Graphical Models for Inference and Decision Making

Instructor: Kathryn Blackmond Laskey

Spring 2019

Unit 2: Graphical Probability Models

Learning Objectives

1. Specify a joint distribution for a collection of uncertain hypotheses using
   - A graph to represent conditional dependence relationships
   - Local probability distributions to represent strength of relationships
2. Given a graphical model, identify
   - The probability of any configuration of hypotheses (exact if directed graphical model; up to normalization constant if undirected graphical model)
   - Whether a set of hypotheses is independent of another set given a third set
3. Use within-distribution structure to simplify specification of local probability distributions
4. Understand and use terms defined in this unit:
   - Sample space
   - Random variable
   - Conditional independence
   - d-separation
   - Directed graphical model (aka Bayesian network)
   - Markov network (undirected graphical model)
   - Explaining away and intercausal dependence
   - Local probability distribution
   - Context-specific independence
   - Independence of causal influence (ICI)

Unit 2 Outline

- ⇨ Graphical Probability Models: Overview
- Graph Theory Basics
- Graphical Probability Models: Formal Definitions
- Node-level Independence

Graphical Representation of Dependence

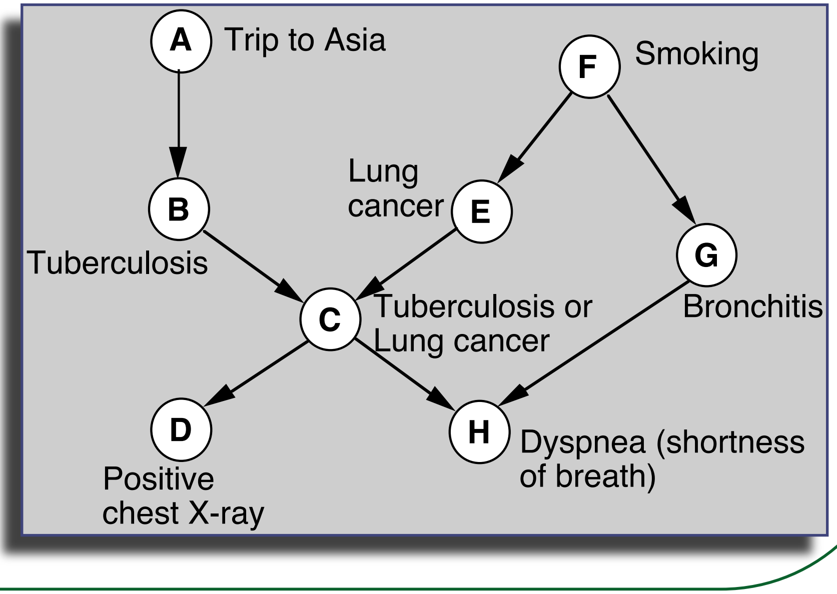
- Graphical models exploit conditional independence to construct tractable and parsimonious statistical models
- Graphs are a useful tool for specifying and visualizing dependencies between random variables
  - Random variables (RVs) are represented by nodes
  - Direct dependencies are represented by edges connecting nodes
  - Absence of an edge between 2 RVs means no direct dependence
- Joint distribution is obtained as product of local distributions involving small numbers of RVs

Example: Chest Clinic

- Direct dependence (e.g., trip to Asia and tuberculosis)
- Indirect dependence (e.g shortness of breath depends on smoking through bronchitis
- Probability distribution factorizes:

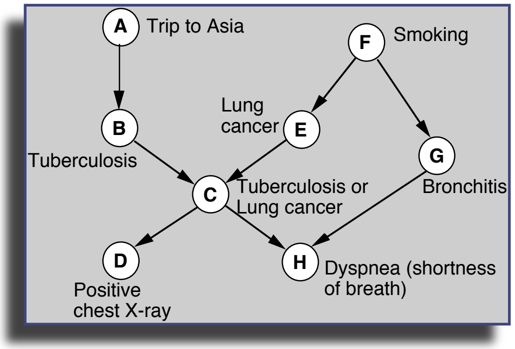
$$P(a,b,c,d,e,f,g,h) \\ = P(a)P(b|a)P(c|b,e)P(d|c)P(e|f)P(f)P(g|f)P(h|c,g)$$

Notation convention: Uppercase letters denote random variables; lowercase letters denote values



Independence Simplifies Specification and Inference

- General probability distribution for 8 random variables with 2 values each:
  - $2^8 = 256$ possible values for (A,B,C,D,E,F,G,H)
  - 255 probabilities need to be specified
  - Marginal probability example:
    - » $P(b_2) = \sum_{a,c,d,e,f,g,h} P(a,b_2,c,d,e,f,g,h)$
    - » 127 addition operations
  - Conditional probability example:
    - » $P(f_2|g_1) = \dfrac{P(f_2,g_1)}{P(g_1)} = \dfrac{\sum_{a,b,c,d,e,h} P(a,b,c,d,e,f_2,g_1,h)}{\sum_{a,b,c,d,e,h} P(a,b,c,d,e,f_1,g_1,h) + \sum_{a,b,c,d,e,h} P(a,b,c,d,e,f_2,g_1,h)}$
    - » 127 additions (63 + 63+1); 1 division



- *What if each node had 10 states?*
- *What if the network had 50 nodes, 5 states per node, 3 parents per node?*

- Network with structure of Chest Clinic example:
  - 256 possible values for (A,B,C,D,E,F,G,H)
  - P(a,b,c,d,e,f,g,h) = P(a)P(b|a)P(c|b,e)P(d|c)P(e|f)P(f)P(g|f)P(h|c,g)
    - » 18 = (1 + 2 + 4 + 2 + 2 + 1 + 2 + 4) probabilities to be specified [14 with deterministic influence at C]
  - Marginal probability example:
    - » $P(b_2) = P(b_2|a_1)P(a_1) + P(b_2|a_2)P(a_2) = (1-P(b_1|a_1))P(a_1) + (1-P(b_1|a_2))(1-P(a_1))$
    - » 1 addition, 3 subtractions, and 2 multiplications
  - Conditional probability example:
    - » $P(f_2|g_1) = \dfrac{P(g_1|f_2)P(f_2)}{P(g_1|f_1)P(f_1) + P(g_1|f_2)P(f_2)}$
    - » 1 addition, 3 subtractions, and 2 multiplications for $P(b_2)$, 3 multiplications, 1 addition, 1 division

Specification and inference scale exponentially in general probability models, linearly in Bayesian networks with simple structure, and in between for more complex Bayesian networks

Types of Graphical Probability Model (1 of 2)

- Bayesian Network
  - Directed acyclic graph represents direct dependence relationships
  - Each RV is independent of its nondescendents given its parents
  - Joint distribution over RVs factorizes as: $p(x) = \prod_{v} p(x_v | pa(x_v))$
    - » $pa(v)$ are parents of v

- Markov Network
  - Undirected graph represents dependencies among propositions
  - Each variable is independent of the rest of the graph given its neighbors
  - Joint distribution over RVs factorizes as: $p(x) = \frac{1}{Z}\prod_{C} \psi_C(x_C)$
    - » C runs over maximal complete subgraphs (called *cliques*)
    - » $\psi_C(x_C)$ is called a *clique potential*
    - » Z is a normalization constant known as the *partition function*
      - This term comes from statistical physics, where $p(x)$ represents a probability distribution over microstates of a physical system in thermal equilibrium at a given temperature, and Z is a function of temperature.

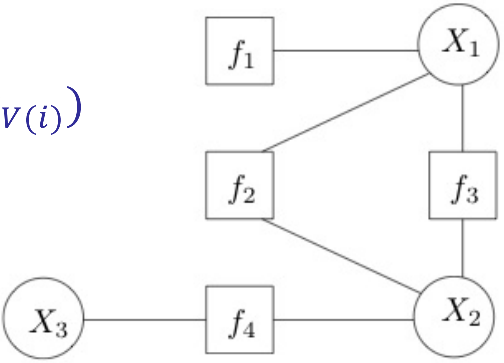
Reference: Cowell, et al, 1999

Types of Graphical Probability Model (2 of 2)

- Graphical Model on Chain Graph (generalizes BNs and MNs)
  - Graph with no directed edges represents dependence relationships
  - Each RV is independent of its nondescendents given its parents and neighbors
  - Joint distribution over RVs factorizes as: $p(x) = \prod_i p(x_{V(i)} | x_{pa(V(i))})$
    - $V(i)$ are disjoint sets of nodes forming a *dependence chain*. Nodes in $V(i)$ are connected by undirected arcs; all directed arcs go from lower to higher i
    - Parents and neighbors of a node x are called the *boundary* of x

Reference: Cowell, et al, 1999

- Factor graph
  - Bipartite graph: 2 kinds of nodes
    - Random variables (circles)
    - Factors (rectangles)
  - Joint distribution over RVs factorizes as: $p(x) = \frac{1}{Z}\prod_i f_i(x_{V(i)})$
    - $V(i)$ are random variable nodes connected to factor *i*
  - Bayesian networks, Markove fields and graphical models on chain graphs can all be represented as factor graphs



Reference: Kschischang, et al, 2003

Unit 2 Outline

- Graphical Probability Models: Overview
- Graph Theory Basics
- Graphical Probability Models: Formal Definitions
- Node-level Independence

Relations and Graphs

- **D2.1**: A *binary relation* R on a set X is a set of ordered pairs of elements (x, x'), where $x \in X$ and x'□X.
  - The *relatives* Rel(x) are those x' for which (x,x')∈R.
    - [Note: x' ∈ Rel(x) does not necessarily imply x∈ Rel(x')]
  - A binary relation is *reflexive* if x∈R ⇒ x∈Rel(x).
  - A binary relation is *irreflexive* if x∈R ⇒ x∉Rel(x).
  - A binary relation is *symmetric* if for all x, x'□X, x∈Rel(x') implies x'∈Rel(x).
- Relations are used to express knowledge about relationships
  - E.g., Children and their mothers:
    - R = {(Sarah Laskey, Kathryn Laskey), (Rob Laskey, Kathryn Laskey), (Kathryn Laskey, Frances Blackmond), (Frances Blackmond, Jane Newell…}
- **D2.2**: A *directed graph* (*digraph*) $G = (V, E)$ consists of a finite set V of vertices (nodes) and a binary relation E on V.
  - The relation E is called the adjacency relation
  - The elements (u, v) of E are called the *edges* of G.
- **D2.3**: An undirected graph $G = (V, E)$ is an irreflexive graph in which the adjacency relation is symmetric. That is, if $(u, v) \in E$ then $(v, u) \in E$.
- **D2.4**: A bipartite graph $G = (V_1, V_2, E)$ is a graph whose vertices consist of two disjoint sets V_1 and V_2, such that every edge connects a vertex in V_1 to a vertex in V_2.

In Bayesian networks and Markov networks:

- **A *node* or *vertex* (plural *vertices*) represents a random variable**
- **An *arc* or *edge* represents a dependence relation**

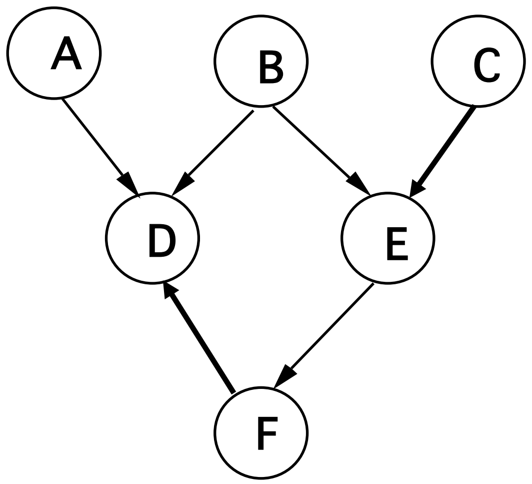
In factor graphs:

- **A *variable node* represents a random variable and a *factor node* represents a factor of the conditional distribution**
- **An *arc* represents a dependence relation**

Examples

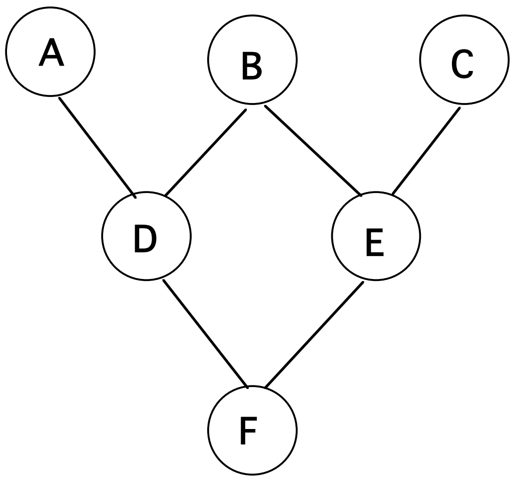
Directed graph

- V = {A, B, C, D, E, F}
- E = {(A,D), (B,D), (B,E), (C,E), (E,F), (F,D)}



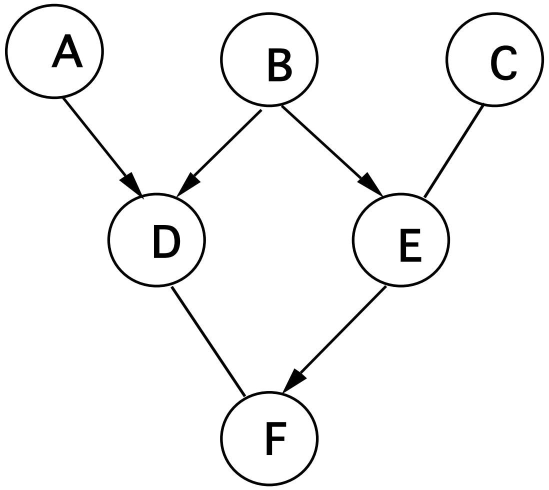
Undirected graph

- V={A, B, C, D, E, F}
- E = {(A,D), (D A), (B,D), (D,B), (B,E), (E,B), (C,E), (E,C), (D,F), (F,D), (E,F), (F,E)}



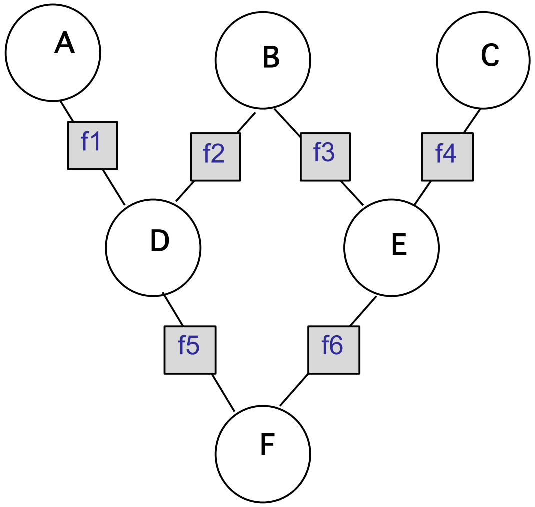
General graph

- V={A, B, C, D, E, F}
- ,E = {(A,D), (B,D), (B,E), (C,E), (E,C), (D,F), (F,D), (E,F)}



Bipartite graph

- V1={A, B, C, D, E, F}
- V2={f1, f2, f3, f4, f5, f6}
- E = {(A,f1), (B,f2), (B,f3), (C,f4), (D,f1), (D,f2), (D,f5), (E,f3), (E,f4), (E,f6), (F,f5), (F,f6)}



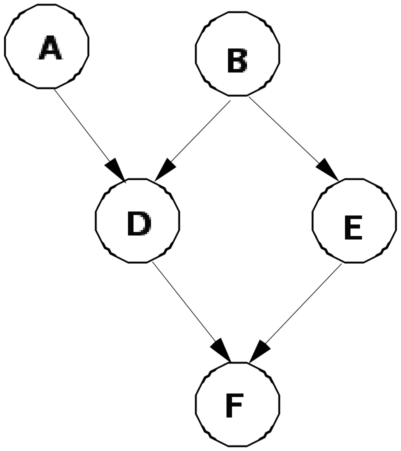
Chains, Paths and Cycles

- **D2.5**: Let G = (V,E) be a graph.
  - A *chain* between vertices v_0 and v_m is a sequence of vertices $[v_1, \ldots, v_m]$ such that either $(v_{i-1}, v_i) \in E$ or $(v_i, v_{i-1}) \in E$ for each i. *Chains do not take arc direction into account.*
  - A *path* between vertices v_0 and v_m is a sequence of vertices $[v_1, \ldots, v_m]$ such that $(v_{i-1}, v_i) \in E$ for each i. *Paths take arc direction into account.*
  - A (directed) *cycle* is a path from v_0 to v_0.
- **D2.6**: A *chain graph* is a graph with no cycles (no directed paths from a node to itself).
- **D2.7**: A *directed acyclic graph* (DAG) is a chain graph with no undirected edges.
  - *Technically we should say acyclic directed graph but DAG is the common term*
- **D2.8**: Let G=(V,E) be a directed graph.
  a. The *parents* (predecessors) pa(v) of v are the vertices u such that $(u,v) \in E$.
  b. The *children* (successors) ch(v) of v are the vertices u such that $(v,u) \in E$.
  c. The *ancestors* of v are the vertices u such that there is a path from u to v.
  d. The *descendents* of v are the vertices u such that there is a path from v to u.
  e. An *ancestral ordering* of the vertices in the graph is an ordering in which all the ancestors of a vertex v are ordered before v.
- **Theorem 2.1**: An ancestral ordering of the vertices in a digraph exists only if the digraph is a DAG.
- **Theorem 2.2**: If v is a vertex in the DAG G=(V,E), it is always possible to obtain an ancestral ordering of the vertices in G so that only the descendents of the vertices v are labeled after v.
  - This theorem will be important in junction tree construction

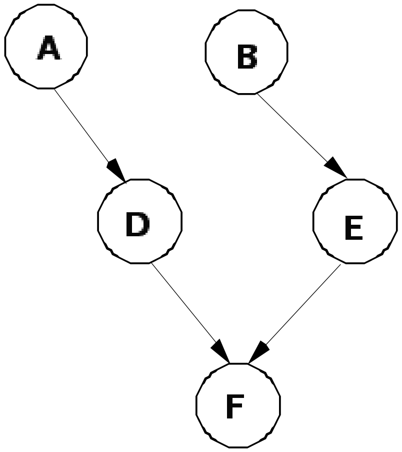
The theorems are from Neapolitan, 1991

Singly Connected Graphs and Trees

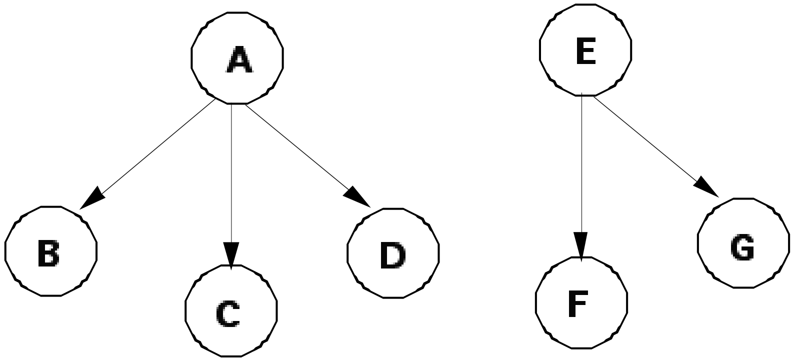
- **D2.9**: Let G=(V,E) be a DAG.
  - A directed graph G is *singly connected* if there is at most one chain between any two vertices.
  - G is a *forest* if every vertex has at most one parent.
  - G is a *tree* if it is a forest and there is only one vertex with no parent.



Multiply Connected DAG



Singly Connected DAG

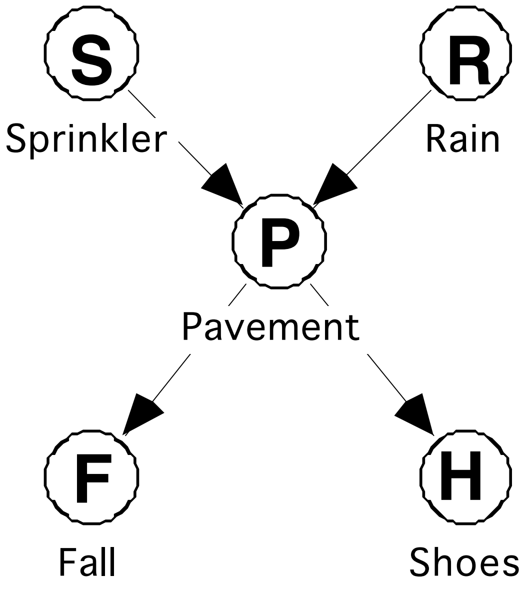


Forest Containing Two Trees

Inference is most efficient in singly connected networks

Terminology: A *node* is the same as a *vertex*

Active and Inactive Chains

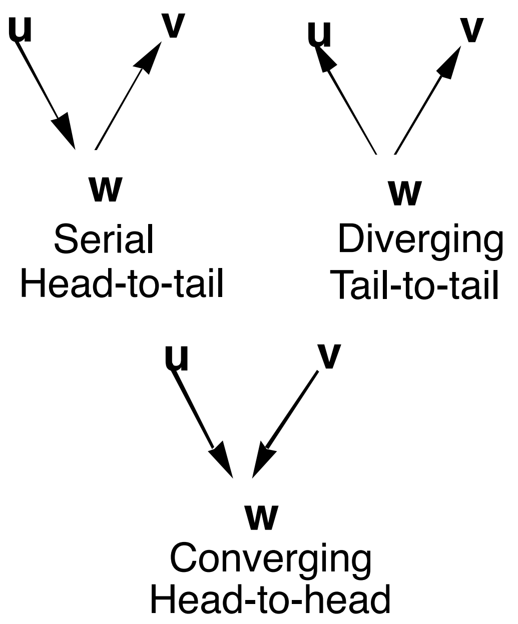


- R and S are independent, but are dependent conditional on P
- F and H are dependent (if fall is more likely, then so are wet shoes) but are independent given P
- S and F are dependent, but are independent given P

- *Active* chains carry information; *inactive* chains do not carry information
- Observations at tail-to-tail or head-to-tail links break a chain
  - The following chains are *active* when there is no evidence at P:
    - » S — P — H
    - » S — P — F
    - » R — P — H
    - » R — P — F
    - » F — P — H
  - Adding evidence at P *inactivates* these chains
- Observations at or below head-to-head links activate a chain
  - The chain S — P — R is *inactive* when there is no evidence at or below P
  - Adding evidence at P, F, and/or H *activates* the chain

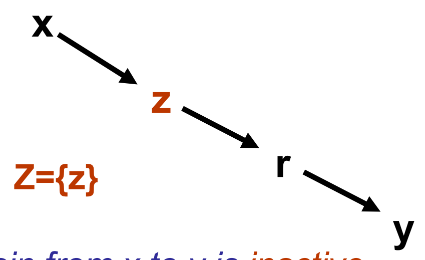
Active and Inactive Chains: Formal Definitions

- **D2.10**: Let G=(V,E) be a DAG with u,v,w∈V.
  - If (w,u) ∈E and (w,v)∈E, then the arcs (w,u) and (w,v) are *diverging* or meet tail-to-tail at w.
  - If (u,w)∈E and (w,v)∈E, then the arcs (u,w) and (w,v) are *serial* or meet head-to-tail at w.
  - If (u,w)∈E and (v,w)∈E, then the arcs (u,w) and (v,w) are *converging* or meet head-to-head at w.

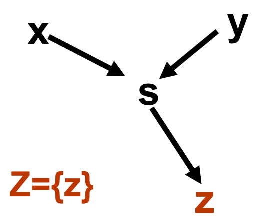


- **D2.11**: Let G=(V,E) be a DAG, and let Z⊂V, and let x and y be vertices not in Z. Then a chain between x∈V and y∈V is *not active* given Z if one of the following is true:
  - There is a vertex z∈Z on the chain such that the connection at z is diverging or serial;
  - There is an intermediate vertex x∈V such that the connection at s is converging and neither s nor any of its descendents is in Z.

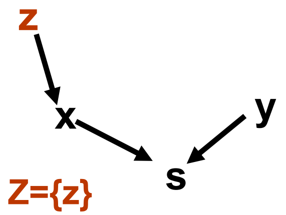
All other chains are *active*.



Chain from x to y is inactive due to serial link at z



Chain from x to y is active due to converging link at s with descendent in Z



Chain from x to y is inactive because Z does not contain s or any of its descendents

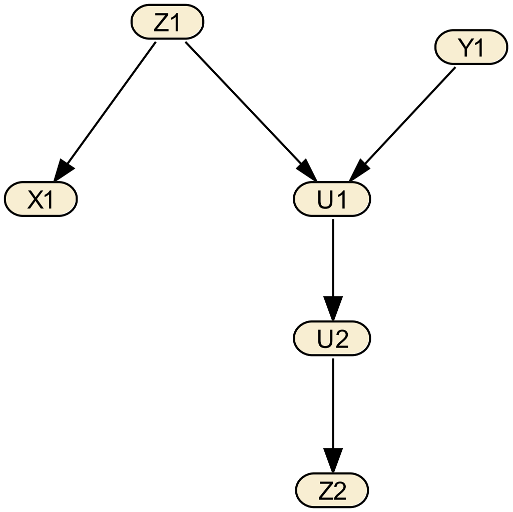
d-Separation

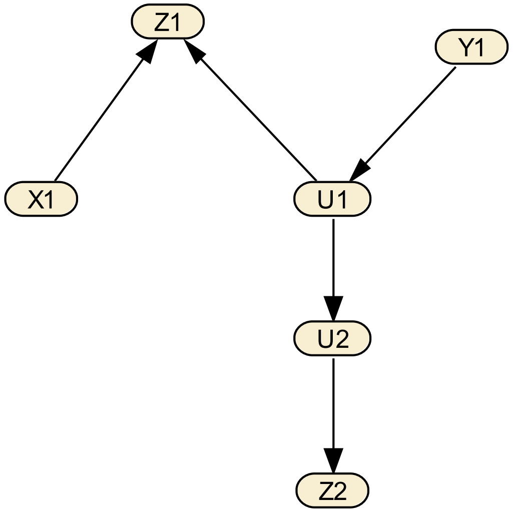
- **D2.12**: Let G=(V,E) be a DAG, and let X, Y, and Z be disjoint subsets of vertices in V. Then X and Y are *d-separated* by Z if every chain between a node in X and a node in Y is not active given Z.
- In words: Two sets of vertices are d-separated by a third set of vertices if there are no active chains between a node in the first set and a node in the second set given the third set

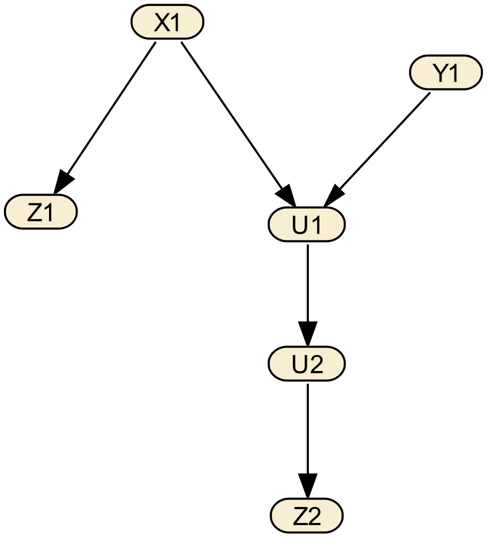
d-separation is a powerful concept

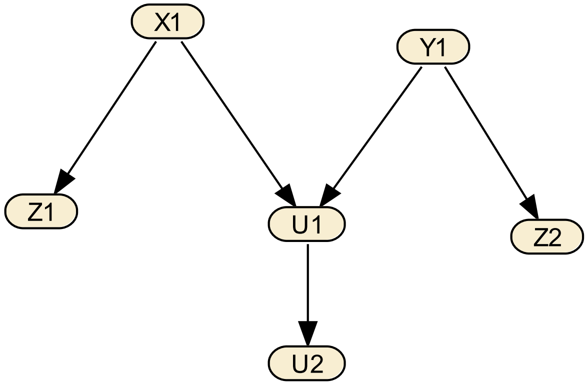
- ***d-separation of vertices in a graph corresponds to conditional independence of the associated random variables***
- ***If X and Y are d-separated by Z in G then X and Y are conditionally independent given Z in any directed graphical probability model associated with G***
- ***d-separation is the mathematical basis for efficient inference algorithms for Bayesian networks***

d-Separation Examples









Does Z d-separate X from Y?

How to Tell Whether Sets of Nodes are d-Separated

- To find out whether Z d-separates X from Y, follow these steps.
  1. Find all chains connecting random variables in X to variables in Y.
  2. Do for all chains until an active chain is found:
     a) Is there a node in Z on the chain at which connection is diverging or serial? If yes, chain is not active.
     b) Is there a node on the chain at which connection is converging? If yes, for each such node:
        - Check whether the node or any of its descendents is in Z. If no, chain is not active.

     c) If you have followed steps a and b and have not declared the chain inactive, then the chain is active.
  3. If an active chain was found in step 2, then X and Y are not d-separated by Z. If no active chain was found, X and Y are d-separated by Z.
- IMPORTANT:
  - *To declare sets not d-separated you only have to find one active chain*
  - *To declare sets d-separated you have to show that all chains are inactive*
- The easiest way to tell whether nodes in a Bayesian network are conditionally independent is to check d-separation:
  - *If Z d-separates X and Y in the graph of the Bayesian network, then X and Y are conditionally independent given Z in the joint distribution represented by the Bayesian network*

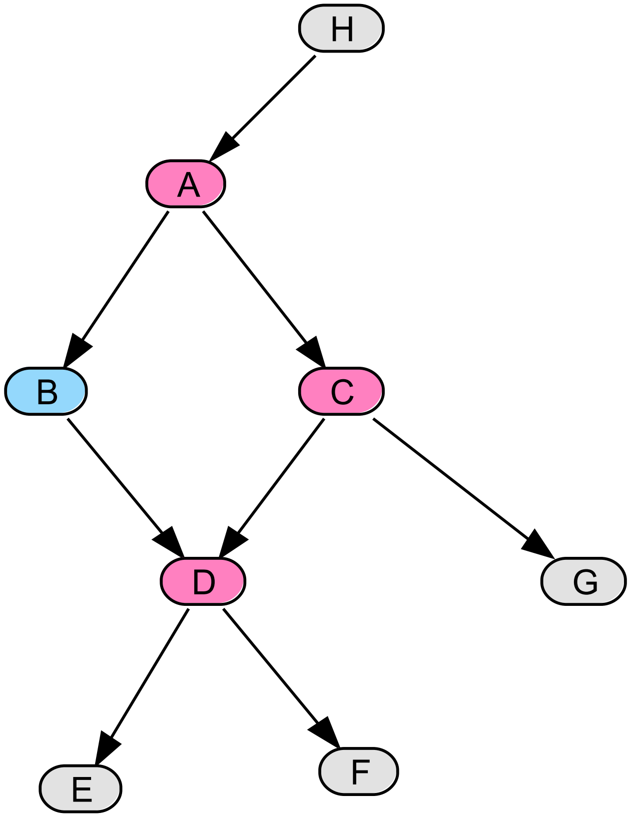
Markov Blanket

- **D2.13**: The *Markov blanket* of a node consists of:
  - its parents, children, and other parents of its children (co-parents) if graph is a DAG;
  - its neighbors if graph is undirected.

  The Markov blanket of a node consists of all nodes whose local distributions mention it, along with all nodes their local distributions mention

- **Theorem 2.3**: A node's Markov blanket d-separates it from all other nodes
  - A node is conditionally independent of all other nodes given its Markov blanket

- Example: Markov blanket of B is A,C,D



$$\frac{P(b_1 \mid a,c,d,e,f,g,h)}{P(b_2 \mid a,c,d,e,f,g,h)} = \frac{P(b_1,a,c,d,e,f,g,h)}{P(b_2,a,c,d,e,f,g,h)}$$

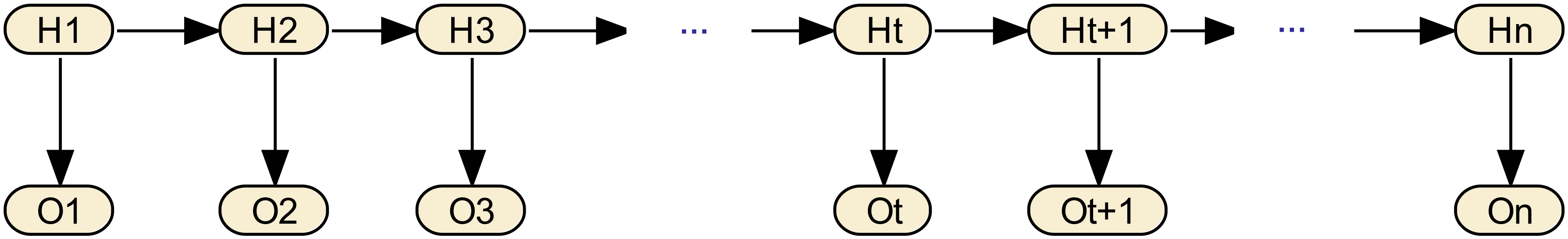
$$= \frac{\cancel{P(a|h)}P(b_1|a)\cancel{P(c|a)}P(d|b_1,c)\cancel{P(e|d)}\cancel{P(f|d)}\cancel{P(g|c)}\cancel{P(h)}}{\cancel{P(a|h)}P(b_2|a)\cancel{P(c|a)}P(d|b_2,c)\cancel{P(e|d)}\cancel{P(f|d)}\cancel{P(g|c)}\cancel{P(h)}}$$

$$= \frac{P(b_1 \mid a)P(d \mid b_1,c)}{P(b_2 \mid a)P(d \mid b_2,c)}$$

The ratio of probabilities of two possible values of B depends only on the values of B's Markov blanket, and does not change if we change nodes outside the Markov blanket of B

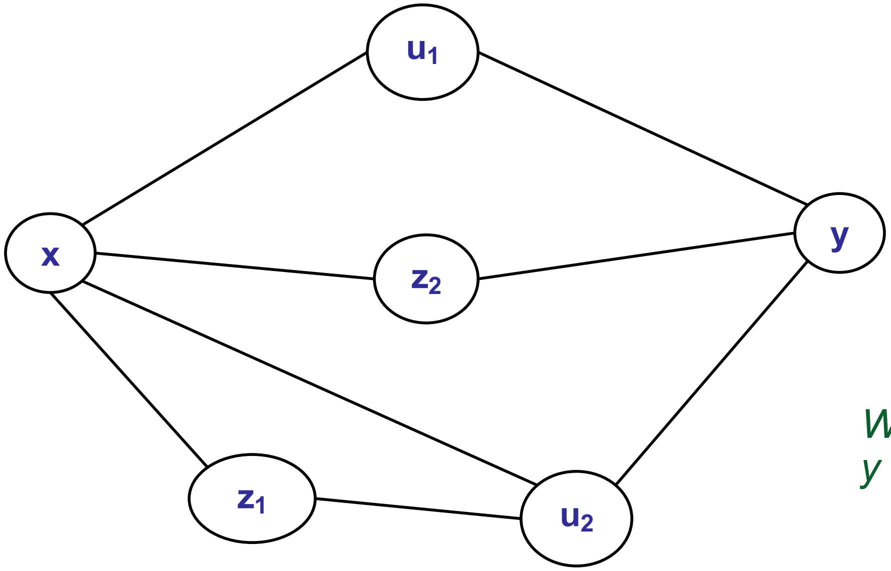
Hidden Markov Model

- A *hidden Markov model* is a kind of directed graphical model used often in applications such as speech recognition and protein sequencing
- The hidden states H_t represent unobservable system states
- The observations O_t represent observations (also called emissions) that depend on the hidden system state
- The Markov blanket of H_t is $\{H_{t-1}, H_{t+1}, O_t\}$
- HMMs have the *memoryless property* – given the immediately preceding state $H_{t-1,}$ the present and future (H_t, O_t, H_{t+1}, O_{t+1}, …) is independent of all other information about the past



Graph Separation in Undirected Graphs

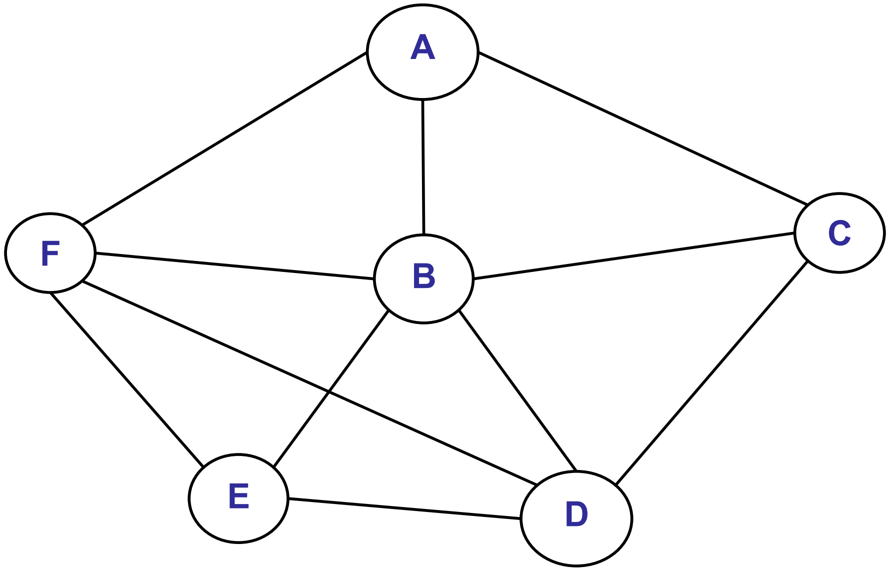
- **D2.14**: Let G=(V,E) be an undirected graph, let Z⊂V, and let x and y be vertices not in Z. Then a chain (path) between x∈V and y∈V is *not active* given Z if there is a vertex z∈Z on the chain.



Which chains between x and y are active given Z={z_1, z_2}?

Undirected Graph: Cliques

- **D2.15**: In an undirected graph G, a *clique C* is a maximally connected subgraph. That is, C ⊂ V is a clique if every pair of nodes in C is connected by an arc and C is the largest such subset of V. That is, if C ⊂ W ⊂ V and every pair of nodes in W is connected by an arc, then C=W.



What are the cliques in this graph?

Unit 2 Outline

- Graphical Probability Models: Overview
- Graph Theory Basics
- ➡ Graphical Probability Models: Formal Definitions
- Node-level Independence

Probability Spaces

- **D2.16**: A *sample space* Ω is a set containing mutually exclusive and exhaustive outcomes for some trial or experiment.
  - Exclusive: only one element of Ω will occur
  - Exhaustive: at least one element of Ω will occur
    - » Example: Sample space for coin flip is {H,T}; for 2 tosses it is {HH,HT,TH,TT}
- **D2.17**: A *field* (also called an *algebra*) $\mathcal{F}$ is a set of subsets of Ω such that:

  a. $\Omega \in \mathcal{F}$

  b. If E_1 and $E_2 \in$ F then $E_1 \cup E_2 \in \mathcal{F}$.

  c. E$\in \mathcal{F}$ implies $E^C \in \mathcal{F}$. (E^C is the complement of E)

    - » a. and c. imply that $\varnothing \in \mathcal{F}$
    - » A field is a σ-field if $\bigcup_{i=1}^{\infty} E_i \in \mathcal{F}$ whenever $E_1, E_2, \ldots \in \mathcal{F}$
- **D2.18**: A *probability space* is a triple (Ω, $\mathcal{F}$,P), where Ω is a sample space, $\mathcal{F}$ is a field over Ω, and P is a probability measure on $\mathcal{F}$.

  a. P(E) ≥ 0

  b. P(Ω) = 1

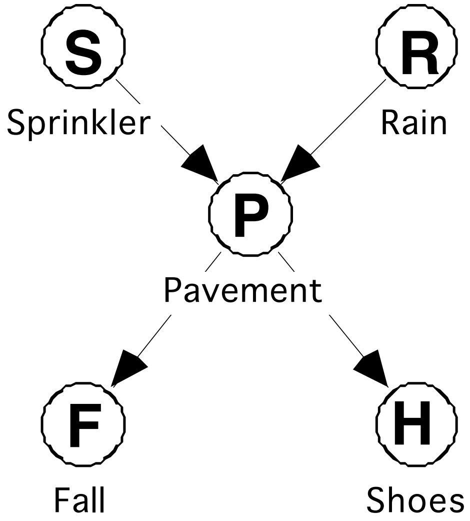
  c. If $E_1 \cap E_2 = \varnothing$ then $P(E_1 \cup E_2) = P(E_1) + P(E_2)$

  Note: If $\mathcal{F}$ is a σ-field we say P is *countably additive* if $P\left(\bigcup_{i=1}^{\infty} E_i\right) = \sum_{i=1}^{\infty} P(E_i)$ when $E_i \cap E_k = \varnothing$ for all i≠k

Random Variable: Formal Definition

- **D2.19**: Let (Ω, $\mathcal{F}$,P) be a probability space. A *random variable* (RV) is a function mapping Ω to a set called the *possible values*
- Sprinkler example (sprinkler or rain could cause wet pavement; wet pavement could cause fall and wet shoes)
  - The sample space consists of *situations* in which it is or is not raining, the sprinkler is or is not on, etc.
    - » Ω = {(s,r,p,f,h,…)}
  - Random variables map situations to features:

» Sprinkler:	S maps (s,r,p,f,h,…) to s.
» Rain:	R maps (s,r,p,f,h,…) to r.
» Pavement:	P maps (s,r,p,f,h,…) to p.
» Fall:	F maps (s,r,p,f,h,…) to f.
» Shoes:	H maps (s,r,p,f,h,…) to h.



- We will often use uppercase letters (e.g., S, R) to refer to a random variable and lowercase letters (e.g., s, r) to refer to possible values
  - For example, we say S=s to mean the random variable S takes on the value s.

Graphical Models and Local Specification

- Traditionally probability distributions are defined "top down"
  - We assume a sample space, a probability measure on the sample space, and a set of random variables mapping elements of the sample space to possible outcomes
  - Joint and marginal probability distributions for the random variables are derived from the global joint distribution
- *In practice, joint distributions of any complexity are almost always built up from elements that interact directly with only a few other elements*
- Powerful innovation: graphical models
  - Define random variables in terms of local distributions involving only a few other random variables
  - Provide conditions under which this process uniquely specifies a global joint distribution
  - Graph theory provides a powerful language for specifying dependence relationships and subsets of RVs on which local distributions are defined

Conditional, Marginal and Joint Distributions

- **D2.20**: Let $(\Omega, \mathcal{F}, P)$ be a probability space, where $\Omega = \Omega_1 \times \Omega_2 \times \ldots \times \Omega_n$. We call Ω the *joint sample space* and the elements $(x_1, \ldots, x_n)$ where $x_i \in \Omega_i$ the *configurations* on the joint sample space. Let X_i be the random variable mapping a configuration to its *i*th coordinate. We call the probability distribution P the *joint distribution* for $(X_1, \ldots, X_n)$.
  - The *marginal distribution* of a subset of random variables $(X_{i_1}, \ldots, X_{i_r})$ is:

$$P(X_{i_1}, \ldots, X_{i_r}) = \sum_{X_j, j \notin \{i_1, \ldots, i_r\}} P(X_1, \ldots, X_n)$$

Note: for continuous variables this sum becomes an integral.

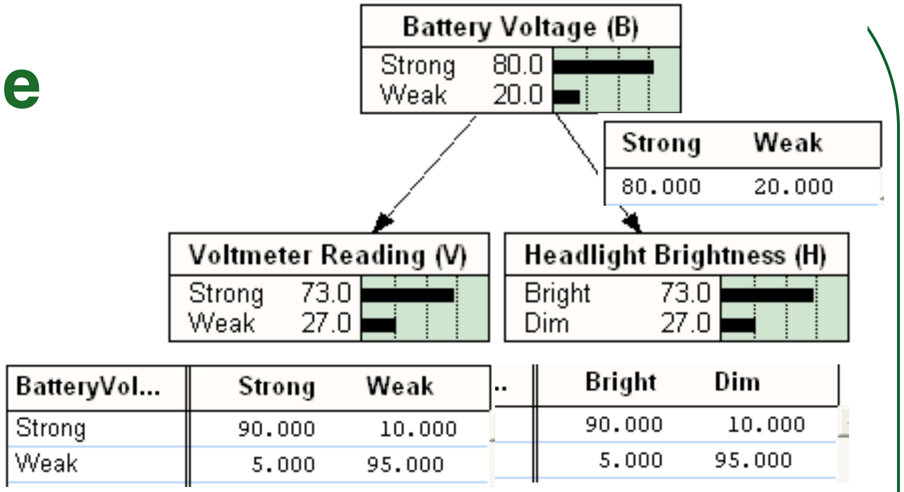
  - (Note that $P(X_i)$ represents a set of numbers $P(x_i)$, one for each possible value of the random variable X_i)
  - The *conditional distribution* of$(X_{i_1}, \ldots, X_{i_r})$ given$(X_{j_1}, \ldots, X_{j_s})$ is given by:

$$P(X_{i1}, \ldots, X_{ir} \mid X_{j1}, \ldots X_{js}) = \frac{P(X_{i1}, \ldots, X_{ir}, X_{j1}, \ldots X_{js})}{P(X_{j1}, \ldots X_{js})}$$

- **D2.21**: We say that a random variable X is *conditionally independent* of Y given Z if $P(X|Y,Z) \;=\; P(X|Z)$. If this is the case, we write $\mathcal{I}_P(X,Z,Y)$.

Example

- Joint distribution on B,V,H:
  - Pr(B,V,H) = Pr(B)Pr(V|B)Pr(H|B)
  - Pr(B=s,V=s,H=d) = 0.80x0.90x0.10
  - Pr(B=w,V=s,H=b) = 0.20x0.05x0.05
- Marginal distribution for V
  - Pr(V=s) = 0.73; Pr(V=w) = 0.27
- Conditional distributions
  - Pr(V=s | B=w) = 0.05
  - Pr(B=w | V=s) = 0.014
- V is conditionally independent of H given B
  - This can be read directly from the graph



Battery Voltage (B)	
Strong	80.0
Weak	20.0

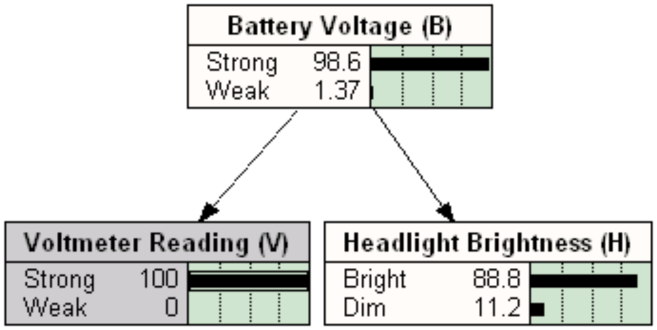
Strong	Weak
80.000	20.000

Voltmeter Reading (V)	
Strong	73.0
Weak	27.0

Headlight Brightness (H)	
Bright	73.0
Dim	27.0

BatteryVol...	Strong	Weak
Strong	90.000	10.000
Weak	5.000	95.000

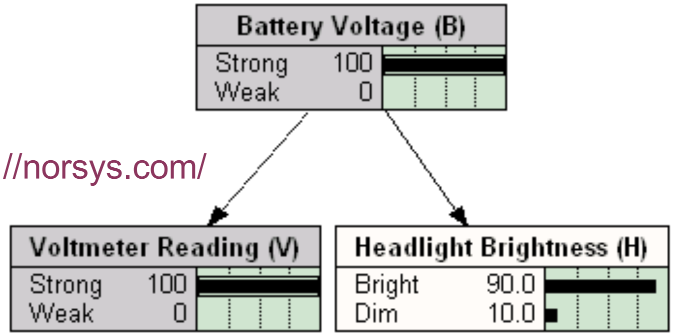
..	Bright	Dim
	90.000	10.000
	5.000	95.000



Battery Voltage (B)	
Strong	98.6
Weak	1.37

Voltmeter Reading (V)	
Strong	100
Weak	0

Headlight Brightness (H)	
Bright	88.8
Dim	11.2

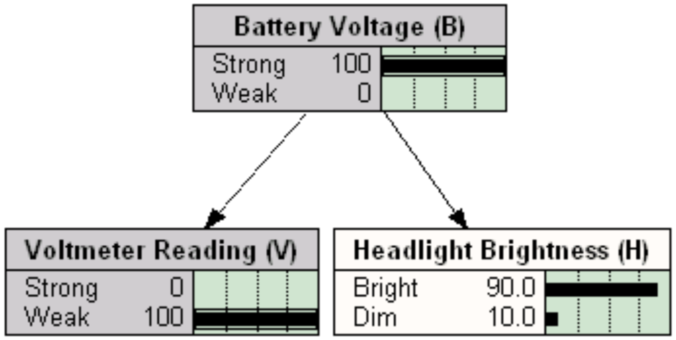


Battery Voltage (B)	
Strong	100
Weak	0

Voltmeter Reading (V)	
Strong	100
Weak	0

Headlight Brightness (H)	
Bright	90.0
Dim	10.0

Netica™ http://norsys.com/

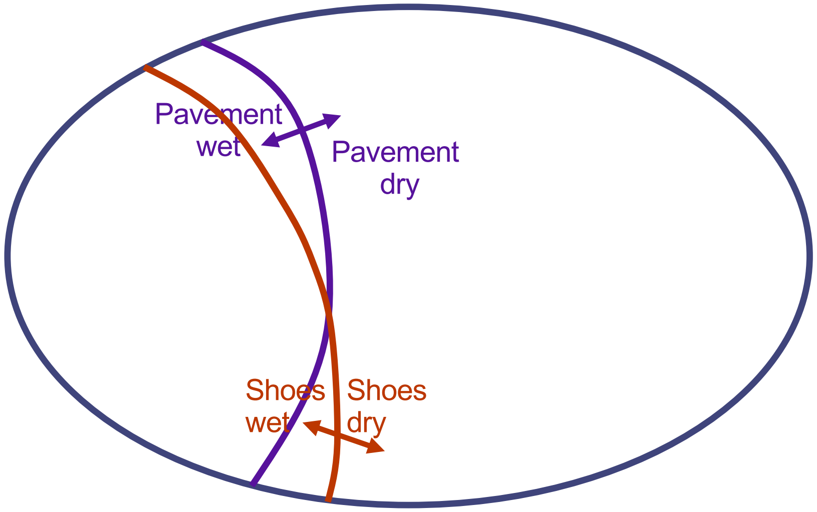


Battery Voltage (B)	
Strong	100
Weak	0

Voltmeter Reading (V)	
Strong	0
Weak	100

Headlight Brightness (H)	
Bright	90.0
Dim	10.0

A Venn Diagram to Illustrate Dependence



Which area represents the range of the random variable

- P(Pavement | Shoes=Wet)?

Which ratio of areas gives:

- P(Pavement=Wet| Shoes=Wet)?
- P(Pavement=Dry| Shoes=Dry)?

What would this picture look like if Pavement and Shoes were independent?

Bayesian Network: Formal Definition

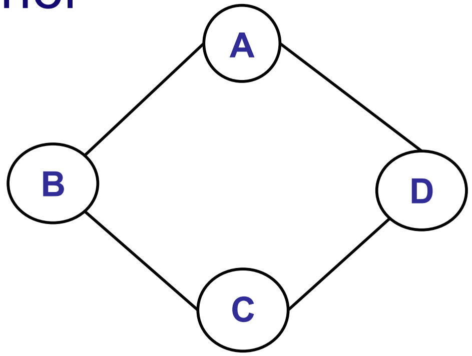
- **D2.22**: Let $G = (V,E)$ be a a directed graph. Let $\{X_v: v\in V\}$ be a set of RVs, one for each $v\in V$. Let $(\Omega,\mathcal{F},P)$ be a joint probability distribution on the X_v. Let $pa(v)$ and $nd(v)$ be the RVs associated with the parents and non-descendants of v, respectively. Then $B = (V,E,P)$ is a *Bayesian network* (BN) for the probability model $(\Omega,\mathcal{F},P)$ if for each RVX_v, $\mathcal{I}_P(X_v, pa(v), nd(v))$
  - Note: this implies $\mathcal{I}_P(X, pa(v), W)$ for each subset W of the random variables in $nd(v)$. (from Graphoid Axioms presented later)
- ***Theorem 2.4***: If $B = (V,E,P)$ is a Bayesian network, then $P(X_{\{v\in V\}})$ is:

$$P(X_{\{v\in V\}}) = \prod_{v\in V} P(X_v|pa(X_v))$$

  - If a DAG and a joint probability distribution form a BN, then we can compute the joint probabilities from the conditional distributions of RVs given their parents.
- ***Theorem 2.5***: Given an arbitrary DAG and a set of conditional distributions of nodes given their parents, they together determine a Bayesian network.
    - » There exists a joint probability distribution consistent with the probability assignments
    - » The distribution is unique
    - » The joint probability distribution yields the specified conditional distributions
    - » The independence assumptions encoded by the DAG are satisfied by the joint probability distribution

Undirected Graphical Models: The Misconception Example

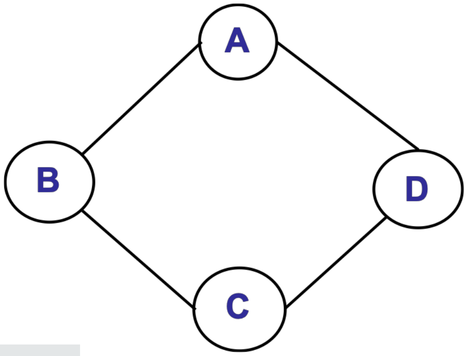
- Professor L had a typographical error in her vugraphs which could give rise to a misconception.
- Each student may resolve the misconception on his or her own
- The students in Professor L's class work in pairs
  - Alice works with Bob
  - Bob works with Charles
  - Charles works with Debbie
  - Debbie works with Alice
  - Alice and Charles do not work together (dislike each other)
  - Bob and Debbie do not work together (had a messy breakup)
- A student who resolves the misconception may enlighten his or her study partner



- *Each node in this Markov network represents whether the student (Alice, Bob, Charles, Debbie) has the misconception.*
- *The graph represents the dependence structure for this problem.*
- *No Bayesian network can represent this dependence structure.*

Misconception Example: Joint Distribution

- In general, exact joint probability distribution for undirected graphical model is intractable
- Misconception example is small enough to calculate joint probabilities of all possible states
  - 0 means has misconception; 1 means does not



A	B	C	D	ψ(A,B)	ψ(A,D)	ψ(B,C)	ψ(C,D)	Product	Joint Prob
0	0	0	0	0.7	0.7	0.7	0.7	0.2401	0.2466
1	0	0	0	0.1	0.1	0.7	0.7	0.0049	0.0050
0	1	0	0	0.1	0.7	0.1	0.7	0.0049	0.0050
1	1	0	0	0.9	0.1	0.1	0.7	0.0063	0.0065
0	0	1	0	0.7	0.7	0.1	0.1	0.0049	0.0050
1	0	1	0	0.1	0.1	0.1	0.1	0.0001	0.0001
0	1	1	0	0.1	0.7	0.9	0.1	0.0063	0.0065
1	1	1	0	0.9	0.1	0.9	0.1	0.0081	0.0083
0	0	0	1	0.7	0.1	0.7	0.1	0.0049	0.0050
1	0	0	1	0.1	0.9	0.7	0.1	0.0063	0.0065
0	1	0	1	0.1	0.1	0.1	0.1	0.0001	0.0001
1	1	0	1	0.9	0.9	0.1	0.1	0.0081	0.0083
0	0	1	1	0.7	0.1	0.1	0.9	0.0063	0.0065
1	0	1	1	0.1	0.9	0.1	0.9	0.0081	0.0083
0	1	1	1	0.1	0.1	0.9	0.9	0.0081	0.0083
1	1	1	1	0.9	0.9	0.9	0.9	0.6561	0.6739
							Z =	0.9736	

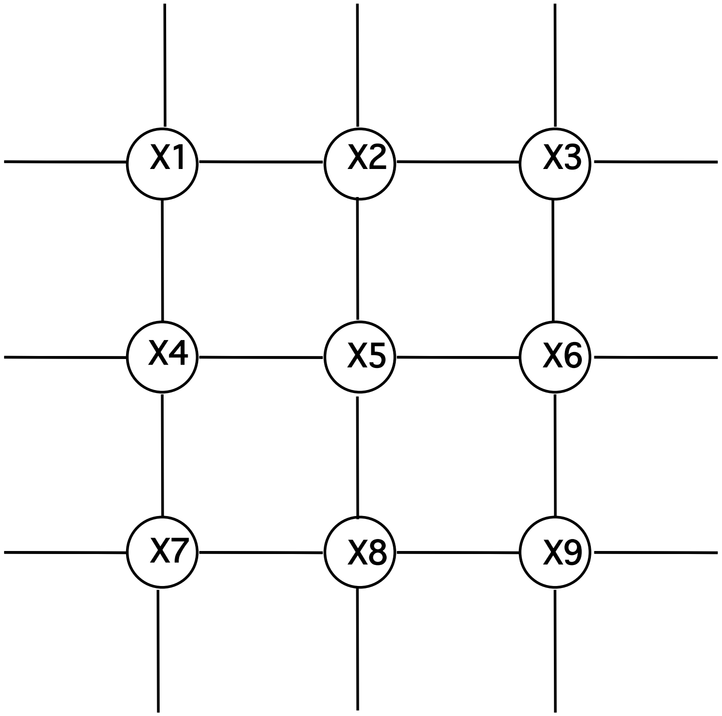
Other Graphical Models: Formal Definitions

- **D2.23**: Let $G = (V, E)$ be a an undirected graph. Let $\{X_v : v \in V\}$ be a set of RVs, one for each $v \in V$. Let $(\Omega, \mathcal{F}, P)$ be a joint probability distribution on the X_v. Let $ne(v)$ be the RVs for *neighbors* of v (nodes connected to X by an arc), and let $no(v)$ be the RVs for all nodes other than v and $ne(v)$. Then $M = (V, E, P)$ is a *Markov network* for the probability model $(\Omega$,F,$P)$ if: $\mathcal{I}_P(X, ne(X), no(X))$.
- **D2.24**: Let $G = (V, E)$ be a an undirected graph. Let $\{X_v : v \in V\}$ be a set of RVs, one for each $v \in V$. Let $(\Omega, \mathcal{F}, P)$ be a joint probability distribution on the X_v. Let $bd(v)$ be RVs for the *boundary* (parents and neighbors) of v and let $nd(v)$ be the RVs for non-descendents of v. Then $C = (V, E, P)$ is a *chain network* for the probability model $(\Omega, \mathcal{F}, P)$ if: $\mathcal{I}_P(X, bd(X), nd(X))$.
- **D2.25**: Let $G = (V, W, E)$ be a bipartite graph. Let $\{X_v : v \in V\}$ be a set of RVs, one for each $v \in V$. For each $w \in W$, let $X_w = \{X_v : (v, w) \in E\}$ be the subset of RVs with an edge connecting them to w. Let $(\Omega, \mathcal{F}, P)$ be a joint probability distribution on the X_v. Then $F = (V, E, P)$ is a *factor graph* for the probability model $(\Omega, \mathcal{F}, P)$ if there are non-negative real-valued functions $f_w(x_w)$ and a real number Z such that for every x:

$$p(x) = \frac{1}{Z} \prod_i f_w(x_w)$$

Example: Pairwise Markov Network for Image Analysis

- In this simple model, the intensity of each pixel depends on the intensities of the neighboring pixels



X5 is independent of the rest of the graph given X2, X4, X6 and X8

See work of S. and D. Geman at Brown University

- A more complex model might have 3 kinds of nodes: pixels, lines, and observations
  - Observation nodes depend on node being observed (in "blur" models observations also depend on neighboring observations)
  - A line between two pixels breaks the dependence between them
  - Lines tend to be continuous and not to bend

Notes on Directed and Undirected Graphical Models

1. In a Bayesian network, a node's *parents* shield it from the influence of its *non-descendents*. (If you know the values of the node's parents, the node's distribution remains unchanged no matter what the value of other non-descendents.)

2. In a Markov network, a node's *neighbors* shield it from the influence of *all other nodes*. (If you know the values of the node's neighbors, the node's distribution remains unchanged no matter what the value of any other nodes in the network.)

3. We will learn how to transform a Bayesian network into a Markov network that represents the same probability distribution.

4. Graphical models on chain graphs generalize both Bayesian networks and Markov networks.

5. Markov networks can represent problems (e.g., image analysis) in which there is no natural direction of influence. Chain networks can be used when there is a natural direction of influence for some RVs but not others.

6. Can read conditional independence from the graph structure.
   a. In an undirected graphical model, if no path from A to B exists except through Z, then $\mathcal{I}$(A,Z,B).
   b. Reading independence from directed graphs is more complex: if Z d-separates A from B then $\mathcal{I}$(A,Z,B)

d-Separation and Conditional Independence

- **Theorem 2.6**: Let $C = (V, E, P)$ be a Bayesian network with $G = (V, E)$ the DAG. Suppose that X and Y are d-separated by Z. Then $\mathcal{I}_P(X, Z, Y)$.
  - Recall: $\mathcal{I}_P(X, Z, Y)$ means X is independent of Y given Z, i.e., $P(X, Y \mid Z) = P(X \mid Z)\, P(Y \mid Z)$

If two sets of nodes are d-separated by the evidence then:

- ***They are conditionally independent of each other given the evidence***
- ***Given the evidence, learning the value of any nodes in the first set will not change the probability distribution of the nodes in the second set***

This mathematical property is the basis of:

- **Computationally efficient inference in Bayesian networks**
- **Parsimonious knowledge representation**
- **Statistically efficient parameter and structure learning methods for Bayesian networks**
- **Feasible knowledge elicitation in Bayesian networks**

Relevance and Dependence

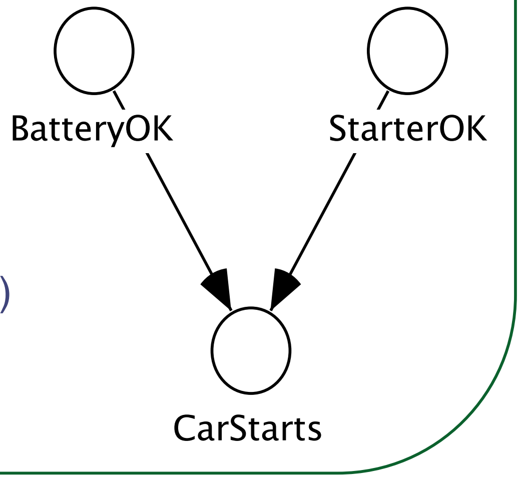
- Conditional independence is a basic concept in building graphical models. It is important to develop your intuitions about conditional independence through practice with building models and exploring their behavior.
- We say one random variable is relevant to another in a given context if learning the value of the first influences beliefs about the other
  - Conditional independence models irrelevance given the conditioning variables
- A hierarchy of difficulty of judgment:
  - Is the truth of A *relevant* to the truth of B in context K?
  - What is the *direction* of the influence (does A make B more or less likely?)
  - By *how much* does a change in the certainty of A affect the certainty of B?
- We can specify a Bayesian network by:
  - DAG to encode relevance (conditional dependence) relationships
  - Local belief tables to express direction and strength of relationships
- Pearl sees DAGs as a model for human knowledge representation:
  - "...the notions of dependence and conditional independence are more basic to human reasoning than are the numerical values attached to probability judgments"
  - "... these graphical metaphors suggest that the fundamental structure of human knowledge can be represented by dependency graphs and that mental tracing of links in these graphs are the basic steps in querying and updating that knowledge."
- Bayesian models of cognition are becoming increasingly popular

Relevance and Causality

- Pearl sees causation as fundamental to our ability to process information
  - "Causal claims are much bolder than those made by probability statements; not only do they summarize relationships underlying the data, but they also predict relationships that should hold when the distribution changes…a *stable* dependence between X and Y…that cannot be attributed to some prior cause common to both…[and is] preserved when an exogenous control is applied to X."
  - "The asymmetry conveyed by causal directionality is viewed as a notational device for encoding still more intricate patterns of relevance relationships… Two events do not become relevant to each other merely by virtue of predicting a common consequence, but they do become relevant when the consequence is actually observed. The opposite is true for two consequences of a common cause."
    - » Rain and Sprinkler are *independent* until wet pavement is observed; then they become negatively related.
    - » Shoes and Fall are *dependent* until Pavement is observed; then they become independent.
  - Causality functions to *facilitate communication*, *reduce computation* (by creating sparse networks), and *simplify specification* (by creating networks with fewer links)
- When A is relevant to B in context K we tend to look for a causal explanation:
  - Does A cause B in context K?
  - Does B cause A in context K?
  - Does some other variable C cause A and B?

Causation and Explaining Away

- Alternate causes of an event E are often (approximately) unconditionally independent given some generally understood context. They become *dependent* when E is observed. (Knowing that cause A is true "explains away" other potential causes of E)
- Learning about an effect that could be caused by either of two variables introduces an *informational* dependence between them
  - This type of dependence is called *intercausal dependence*
- Informational dependence is different from causal dependence
  - We can change an effect by intervening to change the cause
    - » We can make the car start if the battery is dead by putting in a new battery
  - We cannot change the state of a variable by changing one on which it depends informationally
    - » If the car won't start and we learn that the starter is bad we infer that the battery is probably OK (intercausal dependence)
    - » But we cannot make the battery OK by destroying the starter

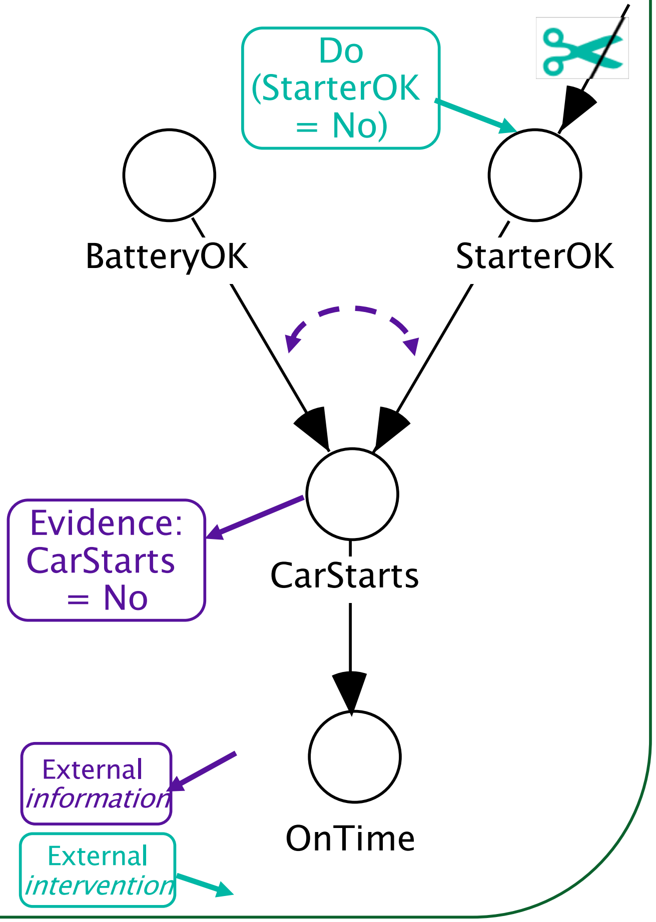


Causality and Arc Direction

- In Bayesian networks arc direction encodes information about conditional independence:
  - A node is conditionally independent of its non-descendents given the values of its parents
  - This property of conditional independence is the only requirement that a DAG must satisfy in a Bayesian network. There need be no relationship between arc direction and
    - direction of inference
    - causality
- In a BN, the predecessors of a node need not be its causes -- but they must shield it from influence of its non-descendents.
- Several authors (especially Pearl) have noted that when causal information is available and arcs are drawn according to causality the resulting independence relationships seem to match our intuitions about how causal relationships behave
  - Drawing arcs in the causal direction can reduce the number of arcs that need to be drawn
  - When available, causal information can be useful for building Bayesian network models, but a causal interpretation is not required.

Causal Bayesian Network

- A **causal Bayesian network** is a Bayesian network in which the arcs represent causal links
- A causal Bayesian network represents *stable* local probabilistic relationships that persist when we intervene to change the world
  - *Do*(*RV=value*) represents an *external intervention* to set a random variable to a given value
  - Intervention performs "surgery" to break links from a random variable's parents
  - Intervention leaves links to children intact
  - Intervention leaves all probability tables intact except for the "Do" RV
- Causation, evidence and interventions
  - An *intervention* causes a change in a random variable's state by a mechanism different from the influence of its other parents
  - *Evidence* is information about the state of a RV
- A causal Bayesian network represents a *family* of probability distributions
  - "Unperturbed" natural distribution
  - Distributions given interventions that could be performed



Unit 2 Outline

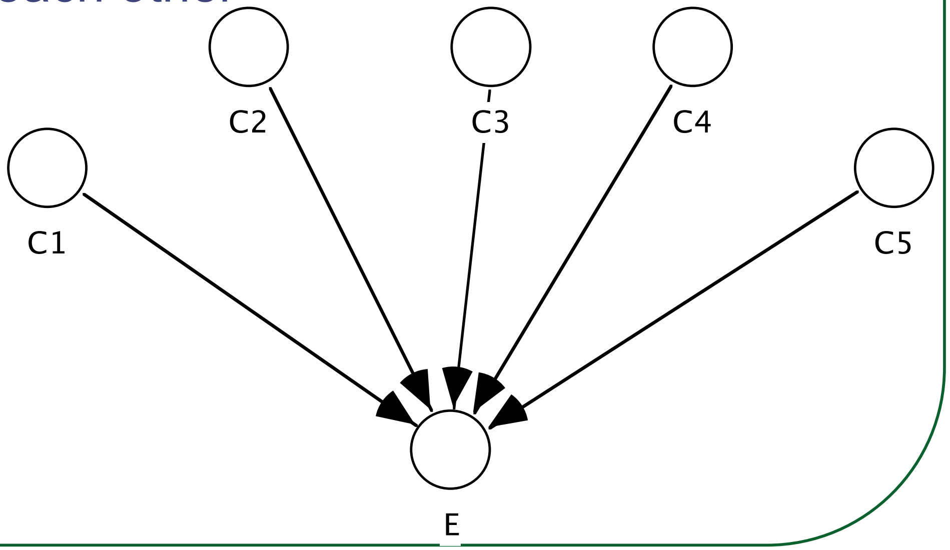
- Graphical Probability Models: Overview
- Graph Theory Basics
- Graphical Probability Models: Formal Definitions
- Node-level Independence

Node-Level Independence

- Independence assumptions are an important tool for simplifying knowledge elicitation and inference
  - We have seen how Bayesian networks use independence to reduce the number of parameters required to specify a probability distribution
  - We will see how the independence assumptions encoded in Bayesian networks reduce the computational complexity of inference
- Bayesian networks are not expressive enough to encode all independence assumptions that can be exploited to:
  - simplify knowledge elicitation
  - reduce computations for inference
- Additional types of independence which can simplify knowledge elicitation and inference:
  - Independence of causal influence (ICI)
    - The mechanism by which one parent variable causes child variable is independent of (1) values of other parent variables and (2) mechanism by which they cause the child
  - Context-specific independence (CSI)
    - Variables may be independent of other variables for some but not all their values
- These kinds of independence act within the local distribution and are not apparent from the graph

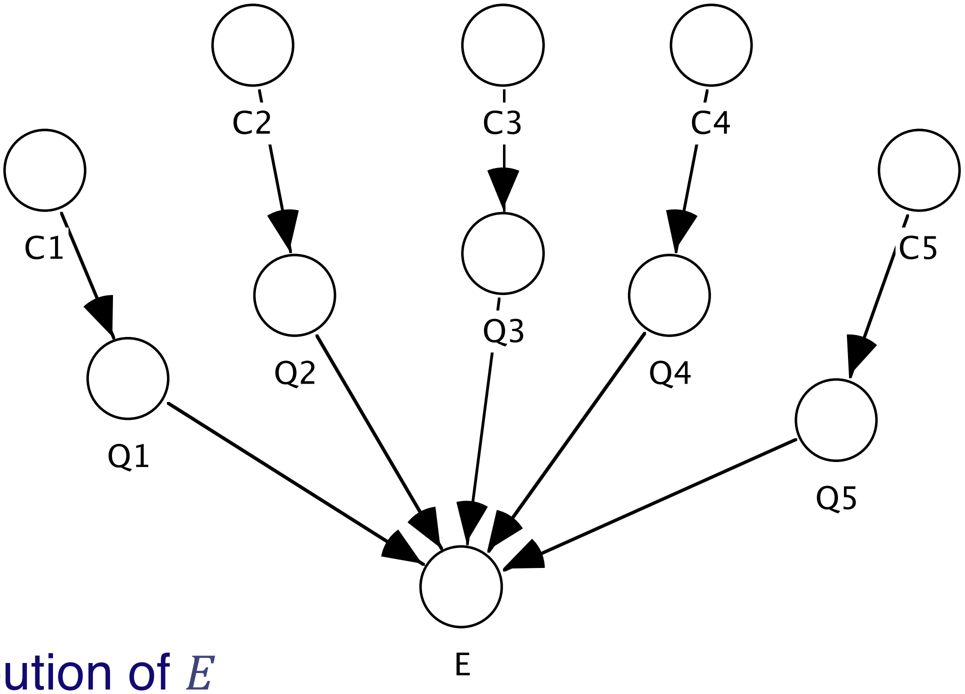
Example of ICI: Noisy-OR

- Problem: When there are multiple causes probability distributions are burdensome to specify and hard to learn
- To specify $P(E|C1, C2, C3, C4, C5)$ we need to elicit or learn
  - 2^5 = 32 probabilities (5 causes with 2 values each)
  - If there were 10 causes this would be 2^{10} = 1024 probabilities to assess
  - If there were 20 causes, it is 2^{20}, or over 1M probabilities 😟
- When applicable, the noisy-OR can simplify assessment & allow learning with much smaller sample sizes
  - Noisy-OR applies when E can be caused by any one of a set of causes which "operate independently" of each other
    - » The mechanism by which $C1$ causes E does not depend on the values of $C2, \ldots, C5$ or the mechanism by which they cause E
    - » Does *not* mean $C1$ is independent of $C2$, etc. (noisy-OR model has nothing to do with whether there are arcs between the C's)



An Equivalent Model

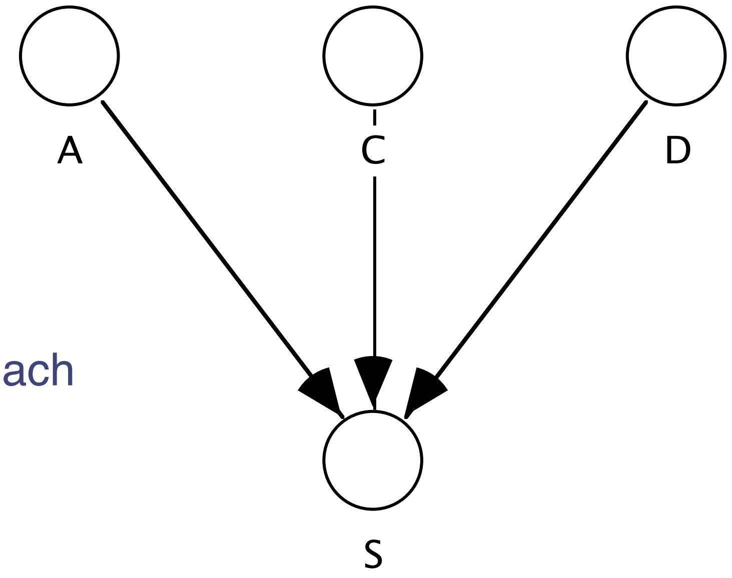
- Define auxillary "trigger" variables Qi (Qi is true if Ci is true and triggers E)
- The C's, Q's and E are related by the following Bayesian network:
- The probability distributions:
  - $P(E=1\,|\,Q1,\ldots,Qn) \;=\; 1$ if at least one of the $Qi \;=1$ and 0 otherwise (E is true if and only if it is caused by one of the C's)
  - $P(Qi=1\,|\,Ci) \;=\; pi$ if $Ci \;=1$ and 0 if $Ci \;=0$ (E can be caused by Ci only if Ci is true; if Ci is true it causes E with probability p_i)



- In this model, the conditional distribution of E given the C’s is the same as the noisy-OR model
  - Independence relationships are now apparent from the graph
- Any ICI model can be reformulated in a similar way

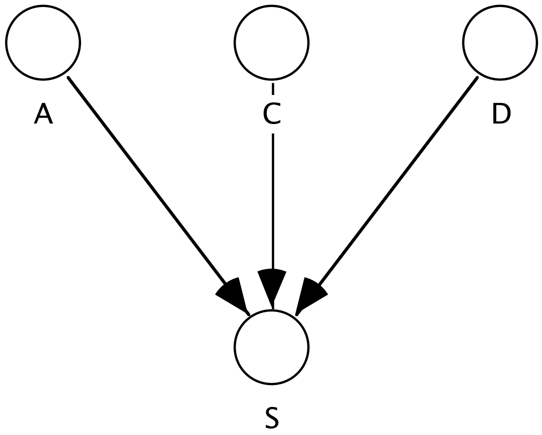
Example of Noisy-OR

- Sneezing can be caused by an allergy (A) , a cold (C), or dust (D) in the air
- The Noisy-OR model:
  - Each cause may or may not "trigger" the effect.
    - Only causes that are true can trigger the effect
    - Causes operate "noisily" - if true, they may or may not trigger the effect
  - There is a "trigger probability" associated with each cause.
    - Allergy triggers sneezing with probability $p_A = .6$
    - Cold triggers sneezing with probability $p_C = .9$
    - Dust triggers sneezing with probability $p_D = .3$
  - Basic assumptions of the "noisy-OR" model:
    - Effect occurs if one or more of its causes has triggered it
    - Whether one cause has triggered the effect is independent of whether another cause has triggered the effect



From Noisy-OR to Local Belief Table

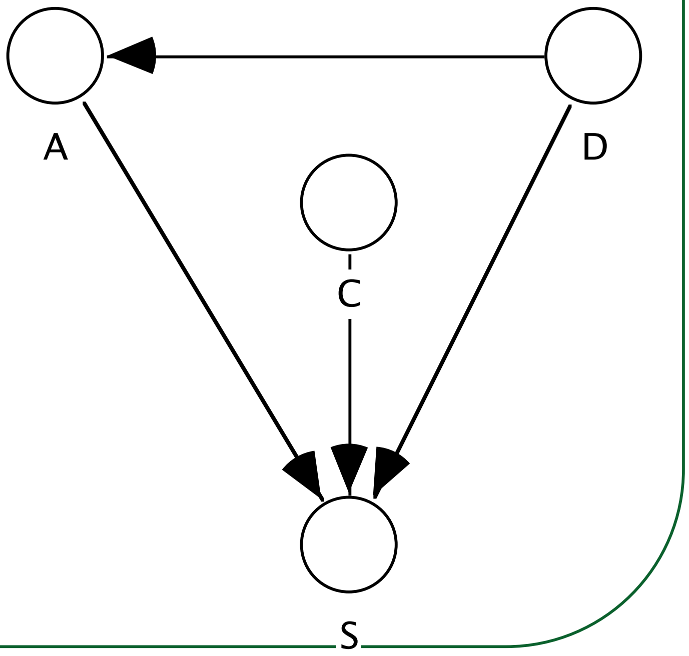
- Noisy-OR simplifies elicitation
  - General model: The distribution P(S|A,C,D) requires 8 probabilities to be elicited (the other 8 can be obtained from the sum-to-1 constraint)
  - Noisy-OR model: The distribution P(S|A,C,D) can be obtained from 3 numbers:
    - » "What is the probability that Effect occurs given Cause i occurs and none of the other causes occur?"
- The ICI assumptions can be used to fill in the probability table [it is more convenient to compute probabilities of S=f; probabilities of S=t can be filled in by using the sum-to-1 constraint]:
  - P(S=f | A=f, C=f, D=f) = 1
  - P(S=f | A=t, C=f, D=f) = (1-.6) = .4
  - P(S=f | A=f, C=t, D=f) = (1-.9) = .1
  - P(S=f | A=f, C=t, D=t) = (1-.3) = .7
  - P(S=f | A=t, C=t, D=f) = (1-.6)(1-.9) = .04
  - P(S=f | A=t, C=f, D=t) = (1-.6)(1-.3) = .28
  - P(S=f | A=f, C=t, D=t) = (1-.9)(1-.3) = .07
  - P(S=f | A=t, C=t, D=t) = (1-.6)(1-.9)(1-.3) = .028



- Many BN packages have native ability to specify noisy-OR
  - Usually specified with a scripting language not a graphical interface
  - In Netica: `P(S | A, C, D) = NoisyOrDist(S,0,A,0.6,C,0.9,D,0.3)`

ICI is NOT Independence of Causes!

- Some literature uses the term “causal independence” instead of ICI, which can generate confusion
  - ICI is a model of the local distribution of a child node given its parents
  - It is the *causal influence* that is independent in ICI, not the causes themselves
  - ICI can apply *whether or not* there are arcs between parent nodes
- Example: Suppose living in an area with a high particulate count caused people to develop allergies
  - There is an arc from D to A in the Bayesian network
  - If the ICI assumption holds, the distribution of S given A, C, and E may still be modeled by a Noisy-OR
- There are no commonly accepted graphical representations for ICI



General Noisy-OR

- Given:
  - n binary "causal events" C1, …, Cn
  - one binary "effect" E
  - Ci and E can take on values 0 (false) or 1 (true)
- Assumptions:
  - For E to happen one of the Ci must be true
  - When Ci is true there is probability pi that Ci causes E
    - This probability does not depend on the other Cj
- **D2.26**: A local distribution P(E|C1,…,Cn) is a Noisy-OR if:

$$P(E=0 | C1,\ldots Cn) = \prod_{C_i=1}(1\text{-}p_i)$$
$$P(E=1 | C1,\ldots Cn) = 1 - \prod_{C_i=1}(1\text{-}p_i) \qquad \textbf{(2.26)}$$

- Elicitation of Noisy-OR:
  - General model for P(E | C1, …, Cn) requires eliciting 2^n probabilities (the other 2^n can be determined from the sum-to-1 constraint)
  - Noisy-OR requires n probabilities p_1, …, p_n from which the others can be computed
  - Elicitation of p_i: “What is the probability that E occurs when Ci is true and all other causes are false?”

Noisy-OR with Leak

- In the noisy-OR model as defined above, E has probability zero of occurring if none of its causes occurs.
- Often we want to have some "background" probability that E occurs in the absence of any of the Ci.
- The noisy-OR equations can be modified to include a “leak probability”:

$$P(E{=}0 \mid C1,\ldots Cn) = (1-p_0)\prod_{C_i=1}(1\text{-}p_i)$$

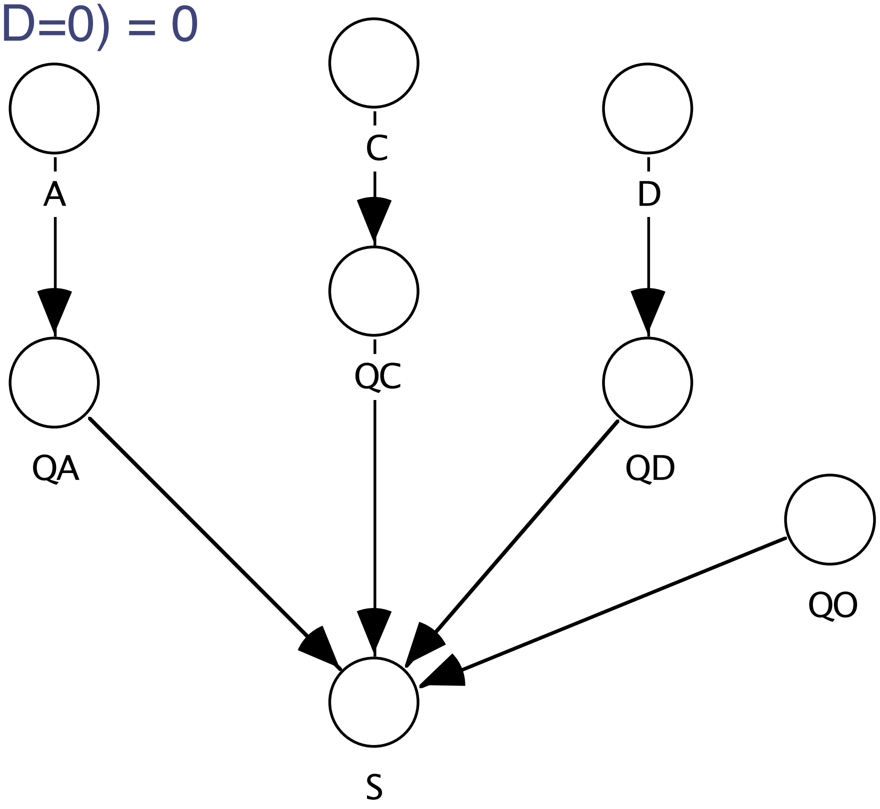
$$P(E{=}1 \mid C1,\ldots Cn) = 1-(1-p_0)\prod_{C_i=1}(1\text{-}p_i)$$

(2.26a)

  - The model can be specified by n+1 numbers, p_0, p_1, …,p_n
  - It is more convenient to assess the following numbers and solve for the pi:
    - » Leak probability” p_0 is the probability that E occurs when none of the name causes is true
    - » Effect probability $a_i = p_0 + (1\text{-}p_0)p_i$ is the probability that E occurs when Ci is the only cause that is true
  - This is formally equivalent to a noisy-OR model with n+1 causes, where "background cause" B is independent of the other causes and $p_0 = P(B=t)P(QB=t|B=t)$

Sneezing Example

- Sneezing is a deterministic function of its parent nodes: it is true if any of them is true
  - $P(Q_A=1|A=1) = p_{A;}$ $P(Q_A|A=0) = 0$
  - $P(Q_C=1|C=1) = p_{C;}$ $P(Q_C|C=0) = 0$
  - $P(Q_D=1|D=1) = p_{D;}$ $P(Q_D|D=0) = 0$
  - $P(Q_O) = p_O$

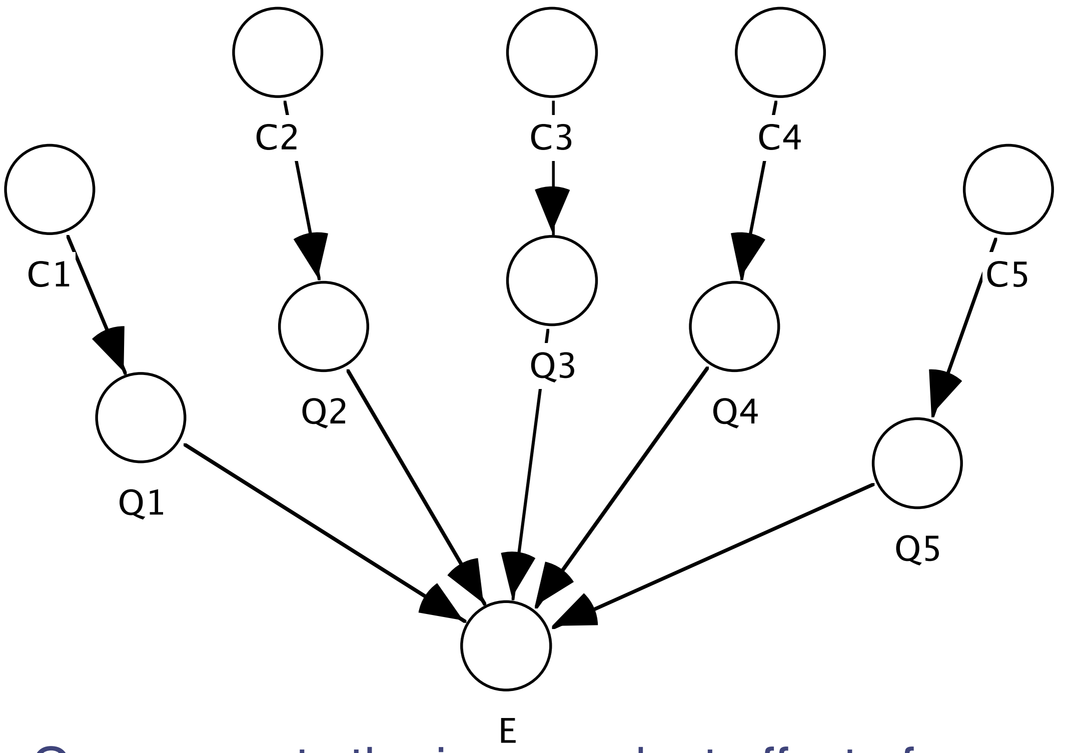


What Have We Gained?

- Specification efficiency:
  - We can specify the local distribution for an effect with n causes using n+1 instead of 2^n numbers
    - probability that effect occurs in the absence of any of the explicitly represented causes (p_0)
    - probability that each cause is sufficient to cause the effect (p_i)
  - Fewer questions to ask the expert or learn from data
    - Do not ask the expert directly about p_i. Ask about $a_i = P(E=1|C_i=1$, all other $C_j=0)$ and solve for p_i
    - We still haven't reduced the number of elements needed in the computer to represent the local model (there are ways to do this)
- Implementations can exploit ICI to speed up computation
- What good is the model with the auxiliary variables?
  - If your software package doesn't allow direct calculation, the auxiliary variables method allows you to specify a noisy-OR
  - The auxiliary variable model portrays a "hidden variables" causal mechanism that may make it easier for the modeler to understand what the independence assumptions mean
  - This method can be generalized to other "noisy deterministic function" models for which most software packages do not have a direct way to specify

General ICI Model

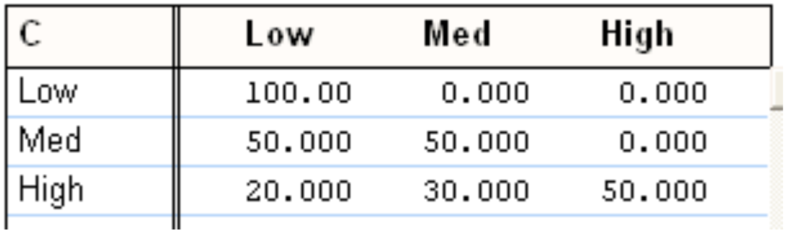
- Given n causes C_i and an effect E (not necessarily binary), we can specify a model:



  - Each Q_i represents the independent effect of cause C_i
  - E is a deterministic function of the Q_i : $E=f(Q_1,\dots,Q_n)$
  - Examples:
    - noisy Boolean functions (OR, AND, XOR)
    - noisy adder
    - Noisy min/max

Example: Noisy-MAX

- All causes C_i and the effect E have the same number of states
- The causes and the effect are ordered from least to most severe (e.g., low/medium/high for 3-valued variables).
  - They need not all have the same state spaces, but the same number of values must be the same, and they need to be ordered lowest to highest.
- We specify a probability distribution for the trigger given each level of the cause. Generally, the trigger distribution will be more severe (i.e., have higher probabilities on states with more severity) for more severe states of the cause.
  - For example:



C	Low	Med	High
Low	100.00	0.000	0.000
Med	50.000	50.000	0.000
High	20.000	30.000	50.000

- The effect node E is equal with probability 1 to the maximum of the trigger nodes Q_i.

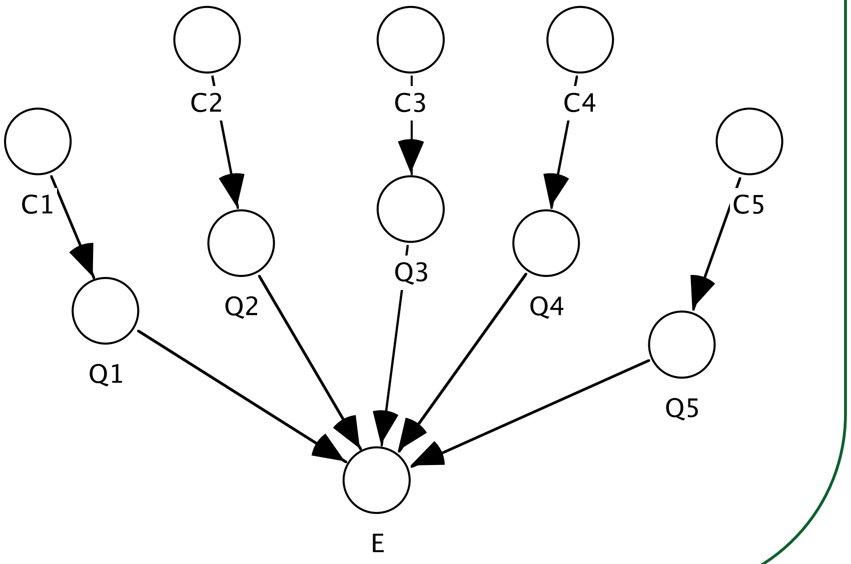
See Netica documentation for Noisy-OR, Noisy-Max and Noisy-Adder (available for download from Blackboard)

ICI: Formal Definition

- **D2.27**: A local distribution P(E|C$_1$,…, C$_n$) exhibits *independence of causal influence* if there are random variables Q$_1$, …, Q$_n$ such that:

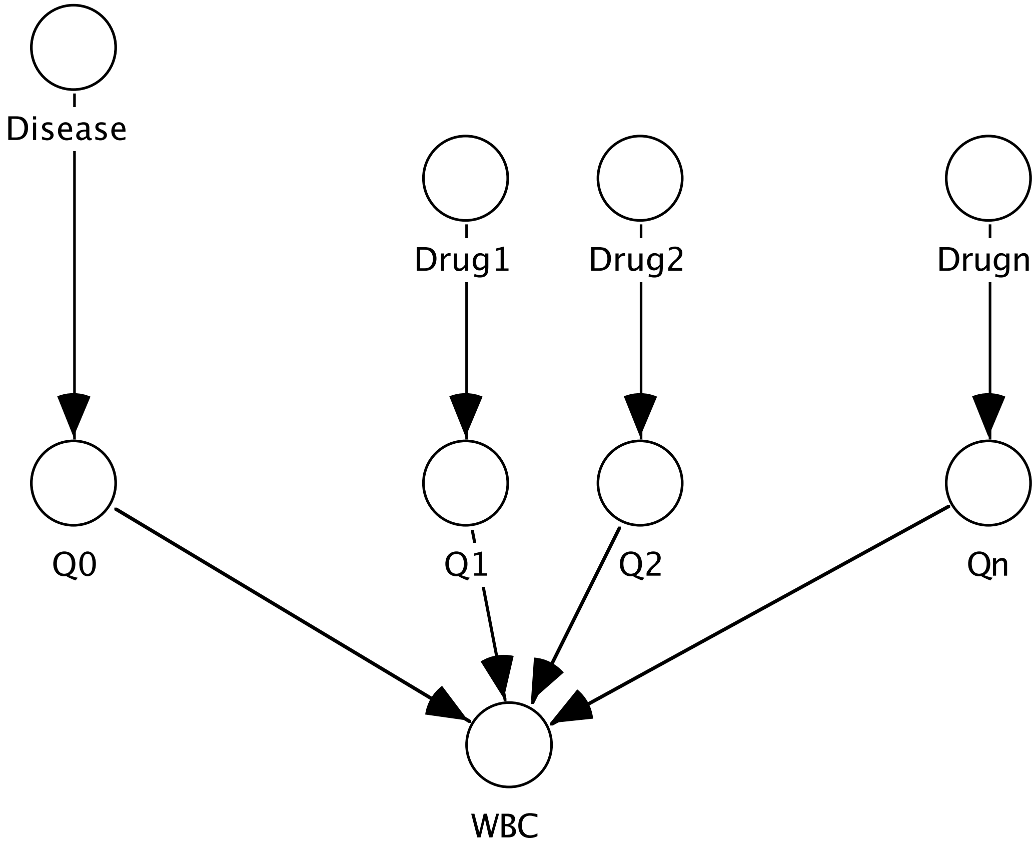
$$\mathrm{P(E{=}e\,|\,C_1,\ldots C_n)} = \sum_{q_1,\ldots,q_n} \left(1_{[e=f(q_1,\ldots,q_n)]} \prod_{k=1}^{n} P(Q_i = q_i \mid C_i) \right)$$

  - $1_{[e=f]}$ is an "indicator" function that has value 1 if e=f and 0 otherwise
  - E is a deterministic function E=f(Q$_1$, …, Q$_n$) of the random variables Q$_1$, …, Q$_n$
  - Q$_i$ depends on C$_i$ but is independent of the other causes
  - Conditional on C$_i$, Q$_i$ is independent of C$_j$ and Q$_j$ for j≠i



Example: Noisy-Adder

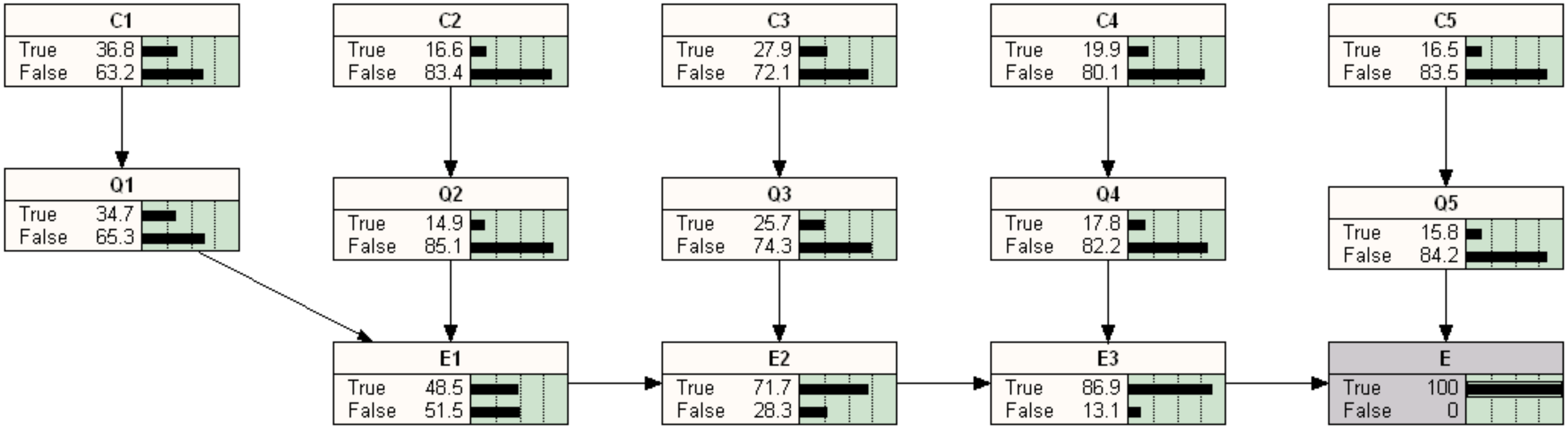
- The model:
  - Each C_i is Boolean (value 0 or 1)
  - Each Q_i can take on values in the range [-q, q]
  - If $C_i = 0$ then $Q_i = 0$ with probability 1
  - Adjustable parameters: $P(Q_i=k|C_i=1) = r_{ik}$
  - $E = Q_1 + Q_2 + \cdots + Q_n$
  - We can add an additional Q_0 to represent effect on E of “unmodeled causes”
- Example: Effect of drugs on patient’s white blood count

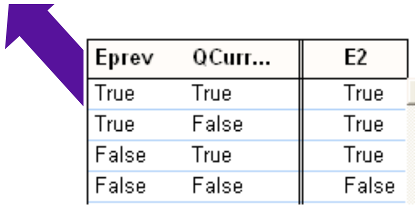


Note: The standard normal linear regression model is an ICI model (effect is weighted sum of values of causes plus a normally distributed error term)

An Alternate Representation for ICI

- We can represent many common ICI models as process in which effects of the trigger variables accumulate sequentially
- This representation has computational advantages
  - No belief table has more than n^2 rows, where n is the number of states in the C_i and E
  - Inference algorithms can exploit this representation to gain efficiency





Eprev	QCurr...	E2
True	True	True
True	False	True
False	True	True
False	False	False

Summary: Independence of Causal Influence

- ICI means that the impact of a parent variable on the probability distribution of the child variable is independent of the values of the other parent variables
  - ICI can simplify both specification and inference
- ICI is *not* the same as independence of the parent variables (causes)
  - You can't see ICI on the graph relating parent to child variables
  - Independence of the parent variables is shown by lack of an arc in the graph
  - ICI may apply *whether or not* parent variables are independent of each other
- Some examples of different kinds of independence:
  - *Not ICI:* Causes of sneezing are independent: Having a cold does not make allergies more likely (no arc between cold and allergy in the Bayesian network)
  - *Not ICI:* Causes of sneezing are dependent: Living in an area with lots of particulates in the air causes person to develop allergies (arc should go from dust to allergy in the Bayesian network)
  - *Not ICI:* Heredity and dust do not cause allergies independently: A person with a genetic tendency toward allergies is more likely to have allergies triggered by dust than a person with no genetic tendency to have allergies (Noisy-OR is not OK for this relationship)
  - *ICI:* Allergies and colds cause sneezing independently: Having a cold does not make it more likely that an allergy will trigger sneezing. Whether the cold triggers sneezing does not affect whether the allergy triggers sneezing. (Noisy-OR model is OK for relationship between allergies, colds, and sneezing)
- *Note: All these relationships can be tested / estimated statistically*

Context-Specific Independence

- An arc from A to B in a Bayesian network means that the probability distribution for B depends on the value of A
- Context-specific independence occurs when random variables are independent for some but not all of their values
  - When weather is sunny detection probability is independent of sensor type but when weather is cloudy the radar sensor performs better than the optical sensor
- Like ICI, context-specific independence
  - Cannot be represented by the directed graph structure of a Bayesian network
  - Can be exploited to simplify elicitation
  - Can be exploited to simplify inference
  - Can be exploited to make learning from data more efficient

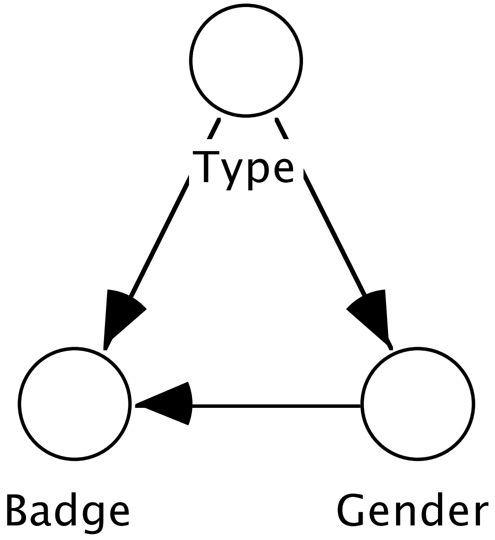
Example

- A guard of a secured building expects three types of person to approach the building's entrance: workers in the building, approved visitors, and spies. As a person approaches the building, the guard notes the person's gender and whether or not the person is wearing a badge. Spies are mostly men. Spies always wear badges in order to fool the guard. Visitors don't wear badges because they don't have one. Female workers tend to wear badges more often than do male workers. The task of the guard is to identify the type of person approaching the building.
  - The variables:

    T = person type (t_W=Worker, t_V=Visitor, t_S=Spy)

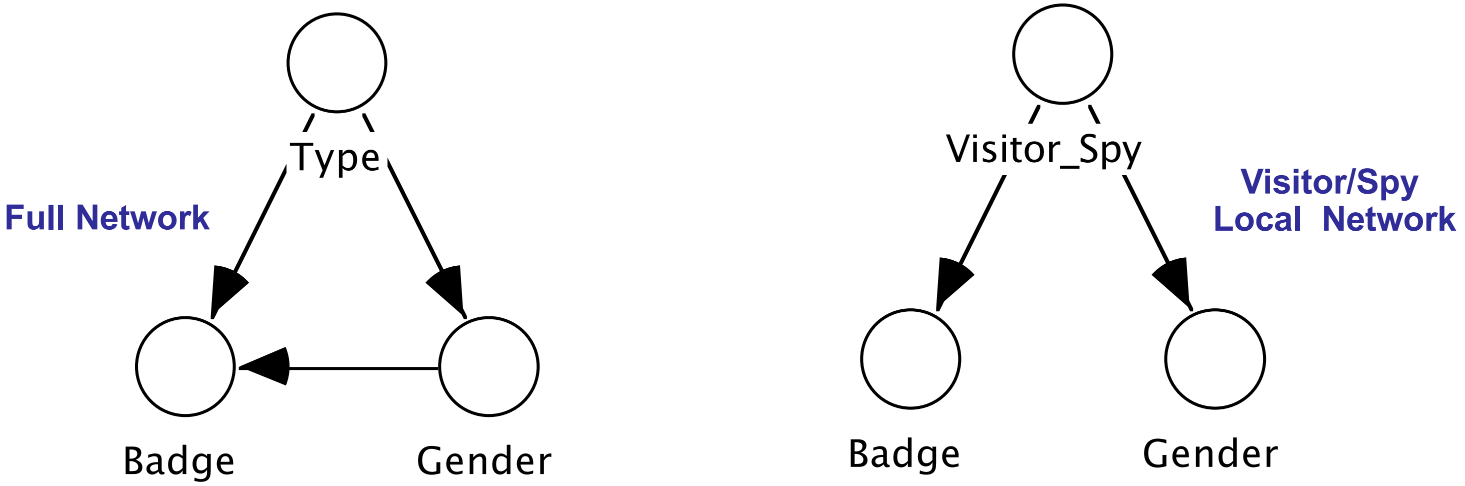
    G = gender (g_M=Male, g_F=Female)

    B = badge (b_Y=Yes, b_N=No)
  - The Bayesian network does not encode all the independence information the story tells us



Context-Specific Independence

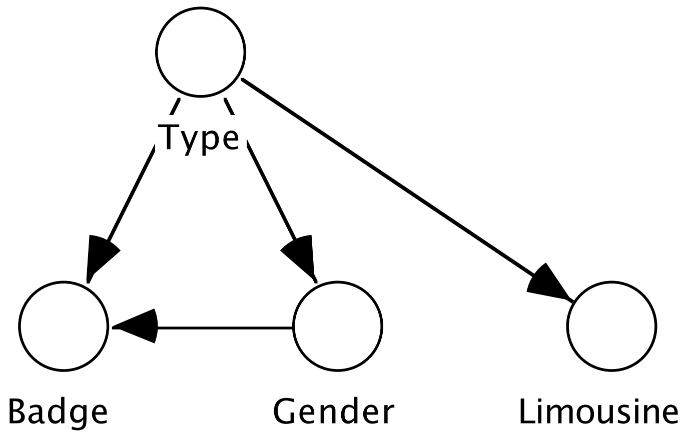
- We can specify a “local network” to distinguish between visitors and spies:

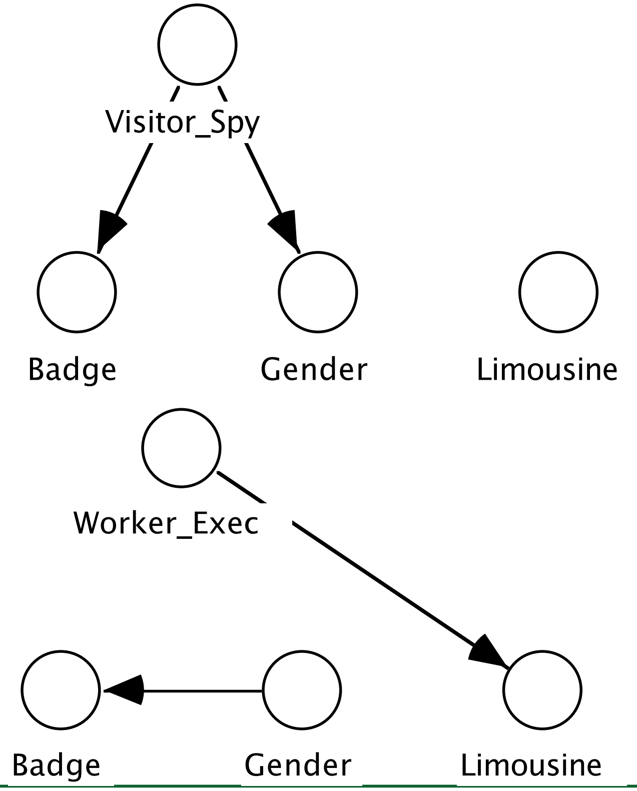


- The variables B and G are independent given some values (visitor and spy) of their parent variable but not others (worker)
  - An arc from G to B is necessary in the full network to encode the dependency of gender on badge-wearing for workers
  - Full network cannot represent Independence G and B for visitors and spies
  - If we restrict attention to visitors and spies we can represent this independence
  - This type of independence is called *context-specific independence*)
  - Context-specific independence can be exploited to reduce elicitation significantly and make inference more efficient
  - Most software does not support CSI except via scripting languages

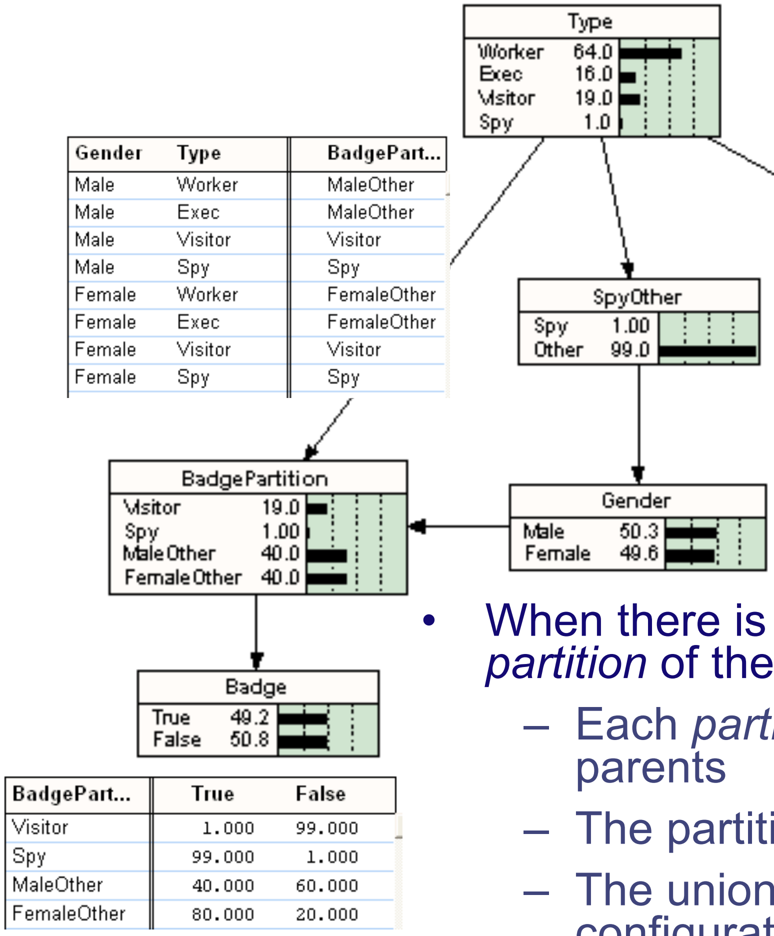
Example Revisited

- The guard of the secured building now expects *four* types of persons to approach the building’ s entrance: executives, regular workers, approved visitors, and spies. The guard notes gender, badge-wearing, and whether or not the person arrives in a limousine. We assume that only executives arrive in limousines and that male and female executives wear badges just as do regular workers.





Using Partitions to Simplify Specification



Gender	Type	BadgePart...
Male	Worker	MaleOther
Male	Exec	MaleOther
Male	Visitor	Visitor
Male	Spy	Spy
Female	Worker	FemaleOther
Female	Exec	FemaleOther
Female	Visitor	Visitor
Female	Spy	Spy

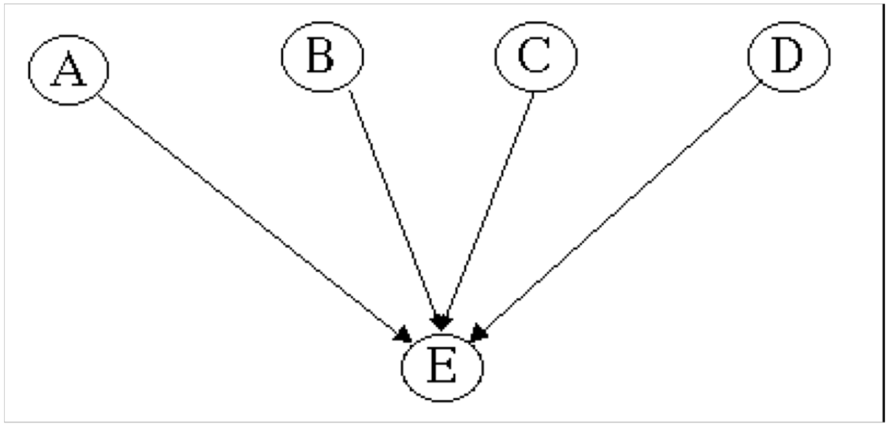
BadgePart...	True	False
Visitor	1.000	99.000
Spy	99.000	1.000
MaleOther	40.000	60.000
FemaleOther	80.000	20.000

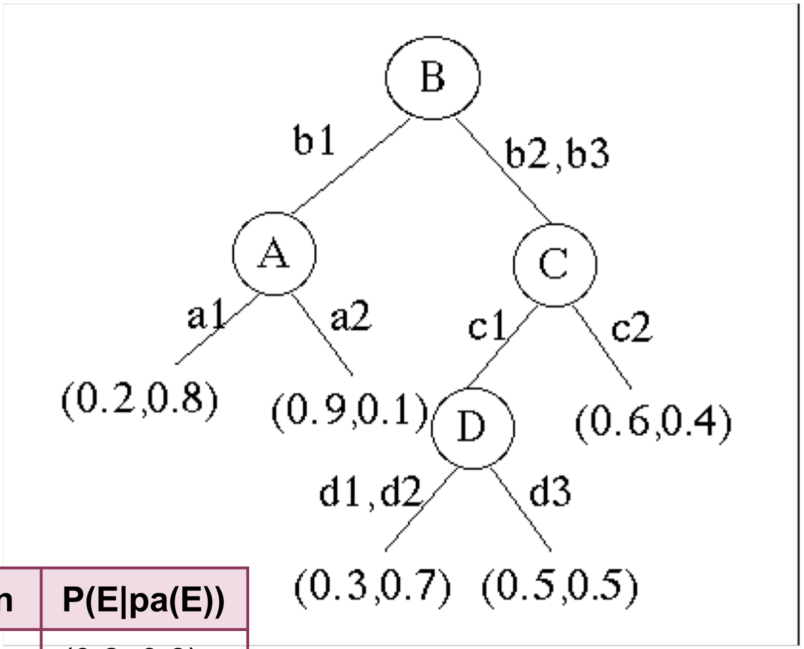
- When there is context-specific independence, we can specify a *partition* of the state space of a node's parent configurations:
  - Each *partition element* is a set of configurations of the node's parents
  - The partition elements are disjoint
  - The union of the partition elements is the entire set of configurations
  - The conditional distribution of the child node is the same for each configuration in a partition element
- We only need to specify one distribution per partition element
- Partitions simplify knowledge elicitation and learning

Local Decision Trees

- Example network:
  - #A = #C = #E = 2, #B = #D = 3
  - To specify distribution for E requires 36 probability judgments

- Example local decision tree for specifying distribution of E:
  - Requires 5 probability judgments

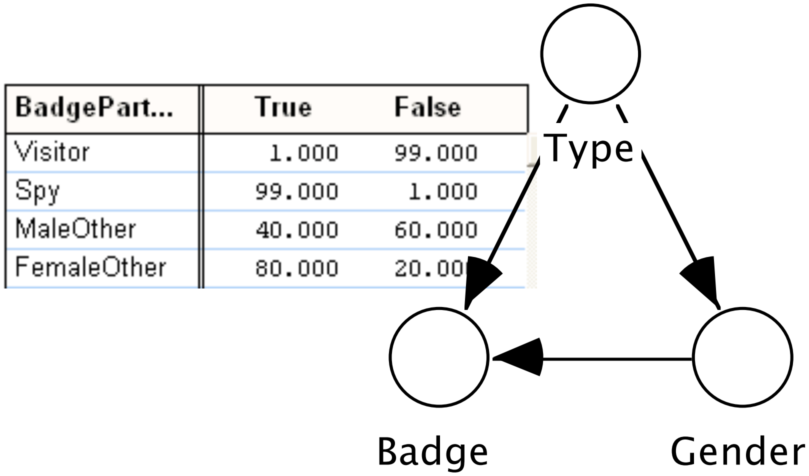




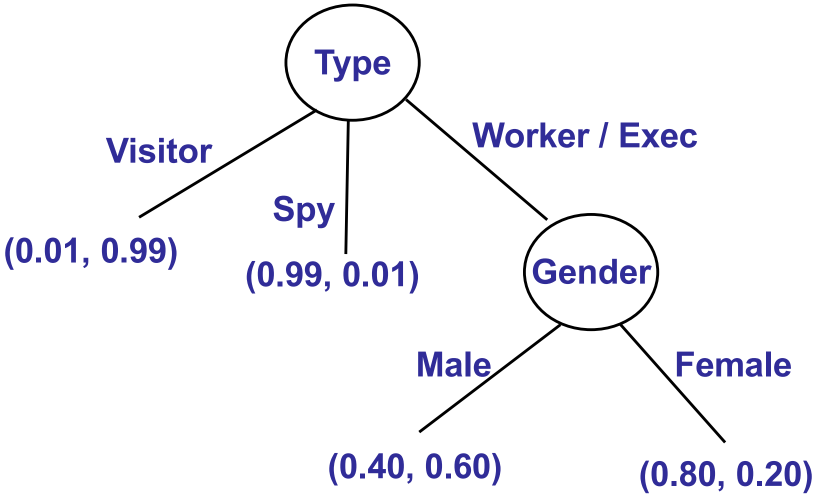
A local decision tree is a simple and natural way to specify partitions for specifying a random variable's local distribution

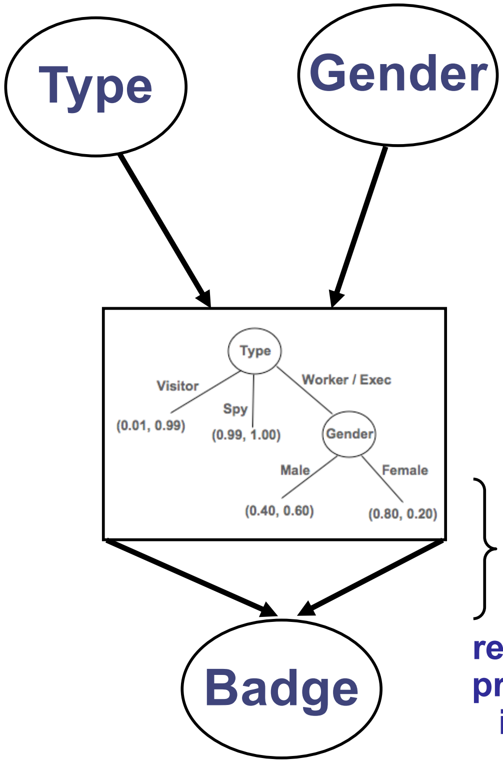
Parent configuration	P(E|pa(E))
B=b1, A=a1	(0.2, 0.8)
B=b1, A=a2	(0.9, 0.1)
B≠b1, C=c1, D≠d3	(0.3, 0.7)
B≠b1, C=c1, D=d3	(0.5, 0.5)
B≠b1, C=c2	(0.6, 0.4)

Local Decision Tree for Badge Partition



BadgePart...	True	False
Visitor	1.000	99.000
Spy	99.000	1.000
MaleOther	40.000	60.000
FemaleOther	80.000	20.00



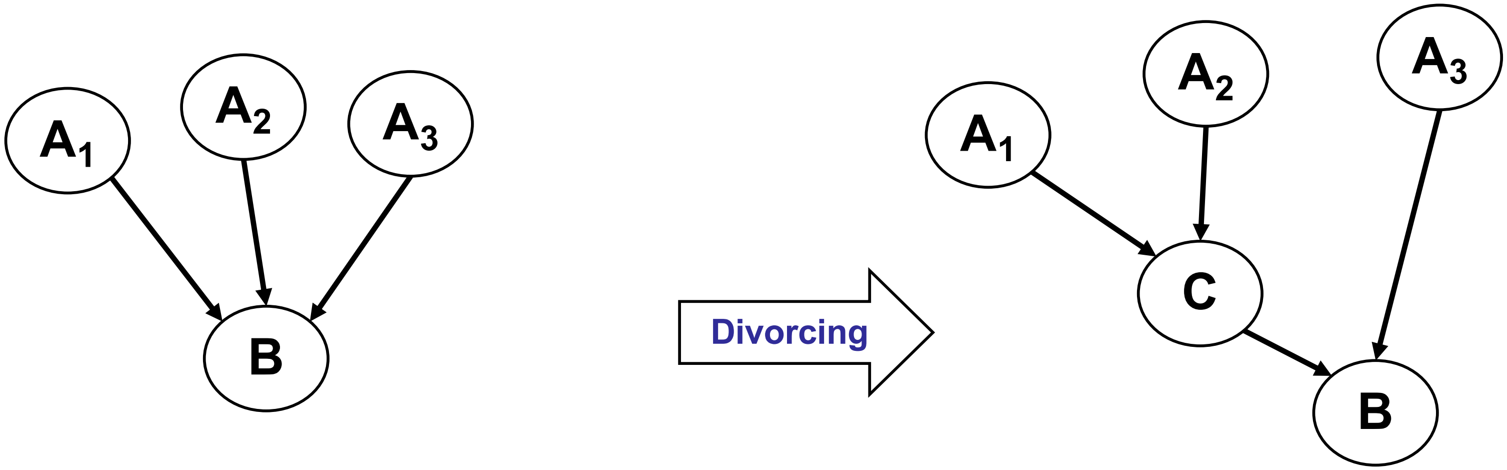


This link represents logical rules to sort parent configurations into the right bin

This link represents a probabilistic influence

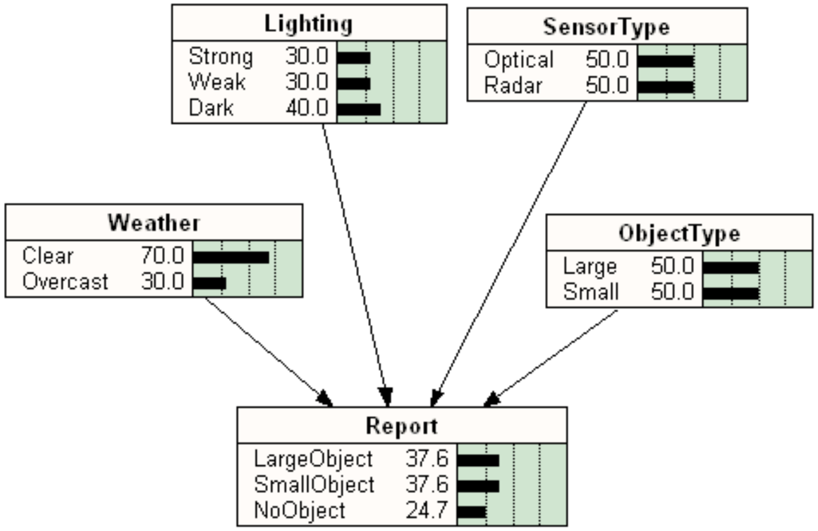
Divorcing

- Divorcing is a trick to express partitions compactly in a general purpose BN package that does not directly support partitions
- Example:
  - B has parents A_1, A_2, A_3
  - C = {c_1, .., c_m} is a partition of $A_1 \times A_2$
- We use divorcing when an asymmetry partition is “rectangular”
  - Partitioning of A_1 and A_2 does not depend on value of A_3

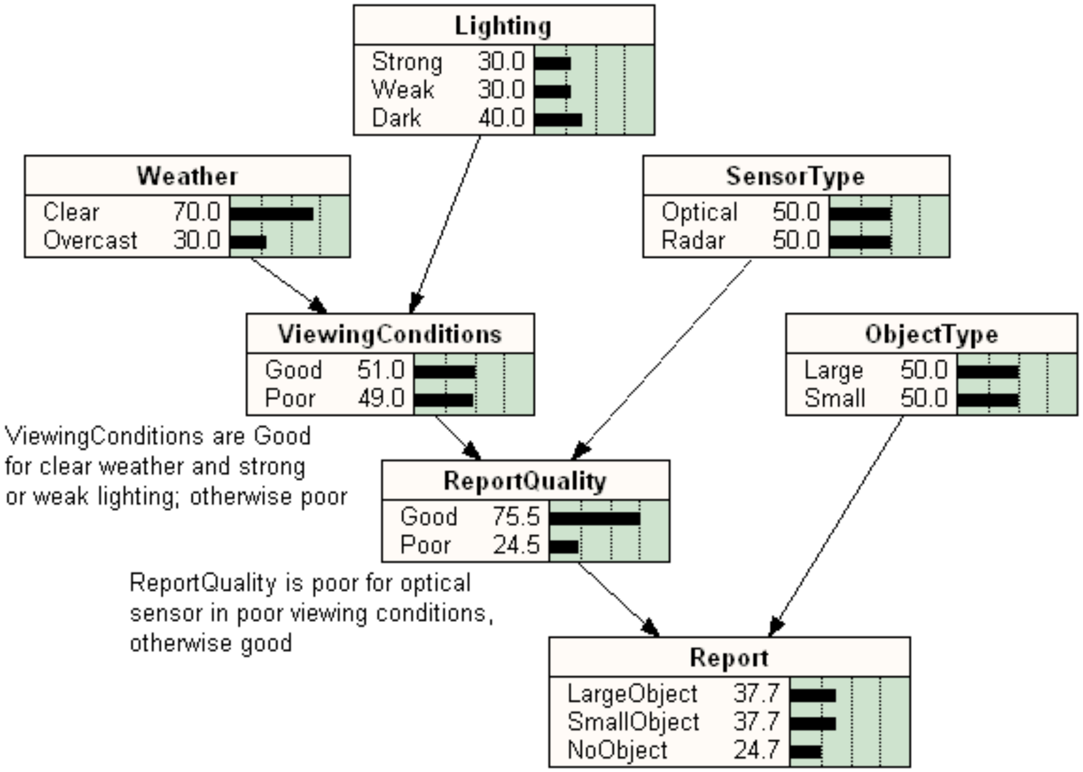


- For some problems, we can achieve considerable savings in both specification and computation

Divorcing Example



There are 24 rows in the belief table for Report



ViewingConditions are Good for clear weather and strong or weak lighting; otherwise poor

ReportQuality is poor for optical sensor in poor viewing conditions, otherwise good

There are 4 rows in the belief table for Report

- Partitions and divorcing can dramatically reduce the number of probabilities to be specified
- Sometimes approximating a distribution by one with context-specific independence is justified by computational or elicitation considerations
- It is important to evaluate the sensitivity of results to approximation error

Local Expression Languages

- A Bayesian network must represent a local distribution for each combination of parent variables
  - A local distribution is a function from configurations of the parent to probability distributions for the child
  - A belief table lists a conditional probability for each parent-child configuration
  - We have examined several more compact representations of local distributions
  - Representation languages should provide a means to encode compact representations
- Inference algorithms can be modified to exploit properties of the local interaction model, resulting in a speedup of inference
- A local expression language can be used to encode:
  - Independence of causal influence models (noisy-OR, noisy adder, etc.)
  - Context-specific independence
  - General functional relationships within local model
    - » P(Xi | P(Xi)) = f(Xi, P(Xi), θ)
    - » E.g., parametric models such as normal, log-normal, logistic, exponential, …
- Parameterized models can improve efficiency of learning local distributions from data
- Many BN packages have local expression languages

Summary and Synthesis

- Graphs provide a parsimonious and understandable language for expressing knowledge about uncertain phenomena
  - Graph represents dependence relationships
  - Numbers represent strength of relationship
- Graphical probability models have made it feasible to represent knowledge about uncertain phenomena in the form of realistically complex probability models for general purpose computing applications
  - Feasible knowledge engineering
  - Tractable computation
  - Tractable learning (to be covered later)
- Along with graph-level independence, additional within-node independence constraints further simplify both model specification and inference
  - Independence of causal influence
  - Context-specific independence

References for Unit 2

- Bayesian networks and graph theory
  - Korb, K. and Nicholson, A. *Bayesian Artificial Intelligence*, Chapman&Hall, 2003. Chapter 2
  - Neapolitan, R. *Learning Bayesian Networks*. Prentice Hall, 2003. Chapter 2
  - Pearl, J. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann, 1988. The first book popularizing Bayesian networks.
- Factor graphs
  - Kschischang, Frank R.; Frey, Brendan J.; Loeliger, Hans-Andrea (2001), "Factor Graphs and the Sum-Product Algorithm", IEEE Transactions on Information Theory, 47 (2): 498–519
- Graphical probability models (general)
  - Cowell, R.G., Dawid, A.P., Lauritzen, S.P., and Spiegelhalter, D.J. *Probabilistic Models and Expert Systems,* Springer, 1999. Chapters 4, 5 and 6.
- Independence of Causal Influence –
  - DE.Heckerman, Causal Independence for Knowledge Acquisition and Inference, in Heckerman, D. and Mamdani, A. (eds.) *Uncertainty in Artificial Intelligence: Proceedings of the Seventh Conference*, pp 122-127, San Mateo, CA: Morgan Kaufmann, 1993
- Context-Specific Independence –
  - Boutilier, C., Friedman, N., Goldszmidt, M. and Koller, D. Context-specific Independence in Bayesian Networks, *Uncertainty in Artificial Intelligence: Proceedings of the Twelfth Conference*, San Mateo, CA: Morgan Kaufmann, 1996
  - Geiger, D. and Heckerman, D., Advances in probabilistic reasoning. in P. Smets, and P.P. Bonissone (eds.) *Uncertainty in Artificial Intelligence: Proceedings of the Seventh Conference*, San Mateo, CA: Morgan Kaufmann, 1991.
  - N.Friedman,M.Goldszmidt, Learning Bayesian Networks with local structure, In Horvitz, E. and Jensen, F. (eds.) *Uncertainty in Artificial Intelligence: Proceedings of the Fifteenth Conference*, pp 252-262, San Mateo, CA: Morgan Kaufmann, 1996