CMSC 726
Lecture 19:Graphical Models

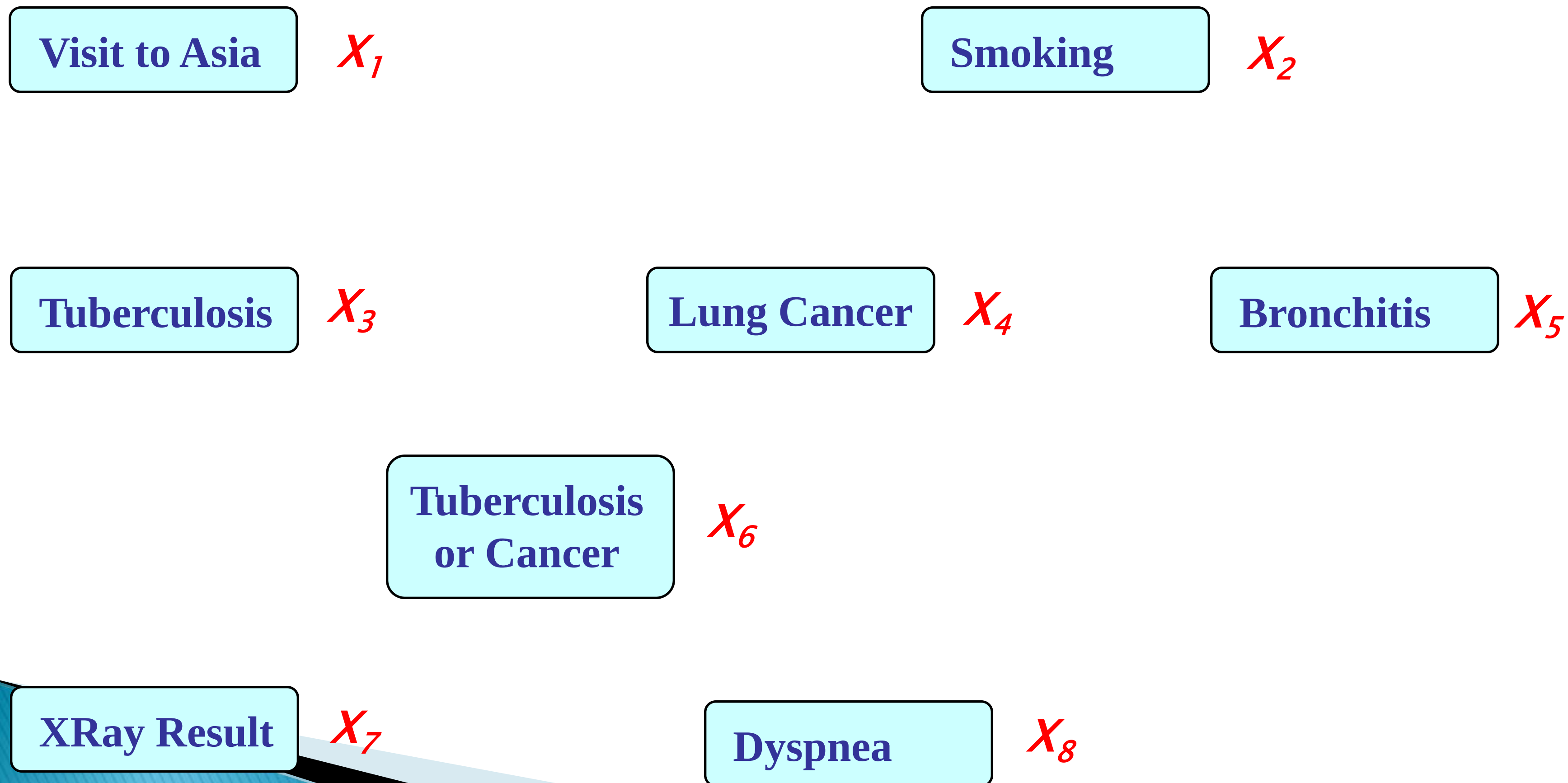
Lise Getoor

November 9, 2010

ACKNOWLEDGEMENTS: The material in this course is a synthesis of materials from many sources, including: Hal Daume III, Mark Drezde, Carlos Guestrin, Andrew Ng, Ben Taskar, **Eric Xing**, and others. I am very grateful for their generous sharing of insights and materials.

What is a graphical model?
--- example from medical diagnostics

- A possible world for a patient with lung problem:



Recap of Basic Prob. Concepts

- Representation: what is the joint probability dist. on multiple variables?

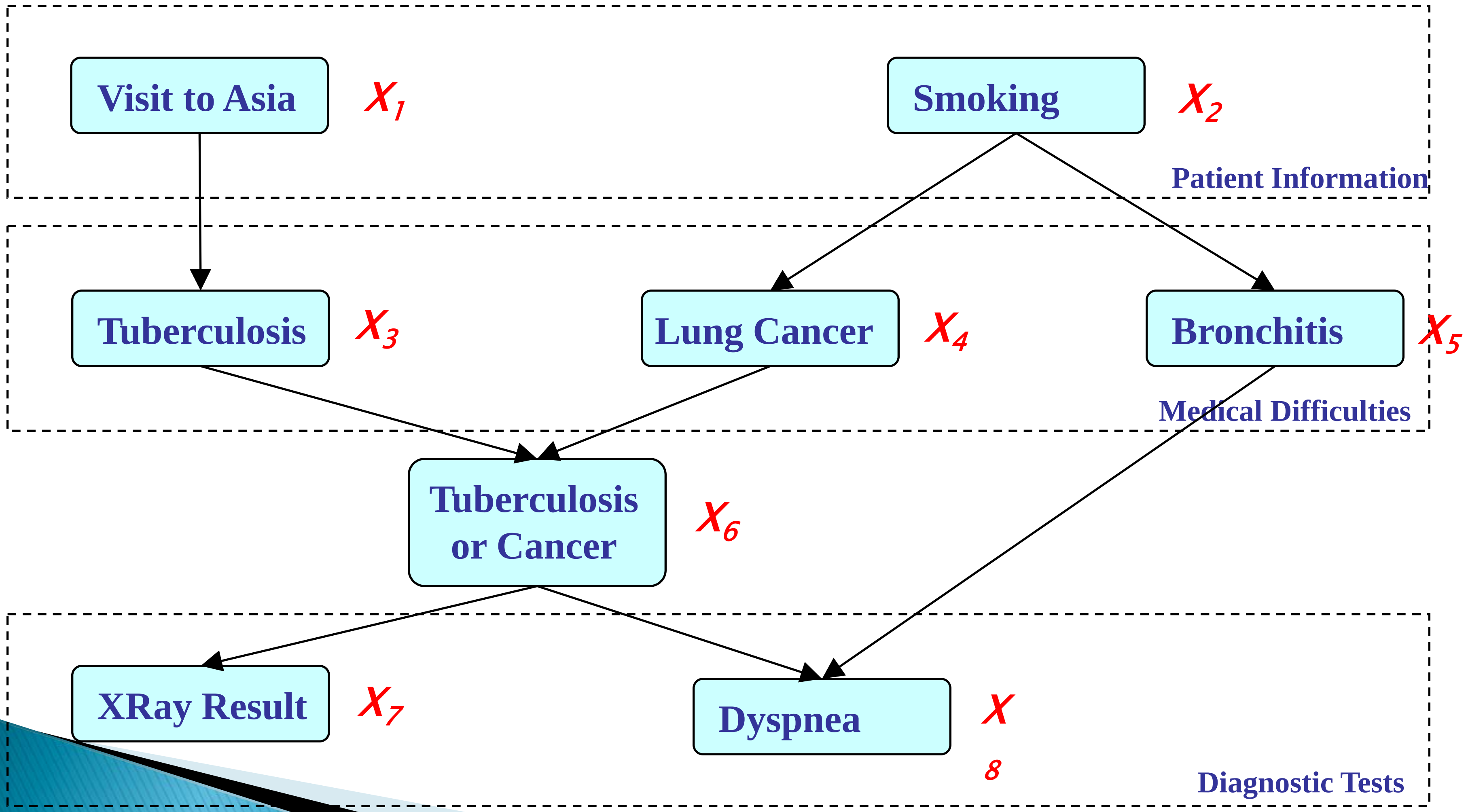
$$P(X_1, X_2, X_3, X_4, X_5, X_6, X_7, X_8,)$$

  - How many state configurations in total? --- 2^8
  - Do they all needed to be represented?
  - Do we get any scientific/medical insight?

- Learning: where do we get all these probabilities?
  - Maximum-likelihood estimation? but how much data do we need?
  - Where do we put domain knowledge in terms of plausible relationships between variables, and plausible values of the probabilities?

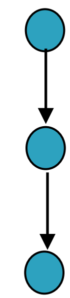
- Inference: If not all variables are observable, how to compute the conditional distribution of latent variables given evidence?

Dependencies among variables



Probabilistic Graphical Models

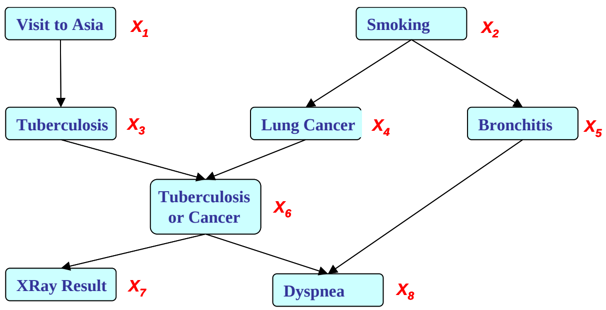
- Represent dependency structure with a graph
  - Node <-> random variable
  - Edges encode dependencies
    - Absence of edge -> conditional independence
  - Directed and undirected versions



- Why is this useful?
  - A language for communication
  - A language for computation
  - A language for development
- Origins:
  - Wright 1920's
  - Independently developed by Spiegelhalter and Lauritzen in statistics and Pearl in computer science in the late 1980's

Probabilistic Graphical Models, cont.

- If X_i's are **conditionally independent** (as described by a **PGM**), the joint can be factored to a product of simpler terms, e.g.,



$$P(X_1, X_2, X_3, X_4, X_5, X_6, X_7, X_8)$$

$$= P(X_1)\, P(X_2)\, P(X_3 | X_1)\, P(X_4 | X_2)\, P(X_5 | X_2)\, P(X_6 | X_3, X_4)\, P(X_7 | X_6)\, P(X_8 | X_5, X_6)$$

- Why favor a PGM?
  - Representation cost: how many probability statements are needed?

    2+2+4+4+4+8+4+8=36, an 8-fold reduction from 2^8!

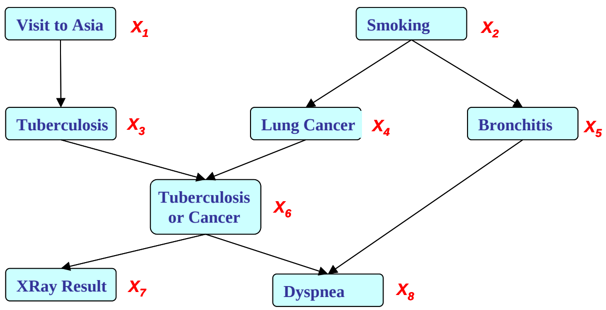
  - Algorithms for systematic and efficient inference/learning computation
    - Exploring the graph structure and probabilistic (e.g., Bayesian, Markovian) semantics
  - Incorporation of domain knowledge and causal (logical) structures

Two types of GMs

- Directed edges give causal relationships (**Bayesian Network** or **Directed Graphical Model**):

- Undirected edges simply give (physical or symmetric) correlations between variables (**Markov Random Field** or **Undirected Graphical model**):

Bayesian Network: Factorization Theorem



$$P(X_1, X_2, X_3, X_4, X_5, X_6, X_7, X_8)$$

$$= P(X_1)\, P(X_2)\, P(X_3 | X_1)\, P(X_4 | X_2)\, P(X_5 | X_2)\, P(X_6 | X_3, X_4)\, P(X_7 | X_6)\, P(X_8 | X_5, X_6)$$

- **Theorem:**

  Given a DAG, The most general form of the probability distribution that is consistent with the graph factors according to "node given its parents":

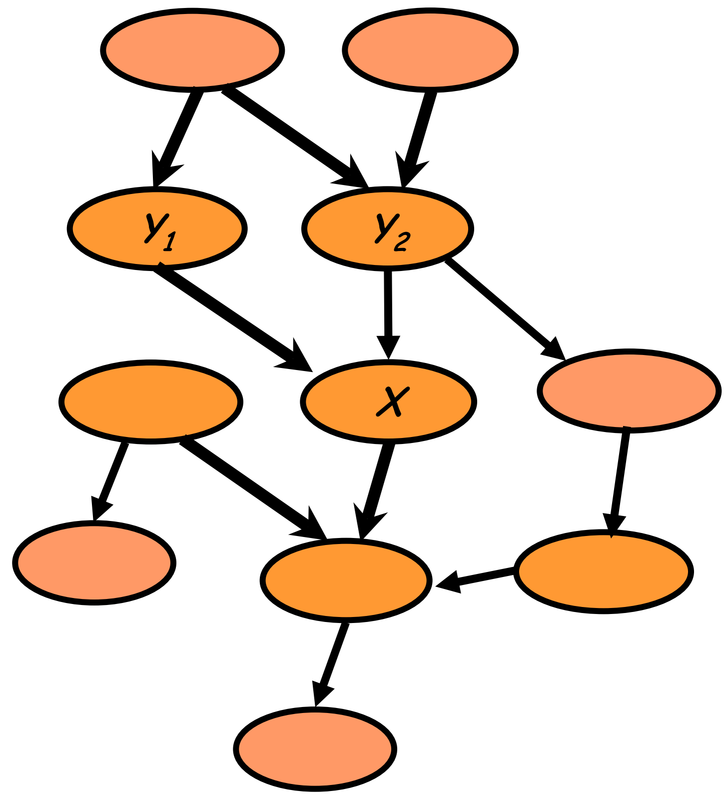
$$P(\mathbf{X}) = \prod_i P(X_i \mid \mathbf{X}_{\pi_i})$$

  where $\mathbf{X}_{\pi_i}$ is the set of parents of x_i.

Bayesian Network: Generative Model

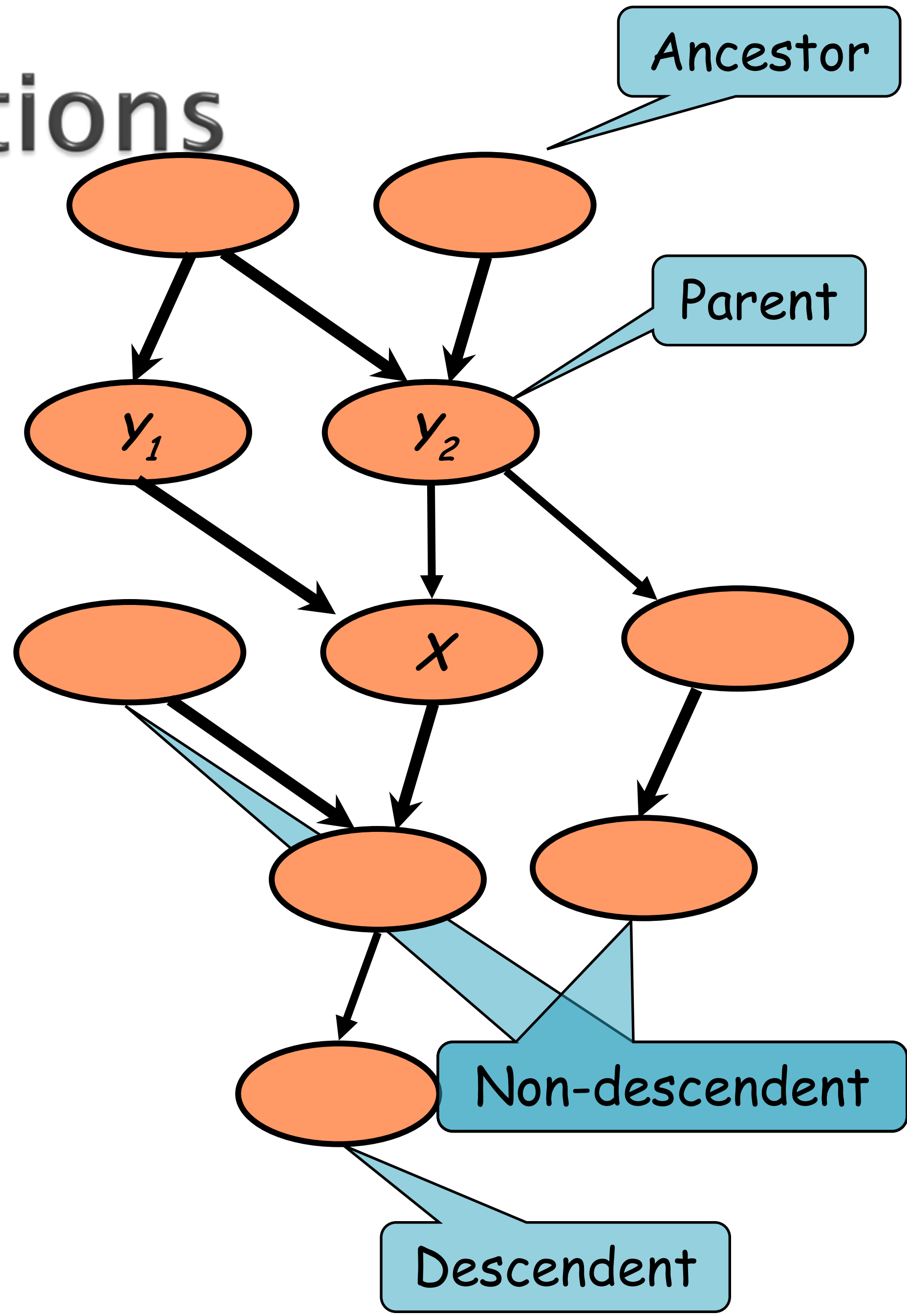
Structure: ***DAG***

- Local conditional distributions (**CPD**) and the **DAG** completely determine the **joint** dist.
- Give **causality** relationships, and facilitate a **generative** process – ancestral sampling



Markov Assumptions

- Each random variable X, is independent of its non-descendents, given its parents Pa(X)
- Formally,
  I (X, NonDesc(X) | Pa(X))
- Markov(G) = a (partial) set of independence statements implied by G



I–Maps

- A DAG G is an **I–Map** of a distribution P if the all Markov assumptions implied by G are satisfied by P

  (Assuming G and P both use the same set of random variables)

Examples:

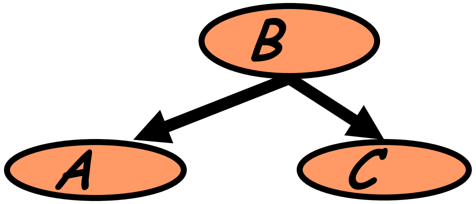
x	y	P(x,y)
0	0	0.25
0	1	0.25
1	0	0.25
1	1	0.25

x	y	P(x,y)
0	0	0.2
0	1	0.3
1	0	0.4
1	1	0.1

Local Structures & Independencies

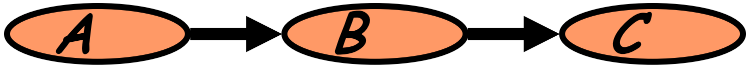
- Common parent
  - Fixing B decouples A and C

    "given the level of gene B, the levels of A and C are independent"



- Cascade
  - Knowing B decouples A and C

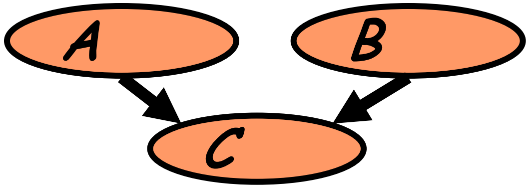
    "given the level of gene B, the level gene A provides no extra prediction value for the level of gene C"



- V-structure
  - Knowing C couples A and B

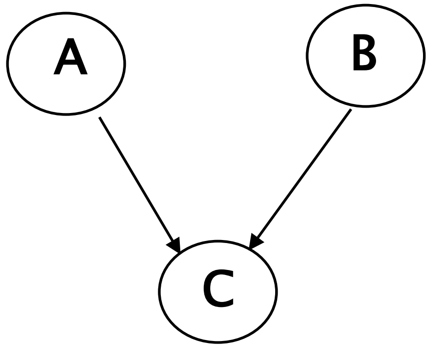
    because A can "explain away" B w.r.t. C

    "If A correlates to C, then chance for B to also correlate to C will decrease"



- The language is compact, the concepts are rich!

Example



p(A,B,C) =

Implied Independencies

- Does a graph *G* imply additional independencies as a consequence of *Markov(G)*?

- We can define a **logic** of independence statements

- Some axioms:
  - $I(X ; Y \mid Z) \Rightarrow I(Y; X \mid Z)$
  - $I(X ; Y_1, Y_2 \mid Z) \Rightarrow I(X; Y_1 \mid Z)$

d-seperation

- A procedure *d-sep(X; Y | Z, G)* that given a DAG *G*, and sets *X, Y,* and *Z* returns either *yes* or *no*

- **Goal**:

  d-sep(X; Y | Z, G) = yes iff *I(X;Y|Z)* follows from *Markov(G)*

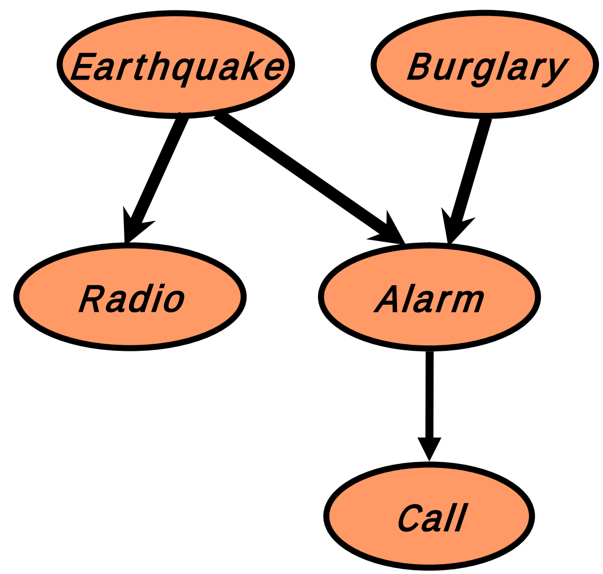
Paths

- **Intuition:** dependency must "flow" along paths in the graph

- A path is a sequence of neighboring variables

Examples:

- $R \leftarrow E \rightarrow A \leftarrow B$
- $C \leftarrow A \leftarrow E \rightarrow R$

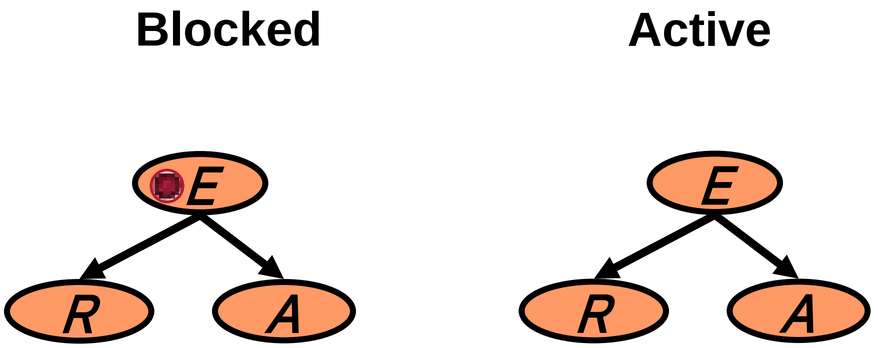


Paths

- We want to know when a path is
  - **active** -- creates dependency between end nodes
  - **blocked** -- cannot create dependency end nodes
- We want to classify situations in which paths are active.

Path Blockage

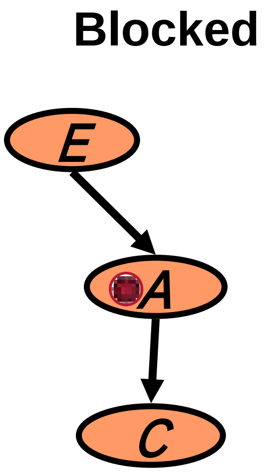
Three cases:
- Common cause
-
-

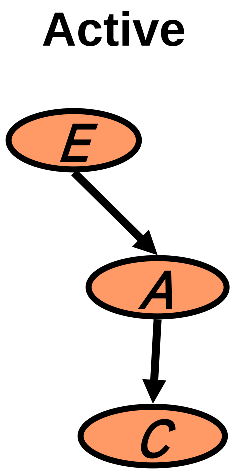


Path Blockage

Three cases:

- Common cause
- Intermediate cause
-



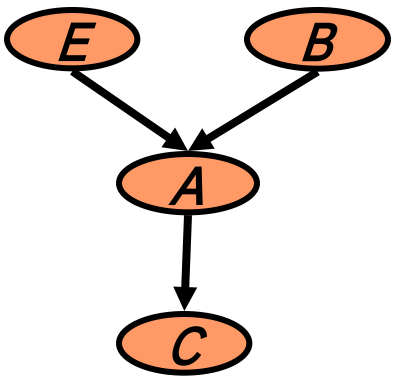


Path Blockage

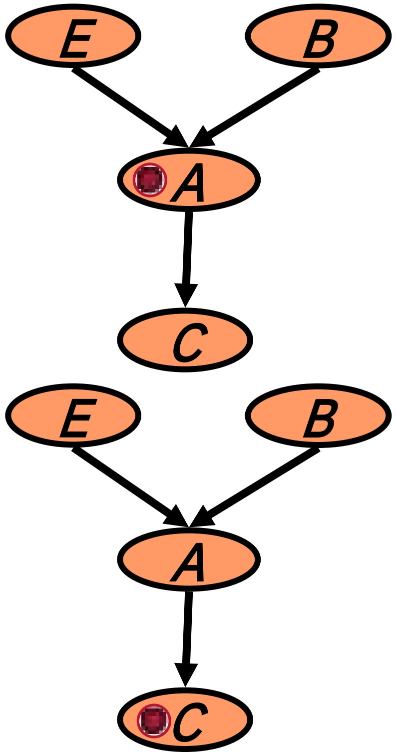
Three cases:

- Common cause
- Intermediate cause
- Common Effect

Blocked



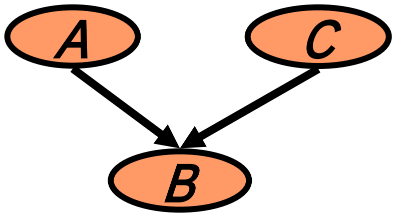
Active



Path Blockage -- General Case

A path is active, given evidence *Z*, if

- Whenever we have the configuration



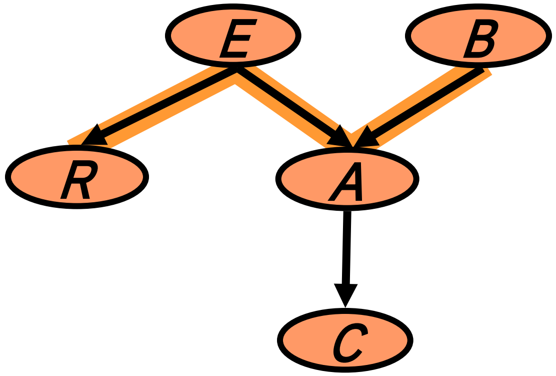
  B or one of its descendents are in *Z*

- No other nodes in the path are in *Z*

A path is blocked, given evidence *Z*, if it is not active.

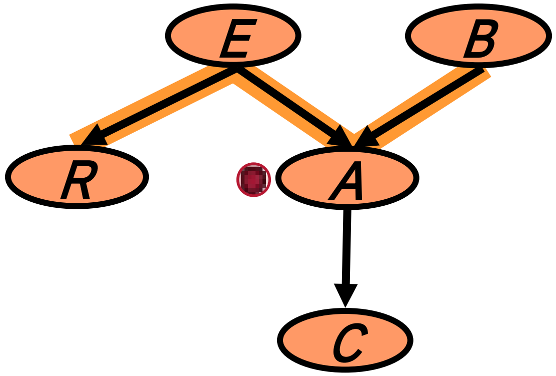
Example

- *Blocked R,B)?*



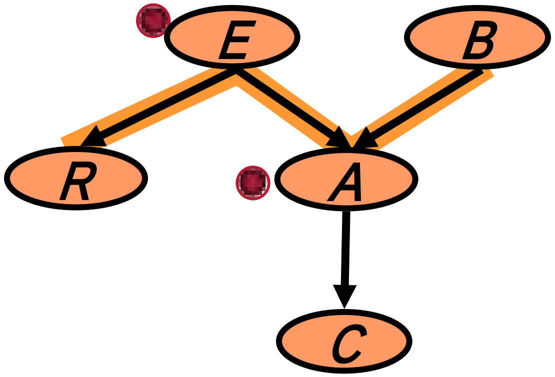
Example

- *Blocked (R,B) = yes*
- *Blcoked (R,B|A)?*



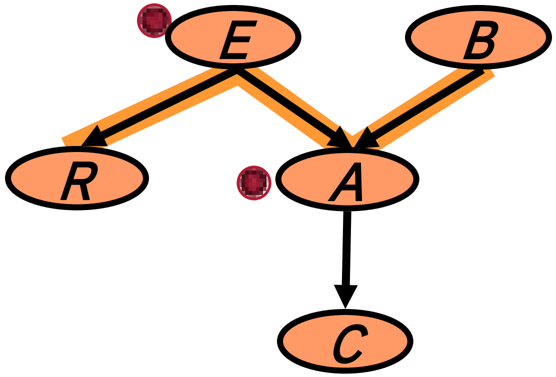
Example

- *Blocked (R,B) = yes*
- *Blocked (R,B|A) = no*
- *Blocked (R,B|E,A)?*



Example

- *Blocked (R,B) = yes*
- *Blocked (R,B|A) = no*
- *Blocked (R,B|E,A)? = yes*



d-Separation

- *X* is **d-separated** from *Y*, given *Z*, if all paths from a node in *X* to a node in *Y* are blocked, given *Z*.

- Checking d-separation can be done efficiently (linear time in number of edges)
  - Bottom-up phase: Mark all nodes whose descendents are in *Z*
  - *X* to *Y* phase: Traverse (BFS) all edges on paths from *X* to *Y* and check if they are blocked

Soundness

Thm:

- If
  - G is an I-Map of P
  - $d\text{-}sep(X; Y \mid Z, G) = yes$
- then
  - P satisfies $I(X; Y \mid Z)$

Informally,

- Any independence reported by d-separation is satisfied by underlying distribution

Completeness

Thm:

- If $d\text{-}sep(X; Y \mid Z, G) = no$
- then there is a distribution P such that
  - G is an I-Map of P
  - P does not satisfy $I(X; Y \mid Z)$

Informally,

- Any independence not reported by d-separation might be violated by the underlying distribution
- We cannot determine this by examining the graph structure alone

DAG Summary

- We explored DAGs as a representation of conditional independencies:
  - Markov independencies of a DAG
  - Tight correspondence between *Markov(G)* and the factorization defined by *G*
  - d-separation, a sound & complete procedure for computing the consequences of the independencies
- This theory is the basis for defining Bayesian networks

CPDs

- So far, we focused on how to represent independencies using DAGs
- The "other" component of a Bayesian networks is the specification of the **conditional probability distributions** (CPDs)

- We start with the simplest representation of CPDs for discrete RVs, and then discuss additional structure, then discuss continuous RVs and mixed discrete & continuous

Tabular CPDs

- When the variables of interest are all discrete, the common representation is as a table:
- For example $P(C|A,B)$ can be represented by

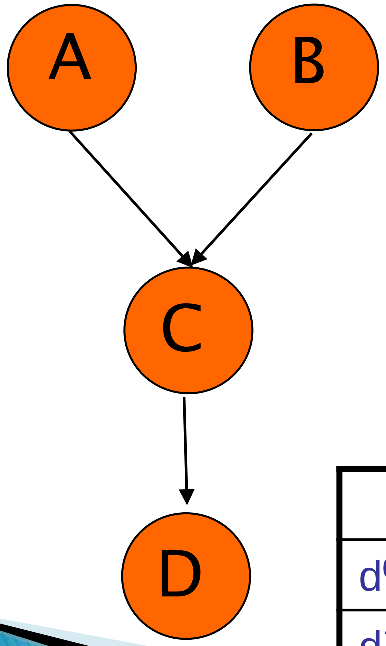
A	B	$P(C = 0 \mid A, B)$	$P(C = 1 \mid A, B)$
0	0	0.25	0.75
0	1	0.50	0.50
1	0	0.12	0.88
1	1	0.33	0.67

Conditional probability tables (CPTs)

a^0	0.75
a^1	0.25

b^0	0.33
b^1	0.67

$$P(a,b,c.d) = P(a)P(b)P(c|a,b)P(d|c)$$



	a^0b^0	a^0b^1	a^1b^0	a^1b^1
c^0	0.45	1	0.9	0.7
c^1	0.55	0	0.1	0.3

	c^0	c^1
d^0	0.3	0.5
d^1	07	0.5

Tabular CPDs

Pros:

- Very flexible, can capture any CPD of discrete variables
- Can be easily stored and manipulated

Cons:

- Representation size grows exponentially with the number of parents!
- Unwieldy to assess probabilities for more than few parents

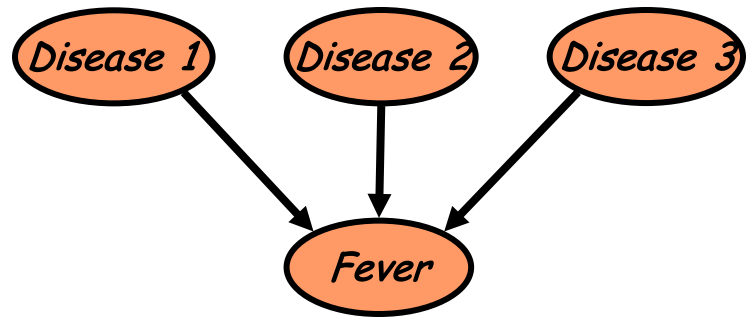
Structured CPD

- To avoid the exponential blowup in representation, we need to focus on specialized types of CPDs
- This comes at a cost in terms of expressive power

- There are several types of structured CPDs
  - Noisy-Or
  - Decision Tree CPDs

Causal Independence

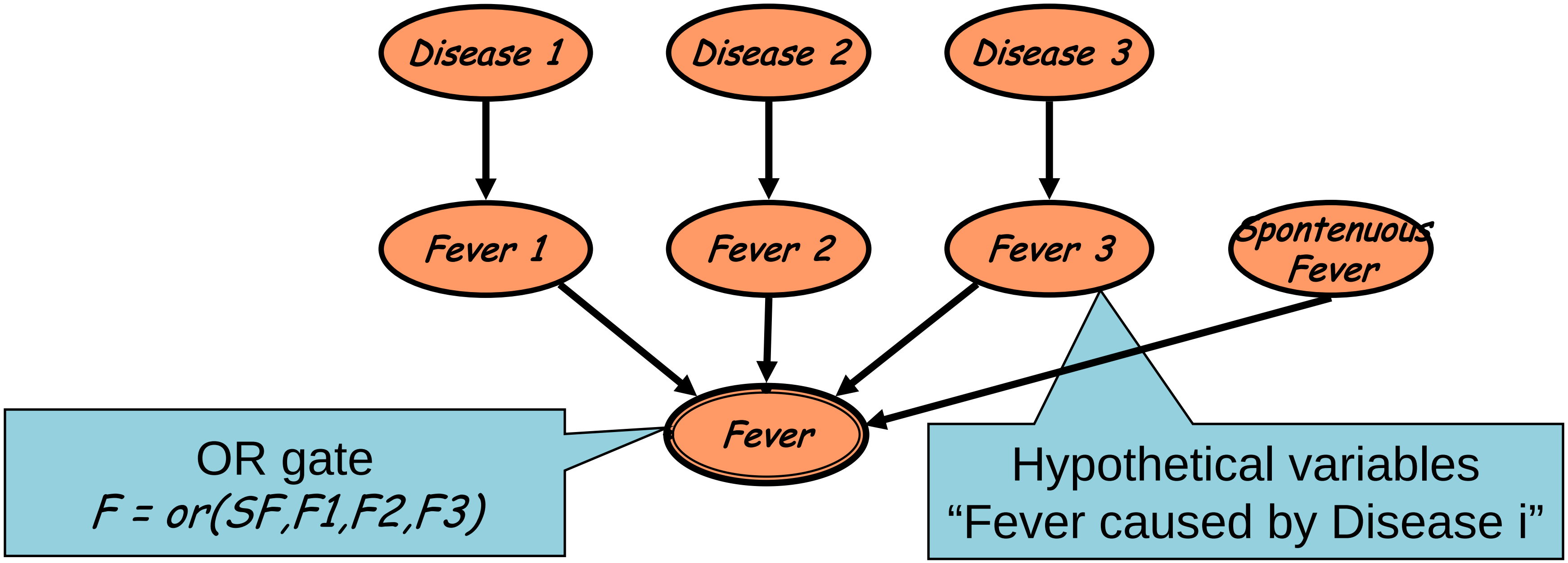
- Consider the following situation



- In tabular CPD, we need to assess the probability of fever in eight cases
- These involve all possible interactions between diseases
- For three disease, this might be feasible….
  For ten diseases, not likely….

Causal Independence

- Simplifying assumption:
  - Each disease attempts to cause fever, **independently** of the other diseases
  - The patient has fever if one of the diseases "succeeds"
- We can model this using a Bayesian network fragment



Noisy-Or CPD

- Models $P(X|Y_1,\ldots,Y_k)$, X, $Y_1,\ldots, Y_k$ are all binary

- Parameters:
  - p_i -- probability of $X = 1$ due to $Y_i = 1$
  - p_0-- probability of $X = 1$ due to other causes

- Plugging these in the model we get

$$P(X=0|Y_1,\ldots,Y_k) = (1-p_0)\prod_i (1-p_i)^{Y_i}$$

$$P(X=1|Y_1,\ldots,Y_k) = 1 - P(X=0|Y_1,\ldots,Y_k)$$

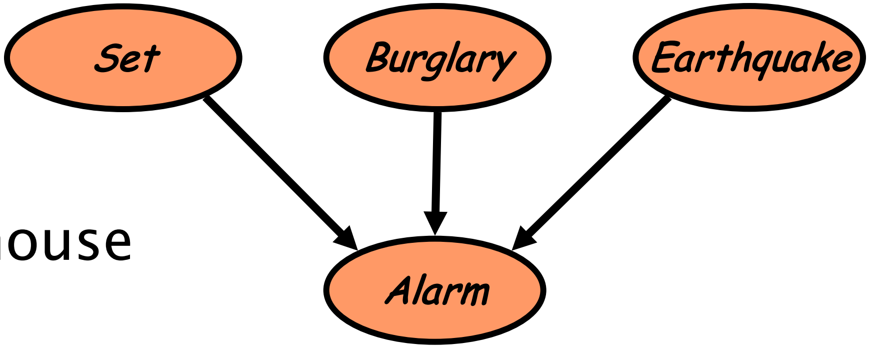
Noisy-Or CPD

- Benefits of noisy-or
  - "Reasonable" assumptions in many domains
    - e.g., medical domain
  - Few parameters.
  - Each parameter can be estimated independently of the others

- The same idea can be extended to other functions:
  noisy-max, noisy-and, etc.
- Frequently used in large medical expert systems

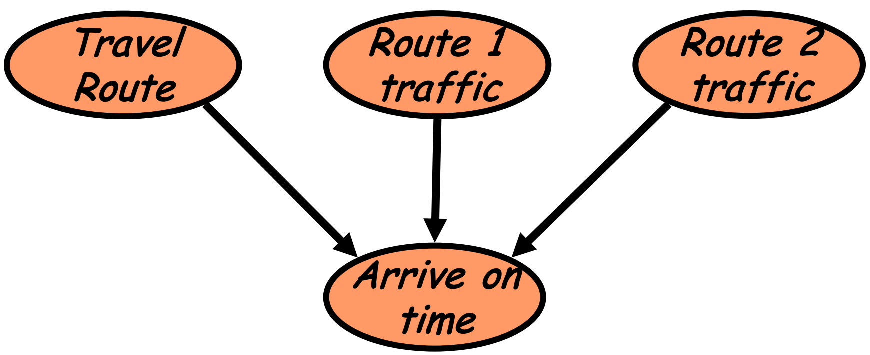
Context Specific Independence

- Consider the following examples:



- Alarm sound depends on
  - Whether the alarm was set before leaving the house
  - Burglary
  - Earthquake

- Arriving on time depends on
  - Travel route
  - The congestion on the two possible routes



Context-Specific Independence

- In both of these example we have **context-specific independencies (CSI)**
  - Independencies that depends on a particular value of one or more variables

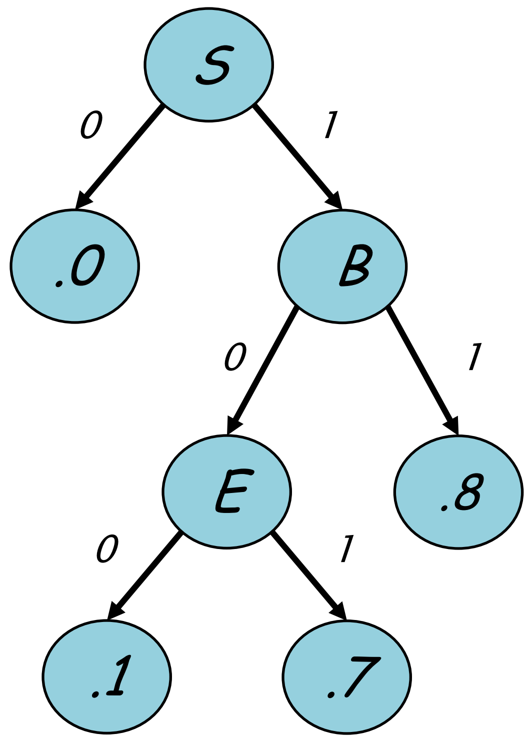
- In our examples:
  - $I(A ; B, E \mid S = 0)$
    Alarm sound is independent of B and E when the alarm is not set
  - $I(A ; R_2 \mid T = 1)$
    Arrival time is independent of traffic on route 2 if we choose to travel on route 1

Representing CSI

- When we have such CSI, $P(X \mid Y_1,\ldots,Y_k)$ is the same for several values of $Y_1,\ldots,Y_k$
- There are many ways of representing these regularities

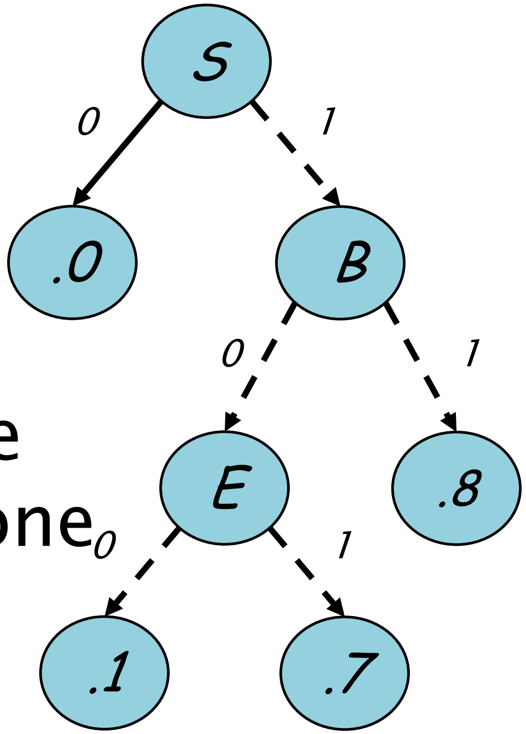
- A natural representation: decision trees
  - Internal nodes: tests on parents
  - Leaves: probability distributions on X

- Evaluate $P(X \mid Y_1,\ldots,Y_k)$ by traversing tree



Detecting CSI

- Given evidence on some nodes, we can identify the "relevant" parts of the trees
  - This consists of the paths in the tree that are consistent with context
- Example
  - Context $S = 0$
  - Only one path of tree is relevant
- A parent is independent given the context if it does not appear on one of the relevant paths



Decision Tree CPDs

Benefits

- Decision trees offer a flexible and intuitive language to represent CSI
- Incorporated into several commercial tools for constructing Bayesian networks

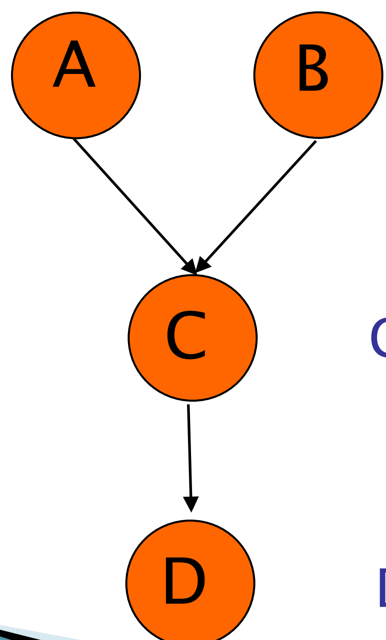
Comparison to noisy-or

- Noisy-or CPDs require full trees to represent
- General decision tree CPDs cannot be represented by noisy-or

Cont. RVs: Conditional probability density func. (CPDs)

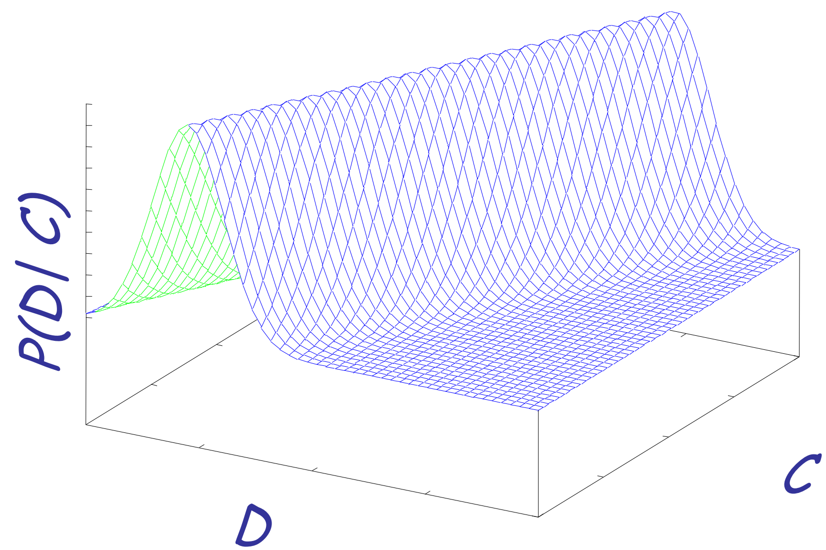
$A \sim N(\mu_a, \Sigma_a)$ $B \sim N(\mu_b, \Sigma_b)$

$$\boxed{\begin{array}{c} P(a,b,c.d) = \\ P(a)P(b)P(c|a,b)P(d|c) \end{array}}$$



$C \sim N(A+B, \Sigma_c)$

$D \sim N(\mu_a + C, \Sigma_a)$



Conditional Gaussian CPDs

- A model for networks that combine discrete and continuous variables
- If X is continuous
  - $y_1,\dots,y_k$ are continuous
  - $Z_1,\dots,Z_l$ are discrete

Conditional Gaussian (CG) CPD:

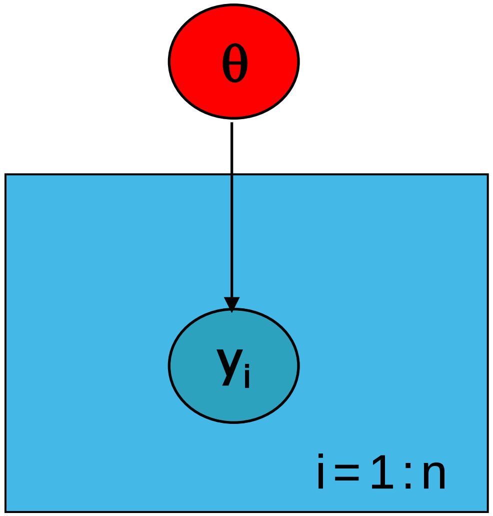
- For each joint value of $Z_1,\dots,Z_l$ define a different Gaussian parameters
- Resulting multivariate distribution: mixture of multivariate Gaussians
  - Each assignment of values to discrete variables selects a multivariate Gaussian over continuous variables

CPD Summary

- Many choices for representing CPDs
- Any "statistical" model of conditional distribution can be used
  - e.g., any discrete model, any regression model

- Representing structure in CPDs can have implications on independencies among variables

Aside: "Plate" Notation



Model parameters

Data = $\{y_1,\ldots y_n\}$

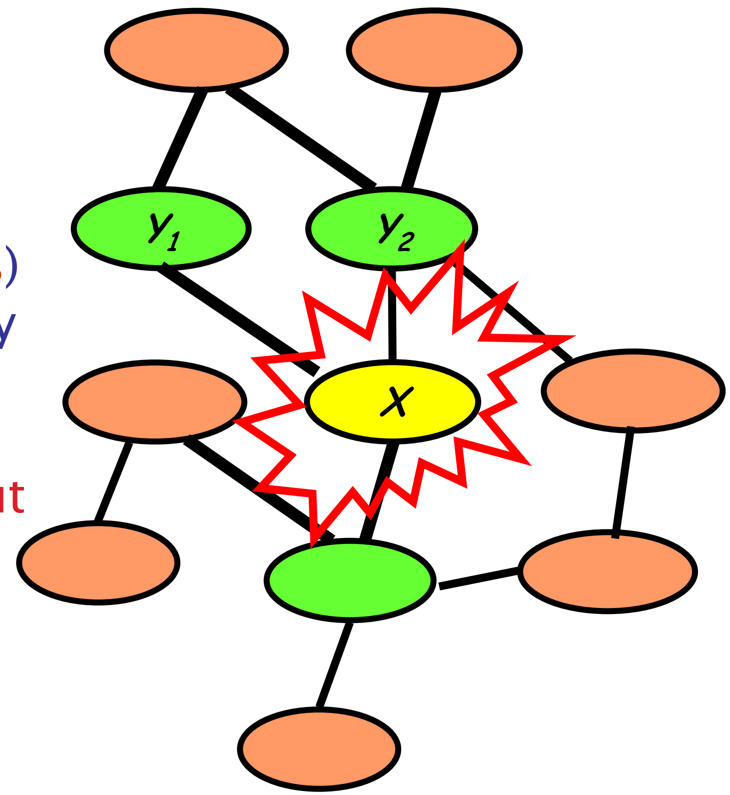
Plate = rectangle in graphical model

variables within a plate are replicated
in a conditionally independent manner

Markov Random Fields

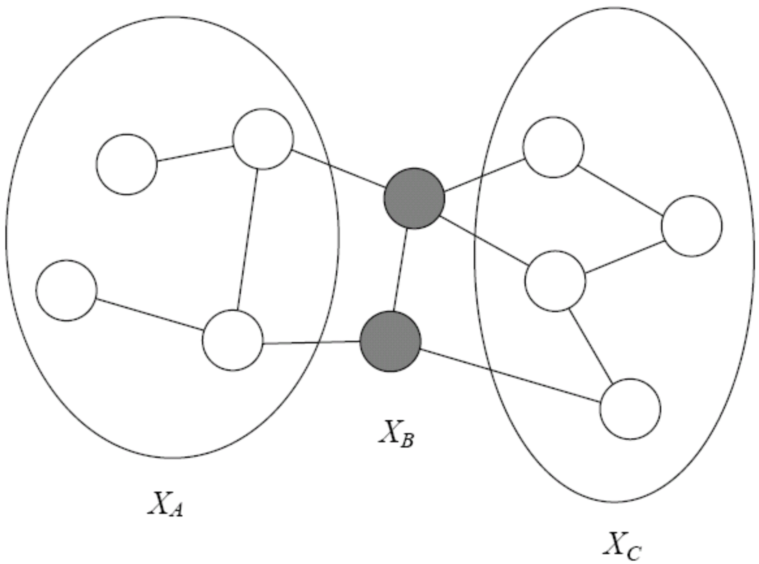
Structure: an *undirected graph*

- Meaning: a node is **conditionally independent** of every other node in the network given its **neighbors**
- Local contingency functions (**potentials**) and the **cliques** in the graph completely determine the **joint** dist.
- Give **correlations** between variables, but no explicit way to generate samples



Semantics of Undirected Graphs

- Let H be an undirected graph:



- B ***separates*** A and C if every path from a node in A to a node in C passes through a node in B: $\mathrm{sep}_H(A;C|B)$
- A probability distribution satisfies the ***global Markov property*** if for any disjoint A, B, C, such that B separates A and C, A is independent of C given B:

$$\mathrm{I}(H) = \{I(A,C|B) : \mathrm{sep}_H(A;C|B)\}$$

Representation

- Defn: an undirected graphical model represents a distribution $P(X_1, \ldots, X_n)$ defined by an undirected graph H, and a set of positive ***potential functions*** ψ_c associated with cliques of H, s.t.

$$P(x_1, \ldots, x_n) = \frac{1}{Z} \prod_{c \in C} \psi_c(\mathbf{x}_c)$$

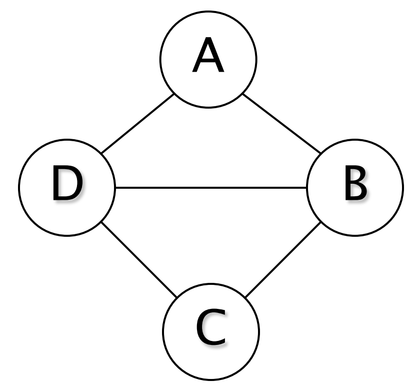
  where Z is known as the partition function:

$$Z = \sum_{x_1, \ldots, x_n} \prod_{c \in C} \psi_c(\mathbf{x}_c)$$

- Also known as Markov Random Fields, Markov networks …
- The ***potential function*** can be understood as an contingency function of its arguments assigning "pre-probabilistic" score of their joint configuration.

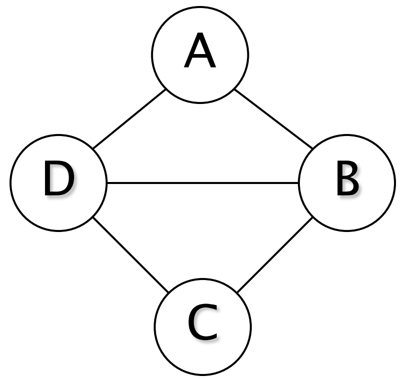
Cliques

- For $G=\{V,E\}$, a complete subgraph (clique) is a subgraph $G'=\{V'\subseteq V, E'\subseteq E\}$ such that nodes in V' are fully interconnected
- A (maximal) clique is a complete subgraph s.t. any superset is not complete.
- A sub-clique is a not-necessarily-maximal clique.

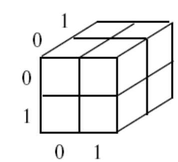


- Example:
  - max-cliques = $\{A,B,D\}$, $\{B,C,D\}$,
  - sub-cliques = $\{A,B\}$, $\{C,D\}$, …→ all edges and singletons

Example UGM – using max cliques



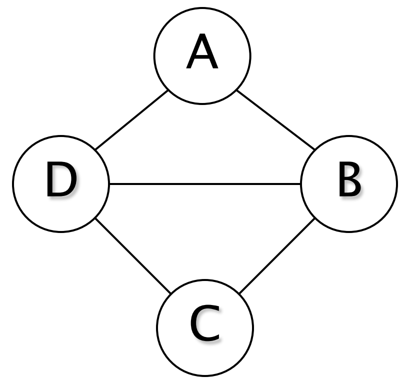
$$P(x_1, x_2, x_3, x_4) = \frac{1}{Z}\psi_c(\mathbf{x}_{124}) \times \psi_c(\mathbf{x}_{234})$$



$$Z = \sum_{x_1, x_2, x_3, x_4} \psi_c(\mathbf{x}_{124}) \times \psi_c(\mathbf{x}_{234})$$

- For discrete nodes, we can represent $P(X_1, X_2, X_3, X_4)$ as two 3D tables instead of one 4D table

Example UGM – using subcliques

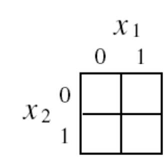


$$P(x_1, x_2, x_3, x_4) = \frac{1}{Z} \prod_{ij} \psi_{ij}(\mathbf{x}_{ij})$$

$$= \frac{1}{Z} \psi_{12}(\mathbf{x}_{12}) \psi_{14}(\mathbf{x}_{14}) \psi_{23}(\mathbf{x}_{23}) \psi_{24}(\mathbf{x}_{24}) \psi_{34}(\mathbf{x}_{34})$$

$$Z = \sum_{x_1, x_2, x_3, x_4} \prod_{ij} \psi_{ij}(\mathbf{x}_{ij})$$

- For discrete nodes, we can represent $P(X_1, X_2, X_3, X_4)$ as 5 2D tables instead of one 4D table



Exponential Form

- Constraining clique potentials to be positive could be inconvenient (e.g., the interactions between a pair of atoms can be either attractive or repulsive). We represent a clique potential $\psi_c(\mathbf{x}_c)$ in an unconstrained form using a real-value "energy" function $\phi_c(\mathbf{x}_c)$:

$$\psi_c(\mathbf{x}_c) = \exp\{-\phi_c(\mathbf{x}_c)\}$$

For convenience, we will call $\phi_c(\mathbf{x}_c)$ a potential when no confusion arises from the context.

- This gives the joint a nice additive strcuture

$$p(\mathbf{x}) = \frac{1}{Z}\exp\left\{-\sum_{c\in C}\phi_c(\mathbf{x}_c)\right\} = \frac{1}{Z}\exp\{-H(\mathbf{x})\}$$

where the sum in the exponent is called the "free energy":

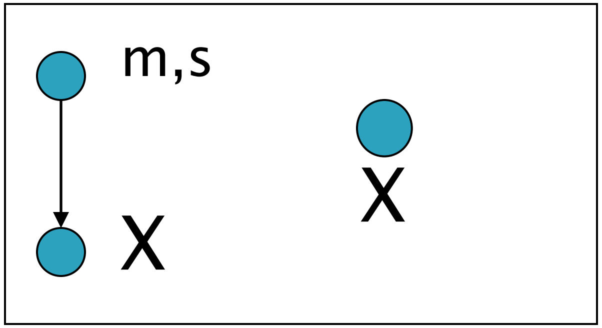
- In physics, this is called the "Boltzmann distribution".
- In statistics, this is called a log-linear model.

$$H(\mathbf{x}) = \sum_{c\in C}\phi_c(\mathbf{x}_c)$$

GMs are your old friends

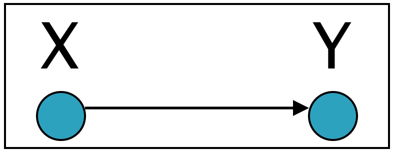
Density estimation

Parametric and nonparametric methods



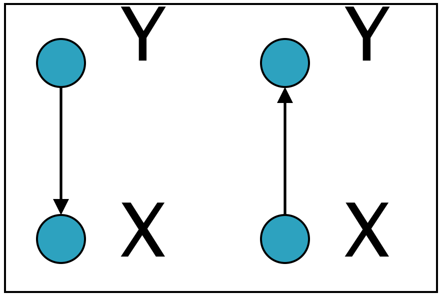
Regression

Linear, conditional mixture, nonparametric



Classification

Generative and discriminative approach



Why graphical models

- **Probability theory** provides the **glue** whereby the parts are combined, ensuring that the system as a whole is consistent, and providing ways to interface models to data.
- The **graph theoretic** side of graphical models provides both an intuitively appealing interface by which humans can model highly-interacting sets of variables as well as a data structure that lends itself naturally to the design of efficient general-purpose algorithms.
- **Many of the classical multivariate probabilistic systems** studied in fields such as statistics, systems engineering, information theory, pattern recognition and statistical mechanics **are special cases of the general graphical model formalism**

  -- examples include mixture models, factor analysis, hidden Markov models, Kalman filters and Ising models.
- The graphical model framework provides a way to view all of these systems as instances of a **common underlying formalism**.

--- M. Jordan

Next Time....

- Inference
- Reading: Bishop ch. 8