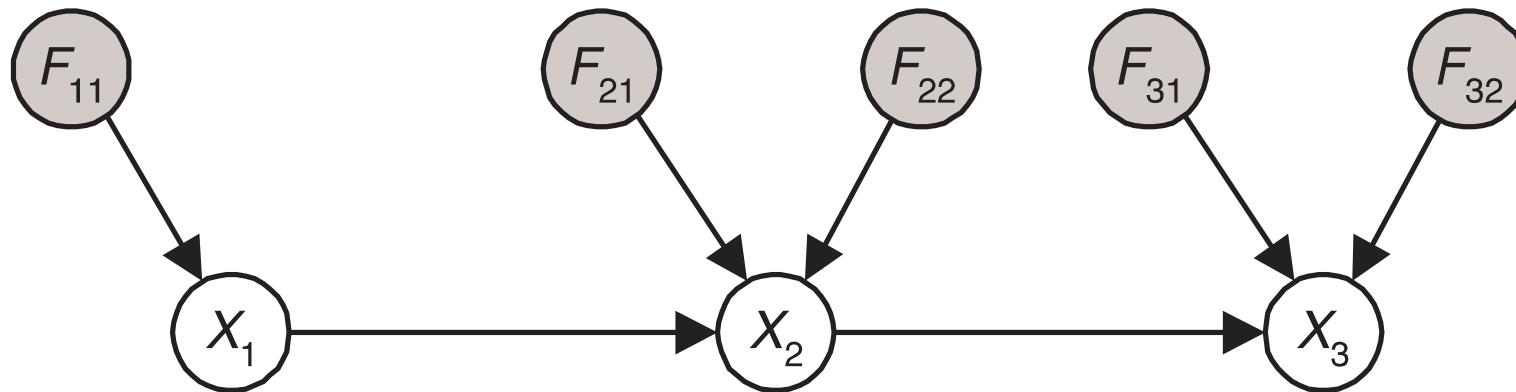
3. Together with global and local parameter independence we get:

$$\rho(f_{11}, f_{12}, \dots, f_{nq_n}) = \rho(f_{11})\rho(f_{12}) \cdots \rho(f_{nq_n})$$

Theorem 6.6: Let an augmented Bayesian network (\mathbb{G}, F, ρ) be given. Then the marginal distribution of P of $\{X_1, \dots, X_n\}$ constitutes a Bayesian network with \mathbb{G} . We say (\mathbb{G}, F, ρ) embeds (\mathbb{G}, P) .

Example: Figure 6.20

$\text{beta}(f_{11}; 8,2)$ $\text{beta}(f_{21}; 2,6)$ $\text{beta}(f_{22}; 1,1)$ $\text{beta}(f_{31}; 2,1)$ $\text{beta}(f_{32}; 3,4)$

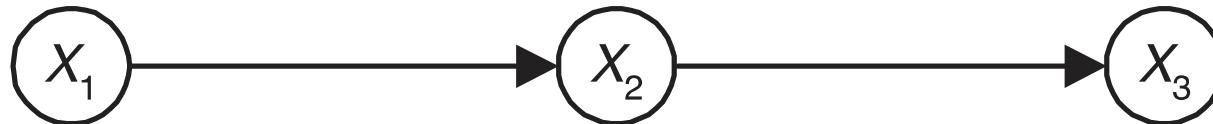


(a)

$$P(X_1 = 1) = 4/5$$

$$\begin{aligned} P(X_2 = 1 | X_1 = 1) &= 1/4 \\ P(X_2 = 1 | X_1 = 2) &= 1/2 \end{aligned}$$

$$\begin{aligned} P(X_3 = 1 | X_2 = 1) &= 2/3 \\ P(X_3 = 1 | X_2 = 2) &= 3/7 \end{aligned}$$



(b)

Fig. 6.20: A binomial augmented BN is in (a), and its embedded BN is in (b).

Theorem 6.7

Theorem 6.7: Let a binomial augmented Bayesian network (\mathbb{G}, F, ρ) be given. Then for each i and each j , the ij -th conditional distribution in the embedded Bayesian network (\mathbb{G}, P) is given by

$$P(X_i = 1 | pa_{ij}) = E(F_{ij})$$

If each F_{ij} has a beta distribution with parameters $a_{ij}, b_{ij}, N_{ij} = a_{ij} + b_{ij}$, then for each i and each j the ij -th conditional distribution in the embedded network (\mathbb{G}, P) is given by

$$P(X_i = 1 | pa_{ij}) = \frac{a_{ij}}{N_{ij}}$$

Note: Inference is always done in the embedded BN using only the variables in V .

Bayesian Network Sample

Definition 6.10: Suppose we have a sample of size M as follows:

1. We have the random vectors

$$\mathbf{X}^{(1)} = \begin{pmatrix} X_1^{(1)} \\ \vdots \\ X_n^{(1)} \end{pmatrix}, \dots, \mathbf{X}^{(M)} = \begin{pmatrix} X_1^{(M)} \\ \vdots \\ X_n^{(M)} \end{pmatrix} \quad \text{with } D = \{\mathbf{X}^{(1)}, \dots, \mathbf{X}^{(M)}\}$$

such that, for every i , each $X_i^{(h)}$ has the same space.

2. There is an augmented Bayesian network (\mathbb{G}, F, ρ) , where $\mathbb{G} = (V, E)$, such that, for $1 \leq h \leq M$

$$\{X_1^{(h)}, \dots, X_n^{(h)}\}$$

constitutes an instance of V in \mathbb{G} resulting in a distinct augmented Bayesian network.

Then the sample D is called a **Bayesian network sample** of size M with parameter (\mathbb{G}, F) .

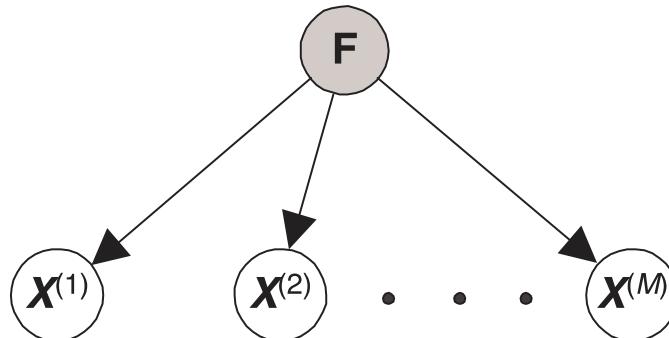
Definition 6.11: Suppose we have a Bayesian network sample of size M such that

1. for every i , each $X_i^{(h)}$ has the space $\{1, 2\}$;
2. its augmented Bayesian network (\mathbb{G}, F, ρ) is binomial.

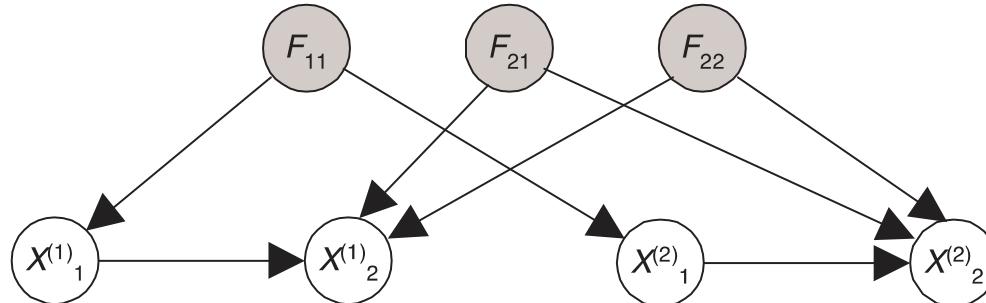
Then the sample D is called a **binomial Bayesian network sample** of size M with parameter (\mathbb{G}, F) .

Note: A binomial sample (Def. 6.3) is a binomial BN sample in which \mathbb{G} contains only one node.

Example of an BN Sample



(a)



(b)

Figure 6.21: The high-level structure of a BN sample is given by the DAG in (a). In that DAG, each node and arc actually represents a set of nodes and arcs respectively. The detailed structure in the case of a binomial augmented BN sample when $m = n = 2$ is shown in (b).

Lemma 6.8: Suppose

1. D is a Bayesian network sample of size M with parameter (\mathbb{G}, F) ;

Lemma 6.10: Suppose

1. D is a **binomial** Bayesian network sample of size M with parameter (\mathbb{G}, F) ;

2. we have a set of values (data) of the $\mathbf{X}^{(h)}$ as follows:

$$\mathbf{x}^{(1)} = \begin{pmatrix} x_1^{(1)} \\ \vdots \\ x_n^{(1)} \end{pmatrix}, \dots, \mathbf{x}^{(M)} = \begin{pmatrix} x_1^{(M)} \\ \vdots \\ x_n^{(M)} \end{pmatrix} \text{ with } \mathbf{d} = \{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(M)}\}.$$

Then

$$P(\mathbf{d}|f_1, \dots, f_n) = \prod_{i=1}^n \prod_{h=1}^M P(x_i^{(h)} | \text{pa}_i^{(h)}, f_i),$$

where $\text{pa}_i^{(h)}$ contains the values of the parents of X_i in the h th case.

Then

$$\begin{aligned} P(\mathbf{d}|f_1, \dots, f_n) &= P(\mathbf{d}|f_{11}, \dots, f_{nq_n}) \\ &= \prod_{i=1}^n \prod_{j=1}^{q_i} (f_{ij})^{s_{ij}} (1 - f_{ij})^{t_{ij}} \end{aligned}$$

where M_{ij} is the number of $\mathbf{x}^{(h)}$'s in which $X_i^{(h)}$'s parents are in their j th instantiation, and of these M_{ij} cases, s_{ij} is the number in which $x_i^{(h)}$ is equal to 1 and t_{ij} is the number in which it equals 2.

P(d)

Theorem 6.8: Suppose we have the conditions in Lemma 6.8. Then

$$P(d) = \prod_{i=1}^n \left(\int_{f_i} \prod_{h=1}^M P(x_i^{(h)} | \text{pa}_i^{(h)}, f_i) \rho(f_i) df_i \right)$$

Theorem 6.11: Suppose we have the conditions in Lemma 6.10. Then

$$P(d) = \prod_{i=1}^n \prod_{j=1}^{q_i} E(F_{ij}^{s_{ij}} [1 - F_{ij}]^{t_{ij}})$$

If additionally each F_{ij} has a beta distribution with the parameters $a_{ij}, b_{ij}, N_{ij} = a_{ij} + b_{ij}$. Then

$$\begin{aligned} P(d) \\ = \prod_{i=1}^n \prod_{j=1}^{q_i} \frac{\Gamma(N_{ij})}{\Gamma(N_{ij} + M_{ij})} \frac{\Gamma(a_{ij} + s_{ij})\Gamma(b_{ij} + t_{ij})}{\Gamma(a_{ij})\Gamma(b_{ij})} \end{aligned}$$

Posterior Global Parameter Independence

Theorem 6.9: (*Posterior Global Parameter Independence*) Suppose we have the conditions in Lemma 6.8. Then the F_i s are mutually independent conditional on D. That is,

$$\rho(f_1, \dots, f_n | d) = \prod_{i=1}^n \rho(f_i | d)$$

Theorem 6.12: (*Posterior Global Parameter Independence*) Suppose we have the conditions in Lemma 6.10. Then the F_{ij} s are mutually independent conditional on D. That is,

$$\rho(f_{11}, f_{12}, \dots, f_{nq_n} | d) = \prod_{i=1}^n \prod_{j=1}^{q_i} \rho(f_{ij} | d)$$

with

$$\rho(f_{ij} | d) = \frac{(f_{ij})^{s_{ij}} (1 - f_{ij})^{t_{ij}} \rho(f_{ij})}{E(F_{ij}^{s_{ij}} [1 - F_{ij}]^{t_{ij}})}$$

If $\rho(f_{ij}) = \text{beta}(f_{ij}; a_{ij}, b_{ij})$ then

$$\rho(f_{ij} | d) = \text{beta}(f_{ij}; a_{ij} + s_{ij}, b_{ij} + t_{ij})$$

P(d|f₁, ..., f_n) and P(d)

Lemma 6.10: Suppose

1. D is a **binomial** Bayesian network sample of size M with parameter (\mathbb{G}, F);
2. we have a set of values (data) of the $\mathbf{X}^{(h)}$ as follows:

$$\mathbf{x}^{(1)} = \begin{pmatrix} x_1^{(1)} \\ \vdots \\ x_n^{(1)} \end{pmatrix}, \dots, \mathbf{x}^{(M)} = \begin{pmatrix} x_1^{(M)} \\ \vdots \\ x_n^{(M)} \end{pmatrix} \quad \text{with} \quad d = \{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(M)}\}.$$

3. Then

$$P(d|f_1, \dots, f_n) = P(d|f_{11}, \dots, f_{nq_n}) = \prod_{i=1}^n \prod_{j=1}^{q_i} (f_{ij})^{s_{ij}} (1 - f_{ij})^{t_{ij}}$$

where M_{ij} is the number of $\mathbf{x}^{(h)}$'s in which $X_i^{(h)}$'s parents are in their j th instantiation, and of these M_{ij} cases, s_{ij} is the number in which $x_i^{(h)}$ is equal to 1 and t_{ij} is the number in which it equals 2.

Theorem 6.11: Suppose we have the conditions in Lemma 6.10. Also each F_{ij} has a beta distribution with the parameters $a_{ij}, b_{ij}, N_{ij} = a_{ij} + b_{ij}$, i.e. $\rho(f_{ij}) = \text{beta}(f_{ij}; a_{ij}, b_{ij})$. Then

$$P(d) = \prod_{i=1}^n \prod_{j=1}^{q_i} \frac{\Gamma(N_{ij})}{\Gamma(N_{ij} + M_{ij})} \frac{\Gamma(a_{ij} + s_{ij}) \Gamma(b_{ij} + t_{ij})}{\Gamma(a_{ij}) \Gamma(b_{ij})}$$

Posterior Global Parameter Independence

Theorem 6.12: (*Posterior Global Parameter Independence*) Suppose we have the conditions in Lemma 6.10 and each F_{ij} has a beta distribution with the parameters $a_{ij}, b_{ij}, N_{ij} = a_{ij} + b_{ij}$, i.e., $(f_{ij}) = \text{beta}(f_{ij}; a_{ij}, b_{ij})$. Then the F_{ij} s are mutually independent conditional on D. That is,

$$\rho(f_{11}, f_{12}, \dots, f_{nq_n} | d) = \prod_{i=1}^n \prod_{j=1}^{q_i} \rho(f_{ij} | d)$$

with

$$\rho(f_{ij} | d) = \text{beta}(f_{ij}; a_{ij} + s_{ij}, b_{ij} + t_{ij}).$$

Theorem 6.10

Definition 6.12: The augmented Bayesian network $(\mathbb{G}, \mathcal{F}, \rho|d)$ with $\rho|d$ denoting $\rho(f_1, \dots, f_n|d)$ is called the ***updated augmented Bayesian network*** relative to the Bayesian network sample D and the data d. The network it embeds is called the ***updated embedded Bayesian network*** relative to the Bayesian network sample D and the data d.

Theorem 6.10: Suppose the conditions in Lemma 6.8 hold, and we create a Bayesian network sample of size $M+1$ by including another random vector

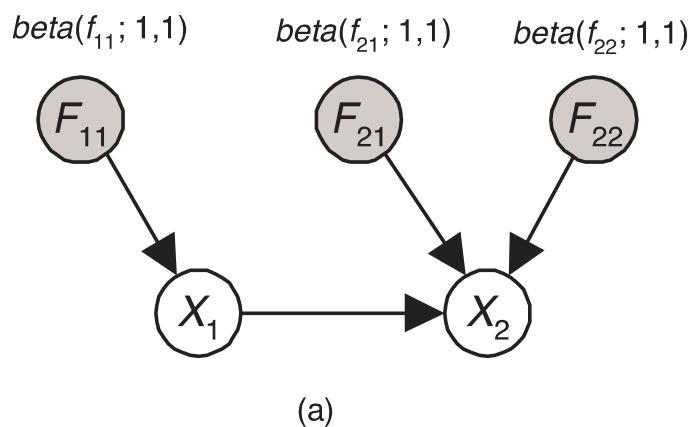
$$X^{(M+1)} = \begin{pmatrix} X_1^{(M+1)} \\ \vdots \\ X_n^{(M+1)} \end{pmatrix}$$

Then if D is the Bayesian network sample of size M , the updated distribution

$$P(x_1^{(M+1)}, \dots, x_n^{(M+1)}|d)$$

is the probability distribution in the updated embedded Bayesian network.

Example 6.20 (1)



$$\begin{array}{ccc}
 X_1 & \xrightarrow{\hspace{1cm}} & X_2 \\
 P(X_1 = 1) = 1/2 & & P(X_2 = 1 | X_1 = 1) = 1/2 \\
 & & P(X_2 = 1 | X_1 = 2) = 1/2
 \end{array}$$

Example 6.20: Suppose we have a binomial Bayesian network sample whose parameter is the augmented Bayesian network in the Figure left. Suppose further that we obtain the data (values of X_1 and X_2) on 8 individuals (cases) shown in Table 6.2.

Counting yields $s_{11} = 3$, $t_{11} = 5$, $s_{21} = 1$, $t_{21} = 2$, $s_{22} = 3$, and $t_{22} = 2$. From Figure left, we see for all i and j that $a_{ij} = b_{ij} = 1$.

Therefore, we get

$$P(d) = \left(\frac{\Gamma(2)}{\Gamma(2+8)} \frac{\Gamma(1+3)\Gamma(1+5)}{\Gamma(1)\Gamma(1)} \right) \left(\frac{\Gamma(2)}{\Gamma(2+3)} \frac{\Gamma(1+1)\Gamma(1+2)}{\Gamma(1)\Gamma(1)} \right) \left(\frac{\Gamma(2)}{\Gamma(2+5)} \frac{\Gamma(1+3)\Gamma(1+2)}{\Gamma(1)\Gamma(1)} \right)$$

Case	X_1	X_2
1	1	2
2	1	1
3	2	1
4	2	2
5	2	1
6	2	1
7	1	2
8	2	2

Table 6.2: Data on 8 cases

Example 6.20 (2)

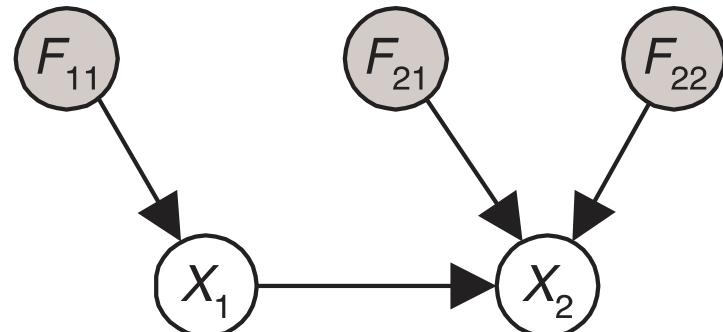
Counting yields

$$\begin{aligned}s_{11} &= 3, t_{11} = 5, \\ s_{21} &= 1, t_{21} = 2, \\ s_{22} &= 3, t_{22} = 2.\end{aligned}$$

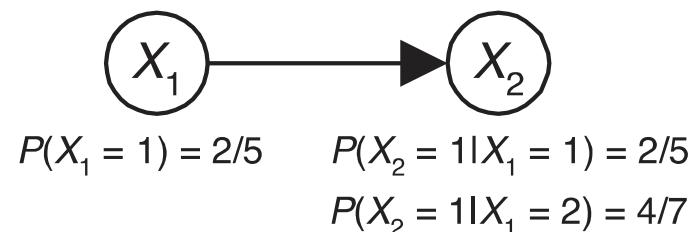
The updated network is shown to the right:

$$\begin{aligned}\rho(f_{11}|d) &= \text{beta}(f_{11}; 1 + 3, 1 + 5) = \text{beta}(f_{11}; 4, 6) \\ \rho(f_{21}|d) &= \text{beta}(f_{21}; 1 + 1, 1 + 2) = \text{beta}(f_{21}; 2, 3) \\ \rho(f_{22}|d) &= \text{beta}(f_{22}; 1 + 3, 1 + 2) = \text{beta}(f_{22}; 4, 3).\end{aligned}$$

$$\begin{aligned}\text{beta}(f_{11}; 4, 6) \\ \text{beta}(f_{21}; 2, 3) \\ \text{beta}(f_{22}; 4, 3)\end{aligned}$$



(c)



(d)

Sample Size Problem (1)

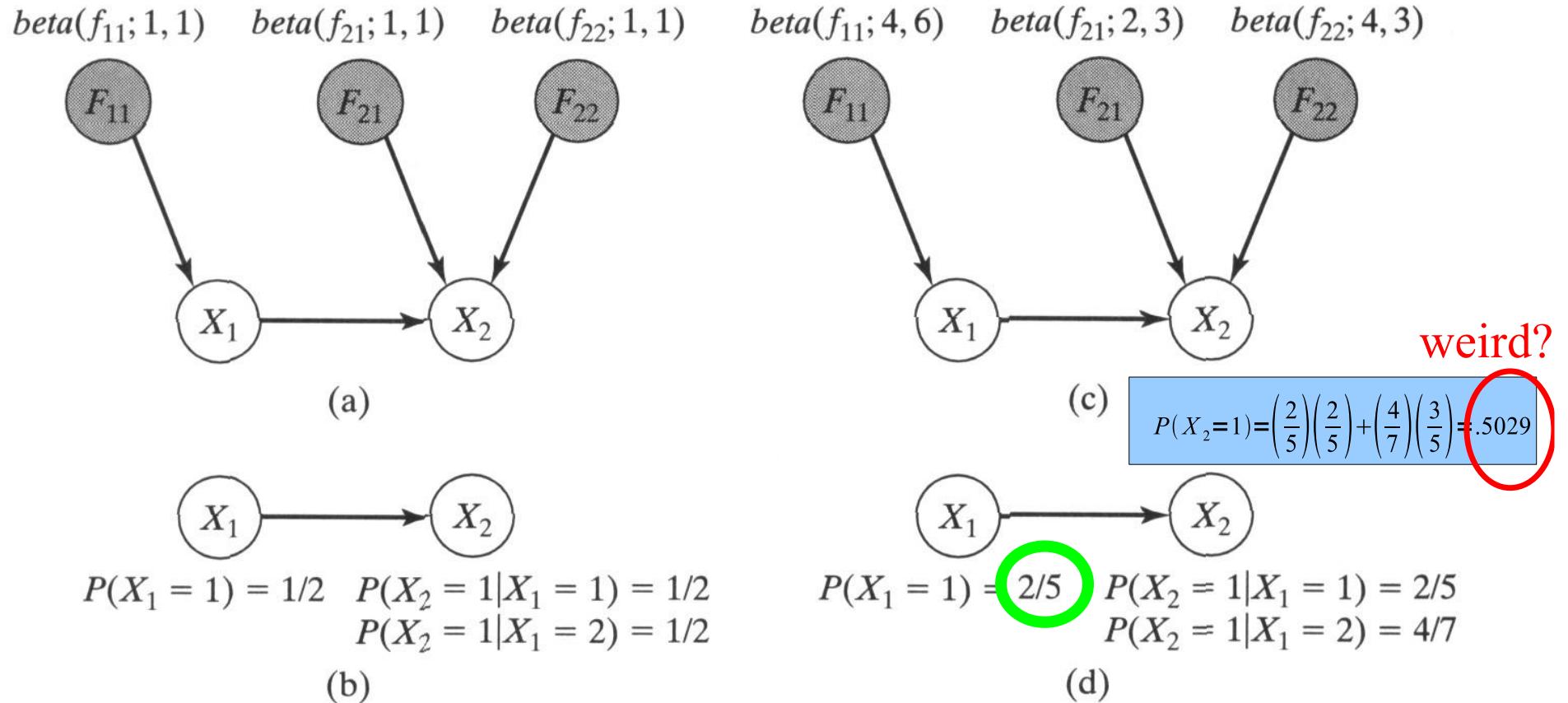
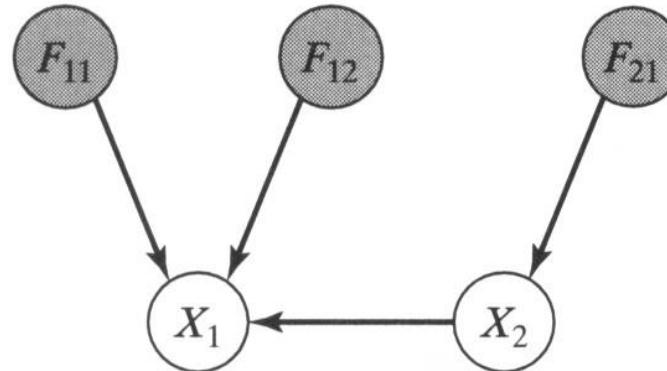


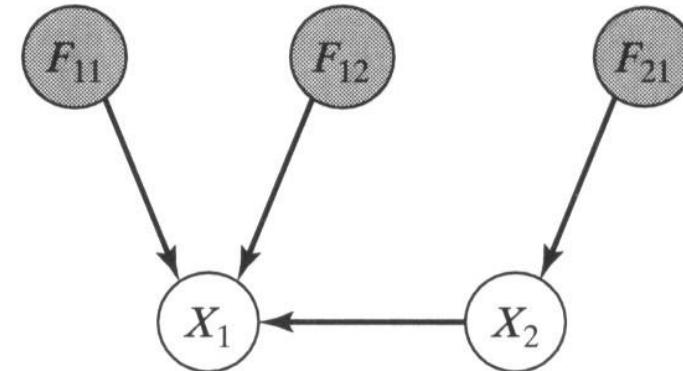
Figure 6.22: An augmented Bayesian network is shown in (a) and its embedded Bayesian network is shown in (b). Updated networks are shown in (c) and (d).

Sample Size Problem (2)

$\text{beta}(f_{11}; 1, 1) \quad \text{beta}(f_{12}; 1, 1) \quad \text{beta}(f_{21}; 1, 1) \quad \text{beta}(f_{11}; 2, 4) \quad \text{beta}(f_{12}; 3, 3) \quad \text{beta}(f_{21}; 5, 5)$



(a)



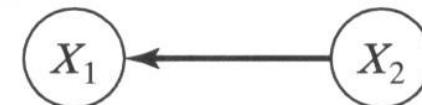
(c)

$$P(X_1=1) = \left(\frac{1}{3}\right)\left(\frac{1}{2}\right) + \left(\frac{1}{2}\right)\left(\frac{1}{2}\right) = \left(\frac{5}{12}\right)$$



$$\begin{aligned} P(X_1 = 1 | X_2 = 1) &= 1/2 & P(X_2 = 1) &= 1/2 \\ P(X_1 = 1 | X_2 = 2) &= 1/2 \end{aligned}$$

(b)



$$\begin{aligned} P(X_1 = 1 | X_2 = 1) &= 1/3 & P(X_2 = 1) &= 1/2 \\ P(X_1 = 1 | X_2 = 2) &= 1/2 \end{aligned}$$

The two graphs are equivalent,
how can the results be different?

Figure 6.23: An augmented Bayesian network is shown in (a) and its embedded Bayesian network is shown in (b). Updated networks are shown in (c) and (d).

Sample Size Problem (3)

So far:

- Our result depend on the which equivalent DAG we use to represent the independencies!
- BUT: The result should not depend on which equivalent DAG we use!
- CAUSE: Mixed sample size

Equivalent Sample Size (1)

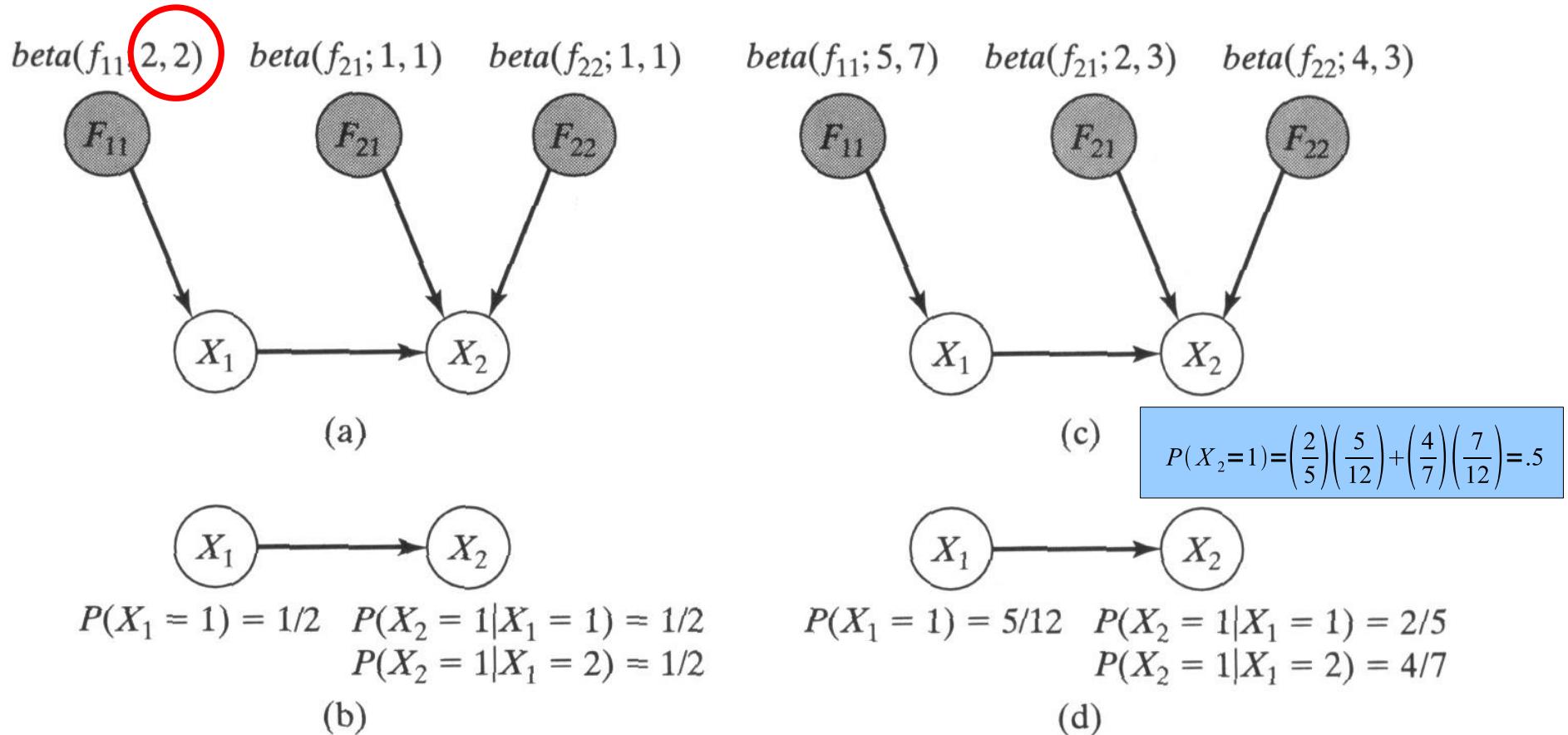
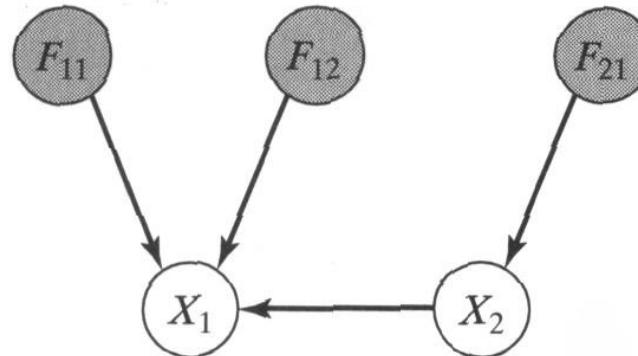


Figure 6.24: An augmented Bayesian network is shown in (a) and its embedded Bayesian network is shown in (b). Updated networks are shown in (c) and (d).

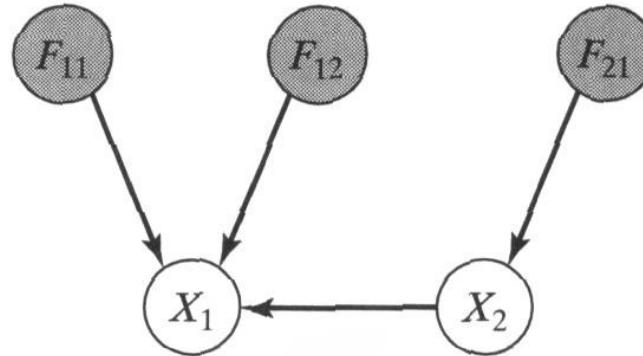
Equivalent Sample Size (2)

$\text{beta}(f_{11}; 1, 1) \quad \text{beta}(f_{12}; 1, 1) \quad \text{beta}(f_{21}; 2, 2)$



(a)

$\text{beta}(f_{11}; 2, 4) \quad \text{beta}(f_{12}; 3, 3) \quad \text{beta}(f_{21}; 6, 6)$



(c)

$$P(X_1=1) = \left(\frac{1}{3}\right)\left(\frac{1}{2}\right) + \left(\frac{1}{2}\right)\left(\frac{1}{2}\right) = \left(\frac{5}{12}\right)$$



$$\begin{aligned} P(X_1 = 1 | X_2 = 1) &= 1/2 & P(X_2 = 1) &= 1/2 \\ P(X_1 = 1 | X_2 = 2) &= 1/2 \end{aligned}$$

(b)

$$\begin{aligned} P(X_1 = 1 | X_2 = 1) &= 1/3 & P(X_2 = 1) &= 1/2 \\ P(X_1 = 1 | X_2 = 2) &= 1/2 \end{aligned}$$

(d)

Figure 6.25: An augmented Bayesian network is shown in (a) and its embedded Bayesian network structure is shown in (b). Updated networks are shown in (c) and (d).

)

Equivalent Sample Size (3)

The idea is that we specify values for a_{ij} and b_{ij} that can actually occur in a sample exhibiting the conditional independencies entailed by the DAG.

Definition 6.13: Suppose we have a binomial augmented Bayesian network in which the density functions are $\text{beta}(f_{ij}; a_{ij}, b_{ij})$ for all i and j . If there is a number N such that, for all i and j

$$N_{ij} = a_{ij} + b_{ij} = P(pa_{ij}) \times N$$

then the network is said to have equivalent sample size N . In case of a root, PA_i is empty and $q_i = 1$ ($P(pa_i) = 1$).

For all nodes X_i we have:

$$\sum_{j=1}^{q_i} N_{ij} = \sum_{j=1}^{q_i} [P(pa_{ij}) \times N] = N \times \sum_{j=1}^{q_i} P(pa_{ij}) = N$$

Equivalent Sample Size (4)

Theorem 6.13: Suppose we specify G , F , and N and assign for all i and j

$$a_{ij} = b_{ij} = \frac{N}{2q_i}$$

Then the resultant augmented Bayesian network has equivalent sample size N , and the probability distribution in the resultant embedded BN is uniform.

Example: Figure 6.24 and Figure 6.25 using $N = 4$.

Theorem 6.14: Suppose we specify G , F , and N , a BN (G, P) , and assign for all i and j

$$\begin{aligned} a_{ij} &= P(X_i = 1 | pa_{ij}) \times P(pa_{ij}) \times N \\ b_{ij} &= P(X_i = 2 | pa_{ij}) \times P(pa_{ij}) \times N \end{aligned}$$

Then the resultant augmented BN has equivalent sample size N , and it embeds the originally specified BN.

Example: Figure 6.20 using $N = 10$.

Equivalent Sample Size (5)

Definition 6.14: Binomial augmented BNs $(\mathbb{G}_1, F^{\mathbb{G}_1}, \rho | \mathbb{G}_1)$ and $(\mathbb{G}_2, F^{\mathbb{G}_2}, \rho | \mathbb{G}_2)$ are called *equivalent* if they satisfy the following:

1. \mathbb{G}_1 and \mathbb{G}_2 are Markov equivalent.
2. The probability distributions in their embedded BNs are the same.
3. The specified density functions in both are beta.
4. They have the same equivalent sample size.

Note: $\rho | \mathbb{G}$ denotes the density function in the augmented BN containing DAG \mathbb{G} , and not a ρ conditioned on \mathbb{G} .

Theorem 6.15: Suppose we have two equivalent binomial augmented Bayesian networks $(\mathbb{G}_1, F^{\mathbb{G}_1}, \rho | \mathbb{G}_1)$ and $(\mathbb{G}_2, F^{\mathbb{G}_2}, \rho | \mathbb{G}_2)$. Let D be a set of random vectors as specified in Definition 6.11. Then, given any set d of values of the vectors in D , the updated embedded Bayesian network relative to D and the data d , obtained by considering D a binomial BN sample with parameter $(\mathbb{G}_1, F^{\mathbb{G}_1})$, contains the same probability distribution as the one obtained by considering D a binomial BN sample with parameter $(\mathbb{G}_2, F^{\mathbb{G}_2})$.

⇒ As long as we use an equivalent sample size, our updated probability distribution does not depend on which equivalent DAG we use to represent a set of conditional independencies.

Expressing Prior Indifference

Use $N=2$

- Root nodes

$$\text{beta}(f; 1,1)$$

- Other nodes with q_i different combinations of parent values

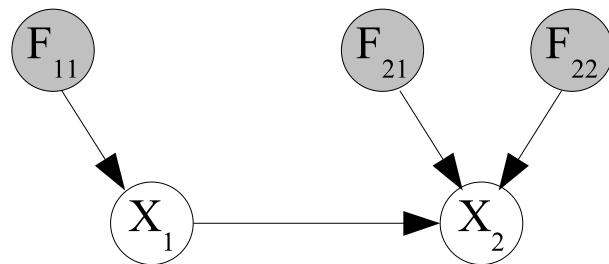
$$\text{beta}(f; \frac{1}{q_i}, \frac{1}{q_i})$$

In this way, the total “sample” size at each node is always 2.

Learning with Randomly Missing Data

Learning with Data

$\text{beta}(f_{11}; 2, 2)$ $\text{beta}(f_{21}; 1, 1)$ $\text{beta}(f_{22}; 1, 1)$



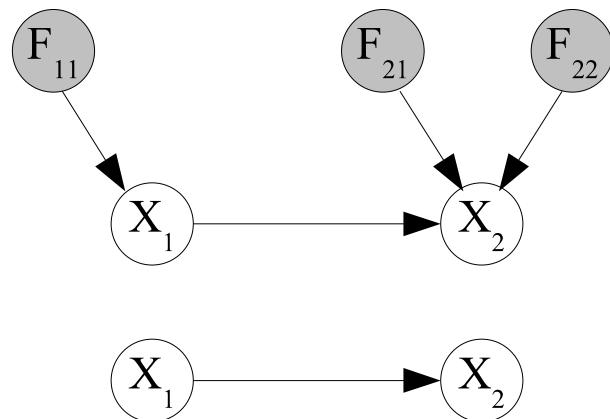
$$P(X_1=1)=1/2$$

$$\begin{aligned} P(X_2=1|X_1=1) &= 1/2 \\ P(X_2=1|X_1=2) &= 1/2 \end{aligned}$$

Case	X1	X2
1	1	1
2	1	1
3	1	1
4	1	2
5	2	2

$$\begin{aligned} s_{11} &= 4 \\ t_{11} &= 1 \\ s_{21} &= 3 \\ t_{21} &= 1 \\ s_{22} &= 0 \\ t_{22} &= 1 \end{aligned}$$

$\text{beta}(f_{11}; 6, 3)$ $\text{beta}(f_{21}; 4, 2)$ $\text{beta}(f_{22}; 1, 2)$

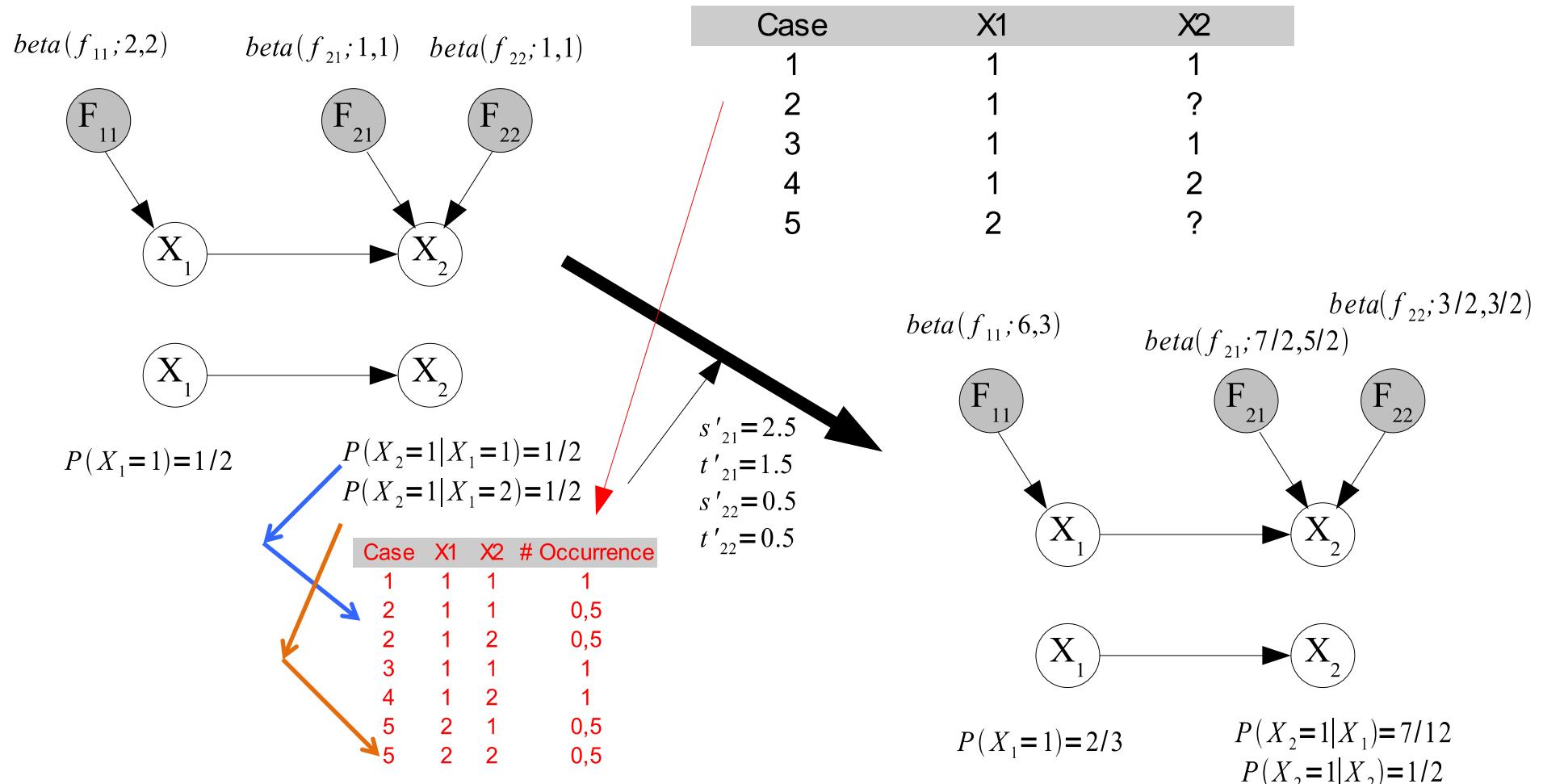


$$\begin{aligned} P(X_1=1) &= 2/3 \\ P(X_2=1|X_1=1) &= 2/3 \\ P(X_2=1|X_1=2) &= 1/3 \end{aligned}$$

Learning with Missing Data

Assumption: Data items are missing randomly.

⇒ Fill in with currently know conditional probabilities of $P(X_2 = 1|X_1 = 1)$ and $P(X_2 = 2|X_1 = 1)$, respectively.



What We Computed

Set $f' = \{f'_{11}, f'_{21}, f'_{22}\} = \{f_{11}, f_{21}, f_{22}\} = f = \{1/2, 1/2, 1/2\}$

$$\begin{aligned}\Rightarrow s'_{21} &= E(s_{21}|d, f') = \sum_{h=1}^5 1 \times P(X_1^{(h)} = 1, X_2^{(h)} = 1|d, f') \\ &= \sum_{h=1}^5 P(X_1^{(h)} = 1, X_2^{(h)} = 1|x^{(h)}, f') \\ &= \sum_{h=1}^5 P(X_1^{(h)} = 1, X_2^{(h)} = 1|x_1^{(h)}, x_2^{(h)}, f') \\ &= 1 + \frac{1}{2} + 1 + 0 + 0 = \frac{5}{2} \\ \Rightarrow t'_{21} &= E(t_{21}|d, f') = 0 + \frac{1}{2} + 0 + 1 + 0 = \frac{3}{2}\end{aligned}$$

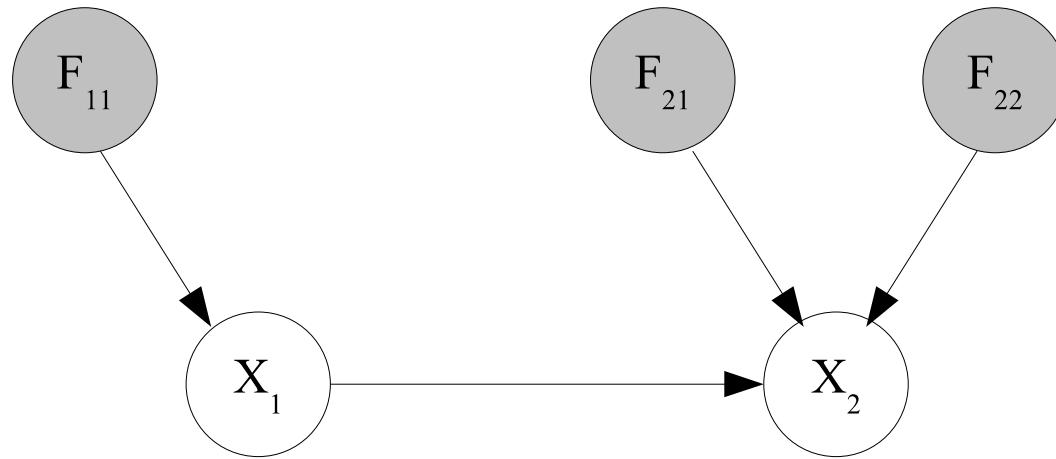
⇒ Estimates are **based only on the 'data' in our prior sample**. They are not based on the data d.

Recall

$$\text{beta}(f_{11}; 6, 3)$$

$$\text{beta}(f_{21}; 7/2, 5/2)$$

$$\text{beta}(f_{22}; 3/2, 3/2)$$



$$P(X_1 = 1) = 2/3$$

$$P(X_2 = 1 | X_1) = 7/12$$

$$P(X_2 = 1 | X_2) = 1/2$$

What We Should Compute

Incorporate the data d in our estimates:

$$f' = \{f'_{11}, f'_{21}, f'_{22}\} = \{2/3, 7/12, 1/2\}$$

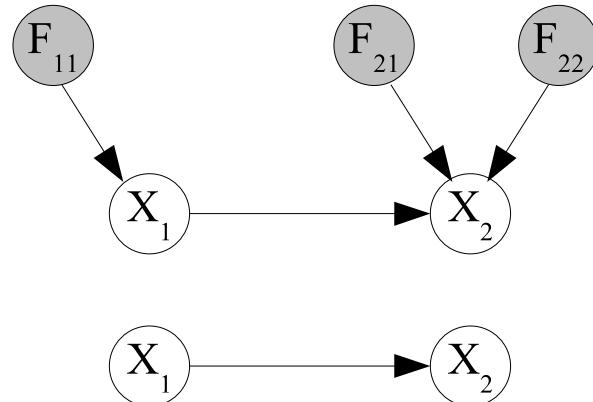
$$\begin{aligned}\Rightarrow s'_{21} &= E(s_{21}|d, f') = \sum_{h=1}^5 1 \times P(X_1^{(h)} = 1, X_2^{(h)} = 1 | d, f') \\ &= \sum_{h=1}^5 P(X_1^{(h)} = 1, X_2^{(h)} = 1 | x^{(h)}, f') \\ &= \sum_{h=1}^5 P(X_1^{(h)} = 1, X_2^{(h)} = 1 | x_1^{(h)}, x_2^{(h)}, f') \\ &= 1 + \frac{7}{12} + 1 + 0 + 0 = 2 \frac{7}{12} \\ \Rightarrow t'_{21} &= E(t_{21}|d, f') = 0 + \frac{5}{12} + 0 + 1 + 0 = 1 \frac{5}{12}\end{aligned}$$

Repeat until convergence.

⇒ Under certain conditions the limit that is approached by f' is a value of f that **locally maximizes $\rho(f|d)$** .

Recall

$$\text{beta}(f_{11}; 6, 3) \quad \text{beta}(f_{21}; \frac{7}{2}, \frac{5}{2}) \quad \text{beta}(f_{22}; \frac{3}{2}, \frac{3}{2})$$



$$P(X_1=1)=2/3$$

$$P(X_2=1|X_1=1)=7/12$$

$$P(X_2=1|X_1=2)=1/2$$

Case	X1	X2	# Occurrence
1	1	1	1
2	1	1	7/12
2	1	2	5/12
3	1	1	1
4	1	2	1
5	2	1	0,5
5	2	2	0,5

Case	X1	X2
1	1	1
2	1	?
3	1	1
4	1	2
5	2	?

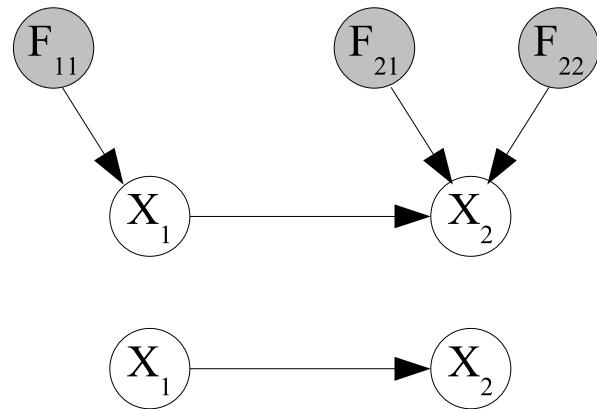
$$s'_{21} = 2 \frac{7}{12}$$

$$t'_{21} = 1 \frac{5}{12}$$

$$s'_{22} = 0,5$$

$$t'_{22} = 0,5$$

$$\text{beta}(f_{11}; 6, 3) \quad \text{beta}(f_{21}; \frac{43}{12}, \frac{29}{12}) \quad \text{beta}(f_{22}; \frac{3}{2}, \frac{3}{2})$$



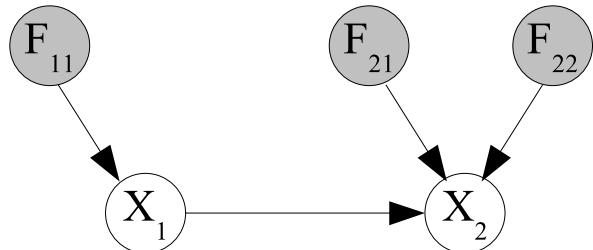
$$P(X_1=1)=2/3$$

$$P(X_2=1|X_1=1)=31/48$$

$$P(X_2=1|X_1=2)=1/2$$

Repeat Update

$$\text{beta}(f_{11}; 6, 3) \quad \text{beta}(f_{21}; \frac{43}{12}, \frac{29}{12}) \quad \text{beta}(f_{22}; \frac{3}{2}, \frac{3}{2})$$



$$P(X_1=1)=2/3$$

$$P(X_2=1|X_1=1)=43/72$$

$$P(X_2=1|X_1=2)=1/2$$

Case	X1	X2	# Occurrence
1	1	1	1
2	1	1	43 / 72
2	1	2	29 / 72
3	1	1	1
4	1	2	1
5	2	1	0,5
5	2	2	0,5

Case	X1	X2
1	1	1
2	1	?
3	1	1
4	1	2
5	2	?

$$s'_{21} = 2 \frac{43}{72}$$

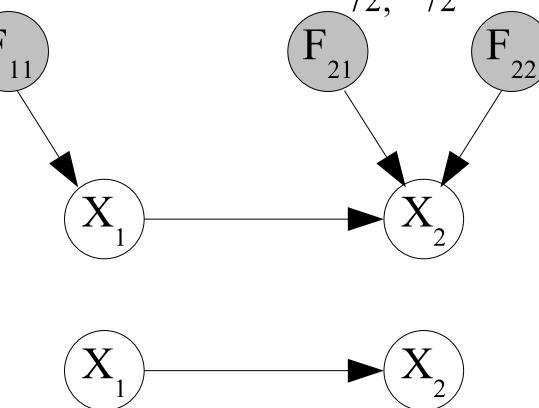
$$t'_{21} = 1 \frac{29}{72}$$

$$s'_{22} = 0.5$$

$$t'_{22} = 0.5$$

$$\text{beta}(f_{11}; 6, 3)$$

$$\text{beta}(f_{21}; 3 \frac{43}{72}, 2 \frac{29}{72}) \quad \text{beta}(f_{22}; \frac{3}{2}, \frac{3}{2})$$



$$P(X_1=1)=2/3$$

$$P(X_2=1|X_1=1)=?$$

$$P(X_2=1|X_1=2)=1/2$$

Algorithm 6.1 (1)

Algorithm 6.1 EM-MAP-determination

Problem: Given a binomial augmented Bayesian network in which the density functions are beta, and data d containing some incomplete data items, estimate $\rho(f|d)$ and the MAP value of the parameter set f .

Inputs: binomial augmented Bayesian network (G, F, ρ) and data d containing some incomplete data items.

Outputs: estimate $\rho(f|d')$ of $\rho(f|d)$ and estimate f' of the MAP value of the parameter set f .

```
void EM_MAP (augmented-Bayesian-network (G, F, rho),
             data d,
             int k,                                // number of
             density-fuction& rho(f|d'),          // iterations
             MAP-estimate& f')
```

Algorithm 6.1 (2)

```

{
  float s'_{ij}, t'_{ij} ;

  for (i = 1; i <= n; i++)
    for (j = 1; j <= q_i; j++)
      assign f'_{ij} a value in the interval (0, 1);
  repeat (k times) {
    for (i = 1; i <= n; i++) // expectation
      for (j = 1; j <= q_i; j++) { // step
        s'_{ij} = E(s_{ij} | d, f') = \sum_{h=1}^M P(X_i^{(h)} = 1, pa_{ij} | x^{(h)}, f');
        t'_{ij} = E(t_{ij} | d, f' = \sum_{h=1}^M P(X_i^{(h)} = 2, pa_{ij} | x^{(h)}, f');
      }
    for (i = 1; i <= n; i++) // maximiza-
      for (j = 1; j <= q_i; j++) // tion step
        f'_{ij} = \frac{a_{ij} + s'_{ij}}{a_{ij} + s'_{ij} + b_{ij} + t'_{ij}};
  }
  \rho(f_{ij} | d') = beta(f_{ij}; a_{ij} + s'_{ij}, b_{ij} + t'_{ij});
}

```

SS 2014 – Bayesian Networks

Parameter Learning: Multinomial Variables

*University of Augsburg
Multimedia Computing and Computer Vision,
Prof. Dr. Rainer Lienhart
Rainer.Lienhart@informatik.uni-augsburg.de
www.multimedia-computing.org*

Reference

Richard E. Neapolitan. **Learning Bayesian Networks.** *Prentice Hall Series in Artificial Intelligence*, ISBN 0-13-012534-2.

Don't forget. Reading the book chapters 1 – 6 is mandatory.

Chapter on ***More Parameter Learning***
(chapter 7)

Figures and text are taken from that book

Dirichlet Density Function

Model:

1. Given is some r -outcome random process.
2. Let X be a random variable whose space $\{1, 2, \dots, r\}$ contains the outcome of the experiments.
3. For $1 \leq k \leq r$:

Let F_k be a random variable whose space is the interval $[0,1]$ and the probability distribution of F_k represents our belief concerning the relative frequency with which $X = k$, i.e.,

$$P(X = k | f_k) = f_k$$

Definition 7.1: The **Dirichlet density function** with parameters $a_1, a_2, \dots, a_r, N = \sum_{k=1}^r a_k$, where a_1, a_2, \dots, a_r are integers ≥ 1 , is defined as

$$\rho(f_1, \dots, f_{r-1}) = \frac{\Gamma(N)}{\prod_{k=1}^r \Gamma(a_k)} f_1^{a_1-1} f_2^{a_2-1} \cdots f_r^{a_r-1} \quad 0 \leq f_k \leq 1, \sum_{k=1}^r f_k = 1.$$

Random variables F_1, F_2, \dots, F_r , that have this density function are said to have a **Dirichlet distribution**. We refer to this function as

$$Dir(f_1, f_2, \dots, f_{r-1}; a_1, a_2, \dots, a_r)$$

Interpretation:

The probability experience is equivalent to having seen the k th value occur a_k times in N trials.

Representing Belief about Relative Frequency

Lemma 7.1: If F_1, F_2, \dots, F_r have a Dirichlet distribution with parameters $a_1, a_2, \dots, a_r, N = \sum_{k=1}^r a_k$ then for $1 \leq k \leq r$:

$$E(F_k) = \frac{a_k}{N}$$

Theorem 7.1: Suppose X is a random variable with space $\{1, 2, \dots, r\}$, and F_1, F_2, \dots, F_r are r random variables such that for all k

$$P(X = k | f_k) = f_k$$

Then

$$P(X = k) = E(F_k)$$

If F_1, F_2, \dots, F_r have a Dirichlet distribution with parameters $a_1, a_2, \dots, a_r, N = \sum_{k=1}^r a_k$ then

$$P(X = k) = \frac{a_k}{N}$$

Definition 7.2: Suppose we have a sample of size M such that

1. Each $X^{(h)}$ has a space $\{1, 2, \dots, r\}$;
2. $F = \{F_1, F_2, \dots, F_r\}$, and for $1 \leq h \leq M$ and $1 \leq k \leq r$:

$$P(X^{(h)} = k | f_1, \dots, f_k, \dots, f_r) = f_k$$

Then D is called a **multinomial sample** of size M with parameter F .

Learning a Relative Frequency (1)

Theorem 7.2+: Suppose

1. D is a multinomial sample of size M with parameter F;
2. we have a set of values $d = \{x^{(1)}, x^{(2)}, \dots, x^{(M)}\}$ of the variables in D (d = data set)
3. s_k is the number of variables in d equal to k
4. F_1, F_2, \dots, F_r have a Dirichlet distribution with parameters $a_1, a_2, \dots, a_r, N = \sum_{k=1}^r a_k$. That is,

$$\rho(f_1, f_2, \dots, f_{r-1}) = Dir(f_1, f_2, \dots, f_{r-1}; a_1, a_2, \dots, a_r)$$

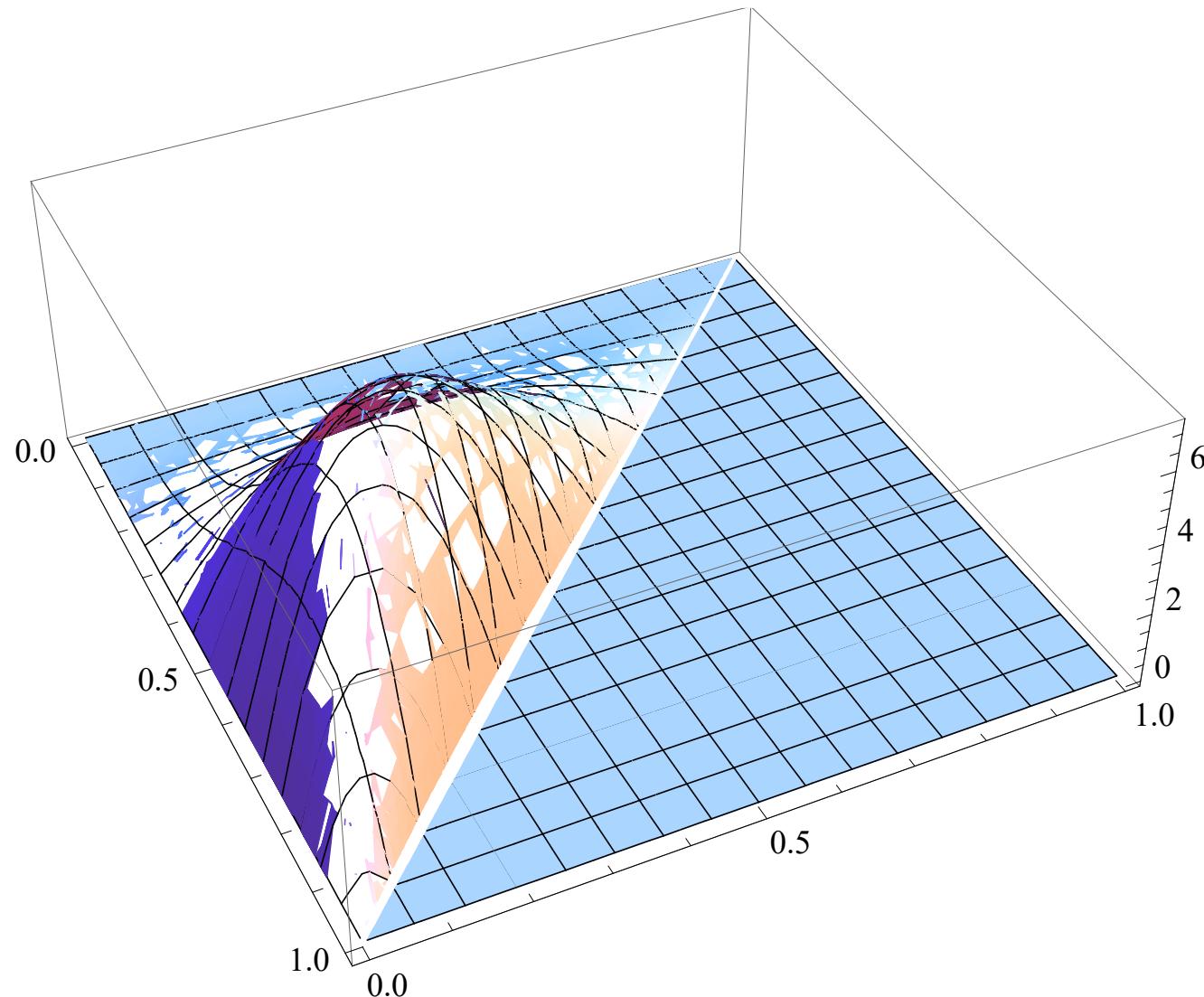
Then

$$P(d) = \frac{\Gamma(N)}{\Gamma(N+M)} \prod_{k=1}^r \frac{\Gamma(a_k + s_k)}{\Gamma(a_k)}$$
$$\rho(f_1, f_2, \dots, f_{r-1}|d) = Dir(f_1, f_2, \dots, f_{r-1}; a_1 + s_1, a_2 + s_2, \dots, a_r + s_k)$$

If we create a multinomial sample of size $M + 1$ by adding another variable $X^{(M+1)}$ to D, then, for all k

$$P(X^{(M+1)} = k|d) = E(F_k|d) = \frac{a_k + s_k}{N + M}$$

Example $Dir(f_1, f_2; 4, 2, 2)$



Guidelines for Assessing the Values of a_k

- $a_1 = a_2 = \dots = a_r = 1$:

These values mean that we consider all combinations of relative frequencies that sum to 1 equally probable. Used if we

- have no knowledge at all concerning the values of the relative frequencies

or

- want to be objective

⇒ we impose none of our beliefs concerning our relative frequencies on the learning algorithm except that we know that at most r things can happen

- $a_1 = a_2 = \dots = a_r > 1$:

These values mean that we feel it more probable that the relative frequency of the k th value is around a_k/N . The larger the values of a_k , the more we believe this.

- $a_1 = a_2 = \dots = a_r < 1$:

These values mean that we feel that the relative frequencies result in not having many different things happening are more probable.

Multinomial Augmented Bayesian Network

Definition 6.9: A *multinomial augmented Bayesian* network $(\mathbb{G}, \mathcal{F}, \rho)$ is an augmented Bayesian network with the following properties:

1. For every i , X_i has space $\{1, 2, \dots, r_i\}$.
2. For every i , there is an ordering $[\text{pa}_{i1}, \text{pa}_{i2}, \dots, \text{pa}_{iq_i}]$ of all instantiations of the parents PA_i in \mathcal{V} of X_i , where q_i is the number of different instantiations of these parents. Furthermore, for every i ,

$$F_i = F_{i1} \cup F_{i2} \cup \dots \cup F_{iq_i},$$

where

$$F_{ij} = \{F_{ij1}, F_{ij2}, \dots, F_{ijr_i}\}$$

and each F_{ij} is a root, has no edge to any variable except X_i , and has density function

$$\rho_{ij}(f_{ij}) = \rho(f_{ij1}, \dots, f_{ij(r_i-1)}) \quad 0 \leq f_{ijk} \leq 1, \sum_{k=1}^{r_i} f_{ijk} = 1.$$

3. For every i, j and k , and all values $f_{i1}, \dots, f_{ij}, \dots, f_{iq_i}$ of $F_{i1}, \dots, F_{ij}, \dots, F_{iq_i}$,

$$P(X_i = k | \text{pa}_{ij}, f_{i1}, \dots, f_{ij}, \dots, f_{iq_i}) = f_{ijk}.$$

If X_i is a root, PA_i is empty. In this case, $q_i = 1$ and $P(X_i = k | f_{i1}) = f_{i1k}$.

F_{ijk} is a random variable whose probability distribution represents our belief concerning the relative frequency with which X_i is equal to k given that the parents of X_i are in their j th instantiation.

Theorem 7.5

Again, we have global parameter independence of the sets F_i as well as local parameter independence of their subsets F_{ij} 's since they are all roots in a BN and thus mutually independent.

A Binomial augmented BN is a multinomial augmented BN in which $r_i = 2$ for all i .

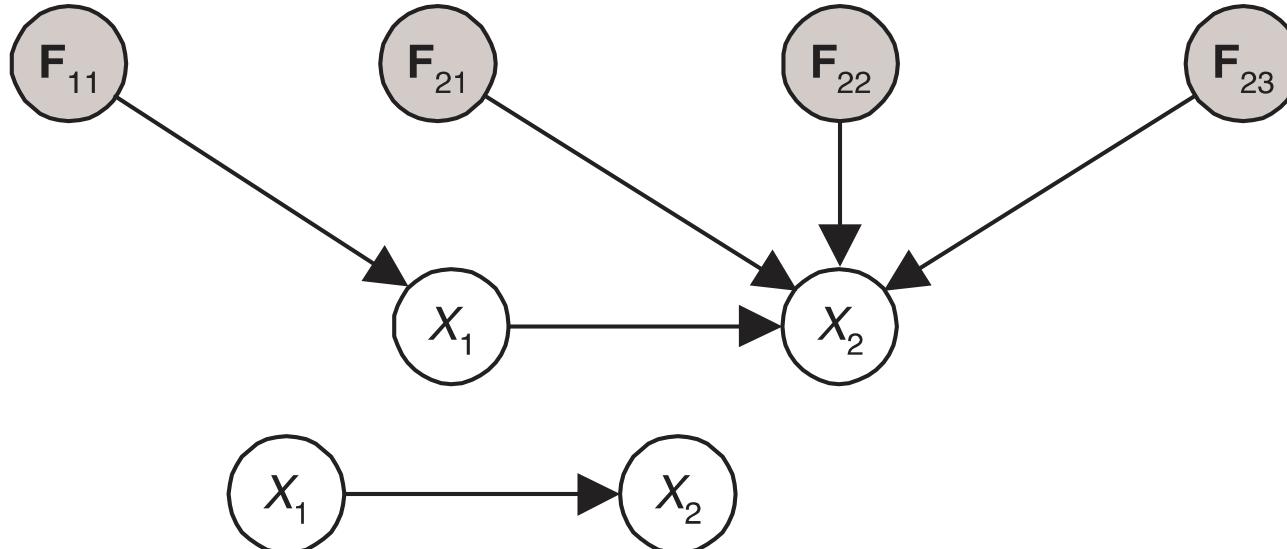
Theorem 7.5: Let a multinomial augmented Bayesian network (\mathbb{G}, F, ρ) be given, where the variables in each F_{ij} have a Dirichlet distribution with parameters $a_{ij1}, a_{ij2}, \dots, a_{ijr_i}, N_{ij} = \sum_k a_{ijk}$. Then for each i and each j , the ijk th conditional distribution in the embedded Bayesian network (\mathbb{G}, P) is given by

$$P(X_i = k | pa_{ij}) = E(F_{ijk}) = \frac{a_{ijk}}{N_{ij}}$$

Note: Inference is always done in the embedded BN using only the variables in V .

Example Figure 7.6

$Dir(f_{111}, f_{112}; 4, 8, 10) \quad Dir(f_{211}, f_{212}, f_{213}; 1, 1, 1, 1) \quad Dir(f_{221}, f_{222}, f_{223}; 2, 4, 1, 1) \quad Dir(f_{231}, f_{232}, f_{233}; 1, 3, 4, 2)$



$$P(X_1 = 1) = 2/11 \quad P(X_2 = 1 | X_1 = 1) = 1/4$$

$$P(X_1 = 2) = 4/11 \quad P(X_2 = 2 | X_1 = 1) = 1/4$$

$$P(X_1 = 3) = 5/11 \quad P(X_2 = 3 | X_1 = 1) = 1/4$$

$$P(X_2 = 4 | X_1 = 1) = 1/4 \quad P(X_2 = 1 | X_1 = 3) = 1/10$$

$$P(X_2 = 2 | X_1 = 3) = 3/10 \quad P(X_2 = 2 | X_1 = 3) = 3/10$$

$$P(X_2 = 3 | X_1 = 3) = 2/5 \quad P(X_2 = 3 | X_1 = 3) = 2/5$$

$$P(X_2 = 4 | X_1 = 3) = 1/5 \quad P(X_2 = 4 | X_1 = 3) = 1/5$$

$$P(X_2 = 1 | X_1 = 2) = 1/4$$

$$P(X_2 = 2 | X_1 = 2) = 1/2$$

$$P(X_2 = 3 | X_1 = 2) = 1/8$$

$$P(X_2 = 4 | X_1 = 2) = 1/8$$

Learning Using a Multinomial Augmented BN (1)

Definition 7.4: Suppose we have a Bayesian network sample of size M such that

4. for every i , each $X_i^{(h)}$ has the space $\{1, 2, \dots, r_i\}$;
5. its augmented Bayesian network (\mathbb{G}, F, ρ) is multinomial.

Then the sample D is called a **multinomial Bayesian network sample** of size M with parameter (\mathbb{G}, F) .

Theorem 7.6: Suppose

1. D is a **multinomial** Bayesian network sample of size M with parameter (\mathbb{G}, F) .
2. We have a set of values (data) of the $\mathbf{X}^{(h)}$ as follows:

$$\mathbf{x}^{(1)} = \begin{pmatrix} x_1^{(1)} \\ \vdots \\ x_n^{(1)} \end{pmatrix}, \dots, \mathbf{x}^{(M)} = \begin{pmatrix} x_1^{(M)} \\ \vdots \\ x_n^{(M)} \end{pmatrix} \quad \text{with} \quad d = \{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(M)}\}.$$

3. M_{ij} is the number of $\mathbf{x}^{(h)}$'s in which $X_i^{(h)}$'s parents are in their j th instantiation, and of these M_{ij} cases, s_{ijk} is the number in which $x_i^{(h)}$ is equal to k .
4. The variables in each F_{ij} have a Dirichlet distribution with the parameters $a_{ij1}, a_{ij2}, \dots, a_{ijr_i}$, $N_{ij} = \sum_k a_{ijk}$, i.e. $\rho(f_{ij}) = Dir(f_{ij1}, f_{ij2}, \dots, f_{ij(r_i-1)}; a_{ij1}, a_{ij2}, \dots, a_{ijr_i}) \quad \forall i, j$. Then

$$P(d) = \prod_{i=1}^n \prod_{j=1}^{q_i} \frac{\Gamma(N_{ij})}{\Gamma(N_{ij} + M_{ij})} \prod_{k=1}^{r_i} \frac{\Gamma(a_{ijk} + s_{ijk})}{\Gamma(a_{ijk})}$$

Learning Using a Multinomial Augmented BN (2)

Theorem 7.7: (*Posterior Local Parameter Independence*) Suppose we have the conditions in Theorem 7.6. Then the F_i s are mutually independent conditional on D. That is,

$$\rho(f_{11}, f_{12}, \dots, f_{nq_n} | d) = \prod_{i=1}^n \prod_{j=1}^{q_i} \rho(f_{ij} | d)$$

with

$$\rho(f_{ij} | d) = Dir(f_{ij1}, f_{ij2}, \dots, f_{ij(r_i-1)}; a_{ij1} + s_{ij1}, a_{ij2} + s_{ij2}, \dots, a_{ijr_i} + s_{ijr_i})$$

Equivalent Sample Size

Definition 7.5: Suppose we have a multinomial augmented Bayesian network in which the density functions are $\text{Dir}(f_{ij1}, f_{ij2}, \dots, f_{ij(r_i-1)}; a_{ij1}, a_{ij2}, \dots, a_{ijr_i})$ or all i and j . If there is a number N such that, for all i and j

$$N_{ij} = \sum_{k=1}^{r_i} a_{ijk} = P(pa_{ij}) \times N$$

then the network is said to have **equivalent sample size** N . In case of a root, PA_i is empty and $q_i = 1$ ($P(pa_i) = 1$).

Theorem 7.8+: Suppose we specify \mathbb{G} , F , and N and

1. assign for all i, j and k

$$a_{ijk} = \frac{N}{r_i q_i}$$

Then the resultant augmented Bayesian network has equivalent sample size N , and the probability distribution in the resultant embedded BN is uniform.

2. assign for all i and j

$$a_{ijk} = P(X_i = k | pa_{ij}) \times P(pa_{ij}) \times N$$

Then the resultant augmented Bayesian network has equivalent sample size N , and it embeds the originally specified BN.

Expressing Prior Indifference

Intuition: The best way is to distribute an equivalent sample size of r evenly among all specified values for a node X_i .

BUT: In general the random variables do not have the same number of values.

Solution: Pick

$$r_{max} = \max_i r_i$$

as the equivalent sample size.

Justification: Suppose Y is a variable with r_{max} values. In order to ‘know’ Y has r_{max} values, it is arguable that minimally our prior experience must be equivalent to having seen each of them occur once. Therefore, our prior sample size must be at least equal to r_{max} . Since X is in that prior sample, there are also r_{max} observations of values of X .