Probabilistic graphical models
CS 535c (Topics in AI)
Stat 521a (Topics in multivariate analysis)

Lecture 1

Kevin Murphy

Monday 13 September, 2004

Administrivia

- Lectures: MW 9.30-10.50, CISR 304
- Regular homeworks: 40% of grade
  - Simple theory exercises.
  - Simple Matlab exercises.
- Final project: 60% of grade
  - Apply PGMs to your research area (e.g., vision, language, bioinformatics)
  - Add new features to my software package for PGMs
  - Theoretical work
- No exams

Admninistrivia

- Please send email to majordomo@cs.ubc.ca with the contents `subscribe cpsc535c` to get on the class mailing list.
- URL `www.cs.ubc.ca/~murphyk/Teaching/CS532c_Fall04/index.html`
- **Class on Wed 15th starts at 10am!**
- No textbook, but some draft chapters may be handed out in class.
  - *Introduction to Probabilistic Graphical Models* , Michael Jordan
  - *Bayesian networks and Beyond*, Daphne Koller and Nir Friedman

Probabilistic graphical models

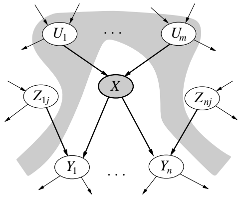
- Combination of graph theory and probability theory.
- Informally,
  - Graph structure specifies which parts of system are directly dependent.
  - Local functions at each node specify how parts interact.
- More formally,
  - Graph encodes conditional independence assumptions.
  - Local functions at each node are factors in the joint probability distribution.
- Bayesian networks = PGMs based on directed acyclic graphs.
- Markov networks (Markov random fields) = PGM with undirected graph.

Applications of PGMs

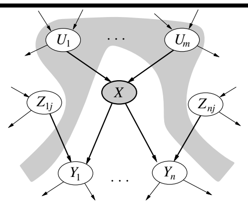
- Machine learning
- Statistics
- Speech recognition
- Natural language processing
- Computer vision
- Error-control codes
- Bio-informatics
- Medical diagnosis
- etc.

Bayesian networks (aka belief network, directed graphical model)

- Nodes are random variables.
- Informally, edges represent "causation" (no directed cycles allowed - graph is a DAG).
- Formally, local Markov property says: node is conditionally independent of its non-descendants given its parents.

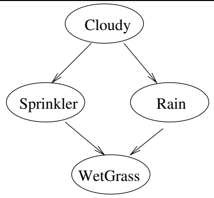


Chain rule for Bayesian networks



$$
\begin{aligned}
P(X_{1:N}) &= P(X_1)P(X_2|X_1)P(X_3|X_1, X_2)\ldots \\
&= \prod_{i=1}^{N} P(X_i|X_{1:i-1}) \\
&= \prod_{i=1}^{N} P(X_i|X_{\pi_i})
\end{aligned}
$$

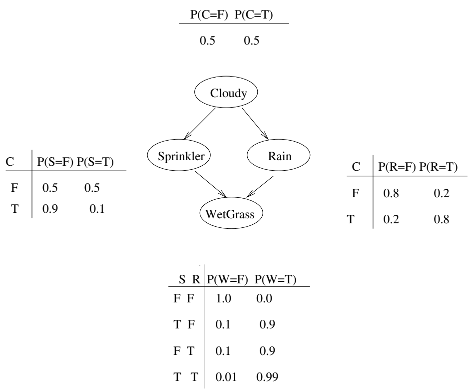
Water sprinkler Bayes net



$$
\begin{aligned}
P(C,S,R,W) &= P(C)P(S|C)P(R|S,C)P(W|S,R,C) && \text{chain rule} \\
&= P(C)P(S|C)P(R|\cancel{S},C)P(W|S,R,C) && \text{since } S \perp R|C \\
&= P(C)P(S|C)P(R|\cancel{S},C)P(W|S,R,\cancel{C}) && \text{since } W \perp C|S, R \\
&= P(C)P(S|C)P(R|C)P(W|S,R)
\end{aligned}
$$

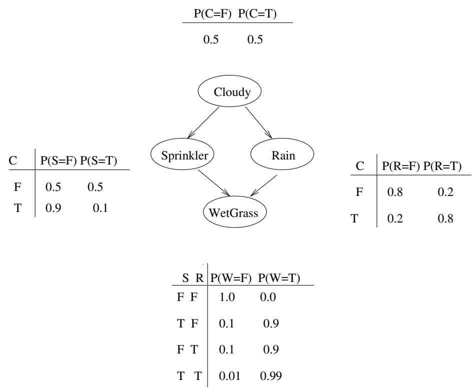
Conditional Probability Distributions (CPDs)

- Associated with every node is a probability distribution over its values given its parents values.
- If the variables are discrete, these distributions can be represented as tables (CPTs).

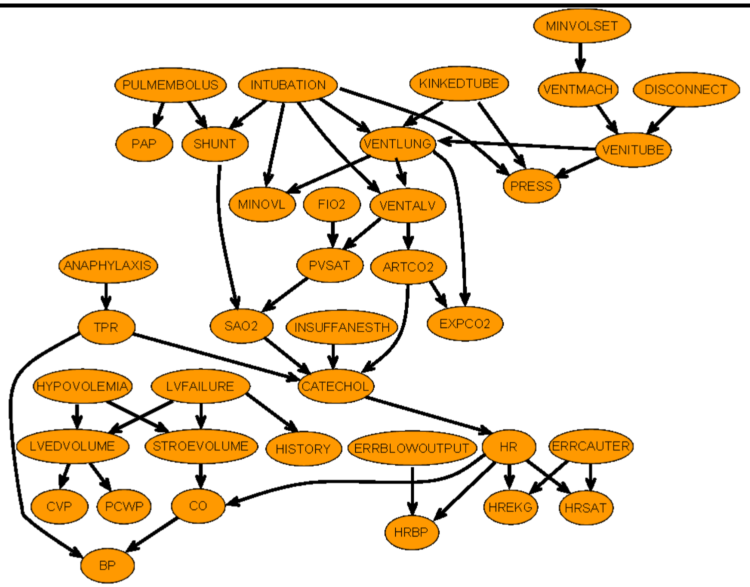


Bayes nets provide compact representation of joint probability distributions

- For N binary nodes, need $2^N - 1$ parameters to specify $P(X_1, \ldots, X_N)$.
- For BN, need $O(N2^K)$ parameters, where K = max. number of parents (fan-in) per node.
- e.g., $2^4 - 1 = 31$ vs $2 + 4 + 4 + 8 = 18$ parameters.



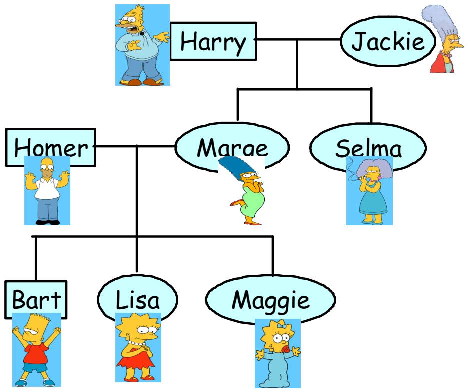
Alarm network



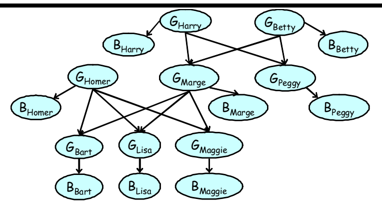
Intensive Care Unit monitoring

Bayes net for genetic pedigree analysis

- $G_i \in \{a, b, o\} \times \{a, b, o\}$ = genotype (allele) of person i
- $B_i \in \{a, b, o, ab\}$ = phenotype (blood type) of person i



Bayes net for genetic pedigree analysis - CPDs

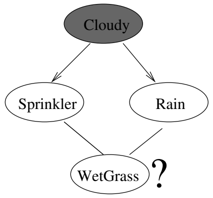


- Mendels laws define $P(G|G_p, G_m)$
- Phenotypic expression specifies $P(B|G)$:

G	$P(B=a)$	$P(B=b)$	$P(B=o)$	$P(B=ab)$
a a	1	0	0	0
a b	0	0	0	1
a o	1	0	0	0
b a	0	0	0	1
b b	0	1	0	0
b o	0	1	0	1
o a	1	0	0	0
o b	0	1	0	0
o o	0	0	1	0

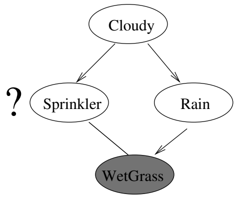
Inference (state estimation)

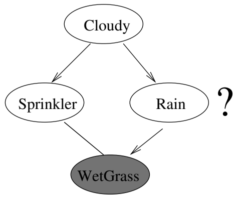
- Inference = estimating hidden quantities from observed.
- Causal reasoning/ prediction (from causes to effects): how likely is it that clouds cause the grass to be wet? $P(w = 1|c = 1)$



Inference (state estimation)

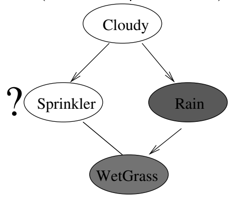
- Inference = estimating hidden quantities from observed.
- Diagnostic reasoning (from effects to causes): the grass is wet; was it caused by the sprinkler or rain?
  $P(S = 1|w = 1)$ vs $P(R = 1|w = 1)$
- Most Probable Explanation:
  $\arg\max_{s,r} P(S = s, R = r|w = 1)$



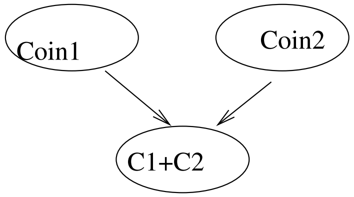


Explaining away

- Explaining away (inter-causal reasoning)
- $P(S=1|w=1,r=1) < P(S=1|w=1)$



- Coins 1, 2 marginally independent, become dependent when observe their sum.



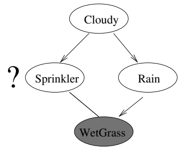
Naive inference

- We can compute any query we want by marginalizing the joint,e.g,

$$P(s=1|w=1) = \frac{P(s=1,w=1)}{P(w=1)}$$

$$= \frac{\sum_{c,r} P(s=1,w=1,R=r,C=c)}{\sum_{c,r,s} P(S=s,w=1,R=r,C=c)}$$

$$= \frac{\sum_{c,r} P(C=c)P(S=1|C=c)P(R=r|C=c)P(W=1|S=s,R=r}{\sum_{c,r,s} P(S=s,w=1,R=r,C=c)}$$



- Takes $O(2^N)$ time
- Homework 1, question 3

Simple inference: classification

- Example: medical diagnosis
- Given list of observed findings (evidence), such as
  - e_1: sex = male
  - e_2: abdomen pain = high
  - e_3: shortness of breath = false
- Infer most likely cause:

$$c^* = \arg\max_c P(c|e_{1:N})$$

Approach 1: learn discriminative classifier

- We can try to fit a function to approximate $P(c|e_{1:N})$ using labeled training data (a set of $(c, e_{1:N})$ pairs).
- This is the standard approach in supervised machine learning.
- Possible functional forms:
  - Support vector machine (SVM)
  - Neural network
  - Decision tree
  - Boosted decision tree
- See classes by Nando de Freitas:
  - CPSC 340, Fall 2004 - undergrad machine learning
  - CPSC 540, Spring 2005 - grad machine learning

Approach 2: build generative model and use Bayes' rule to invert

- We can build a causal model of how diseases cause symptoms, and use Bayes' rule to invert:

$$P(c|e_{1:N}) = \frac{P(e_{1:N}|c)P(c)}{P(e)} = \frac{P(e_{1:N}|c)P(c)}{\sum_{c'} P(e_{1:N}|c')P(c')}$$

- In words

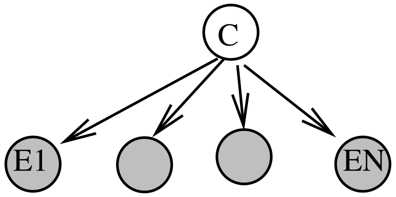
$$\text{posterior} = \frac{\text{class-conditional likelihood} \times \text{prior}}{\text{marginal likelihood}}$$

Naive Bayes classifier

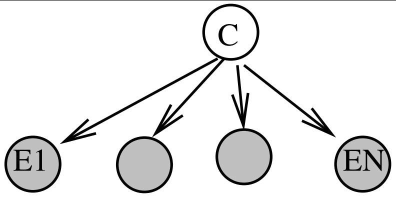
- Simplest generative model: assume effects are conditionally independent given the cause: $E_i \perp E_j | C$

$$P(E_{1:N}|C) = \prod_{i=1}^{N} P(E_i|C)$$

- Hence $P(c|e_{1:N}) \propto P(e_{1:N}|c)P(c) = \prod_{i=1}^{N} P(e_i|c)P(c)$



Naive Bayes classifier

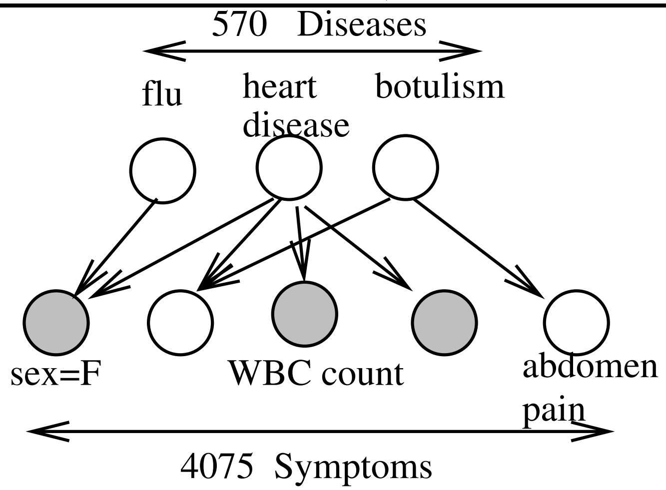


- This model is extremely widely used (e.g., for document classification, spam filtering, etc) even when observations are not independent.

$$P(c|e_{1:N}) \propto P(e_{1:N}|c)P(c) = \prod_{i=1}^{N} P(e_i|c)P(c)$$

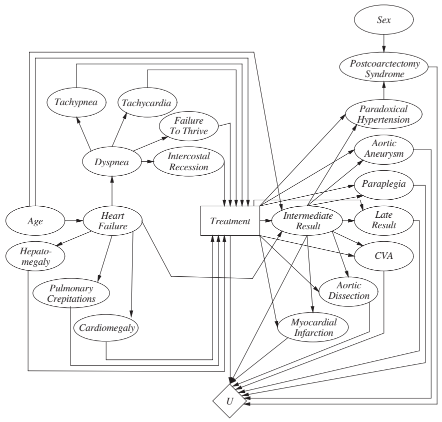
$P(C = \text{cancer}|E_1 = \text{spots}, E_2 = \text{vomiting}, E_3 = \text{fever}) \propto$
P(spots |cancer) P(vomiting|cancer) P(fever|cancer) P(C=cancer)

QMR-DT Bayes net
(Quick medical reference, decision theoretic)



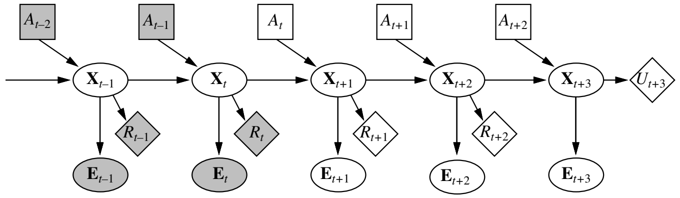
Decision theory

- Decision theory = probability theory + utility theory.
- Decision (influence) diagrams = Bayes nets + action (decision) nodes + utility (value) nodes.
- See David Poole's class, CS 522

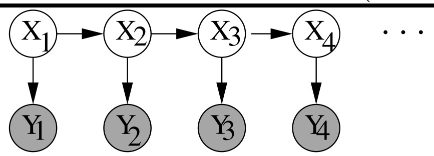


POMDPs

- POMDP = Partially observed Markov decision process
- Special case of influence diagram (infinite horizon)



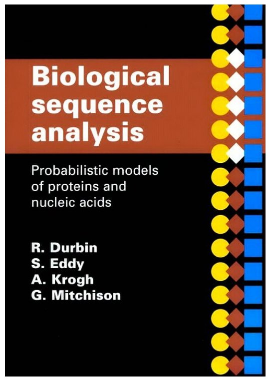
Hidden Markov Model (HMM)



- HMM = POMDP - action - utility
- Inference goal:
  - Online state estimation: $P(X_t|y_{1:t})$
  - Viterbi decoding (most probable explanation): $\arg\max_{x_{1:t}} P(x_{1:t}|y_{1:t})$

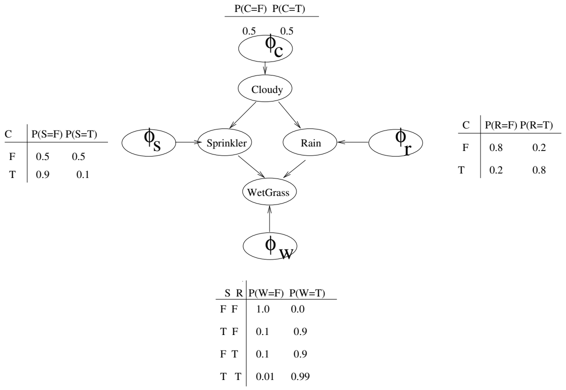
Domain	Hidden state X	Observation Y
Speech	Words	Spectogram
Part-of-speech tagging	Noun/ verb/ etc	Words
Gene finding	Intron/ exon/ non-coding	DNA
Sequence alignment	Insert/ delete/ match	Amino acids

Biosequence analysis using HMMs



Learning

- Structure learning (model selection): where does the graph come from?
- Parameter learning (parameter estimation): where do the numbers come from?



Parameter learning

- Assume we have iid training cases where each node is fully observed: $D = \{c^i, s^i, r^i, w^i\}$.
- Bayesian approach
  - Treat parameters as random variables.
  - Compute posterior distribution: $P(\phi|D)$ (inference).
- Frequentist approach
  - Treat parameters as unknown constants.
  - Find best estimate, e.g., penalized maximum likelihood (optimization):

$$\phi^* = \arg\max_{\phi} \log P(D|\phi) - \lambda C(\phi)$$

Structure learning

- Assume we have iid training cases where each node is fully observed: $D = \{c^i, s^i, r^i, w^i\}$.
- Bayesian approach
  - Treat graph as random variable.
  - Compute posterior distribution: $P(G|D)$
- Frequentist approach
  - Treat graph as unknown constant.
  - Find best estimate, e.g., maxmimum penalized likelihood:

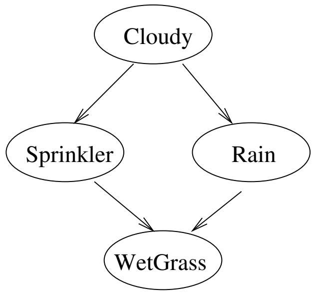
$$G^* = \arg\max_{G} \log P(D|G) - \lambda C(G)$$

Outline of class

- Representation
  - Undirected graphical models
  - Markov properties of graphs
- Inference
  - Models with discrete hidden nodes
    - Exact (e.g., forwards backwards for HMMs)
    - Approximate (e.g., loopy belief propagation)
  - Models with continuous hidden nodes
    - Exact (e.g., Kalman filtering)
    - Approximate (e.g., sampling)
- Learning
  - Parameters (e.g., EM)
  - Structure (e.g., structural EM, causality)

Review: Representation

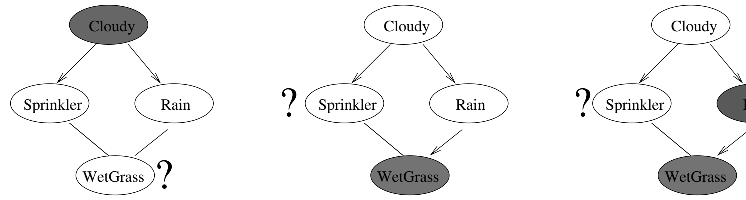
- Graphical models encode conditional independence assumptions.
- Bayesian networks are based on DAGs.



$$P(C, S, R, W) = P(C)P(S|C)P(R|C)P(W|S, R)$$

Review: Inference (state estimation)

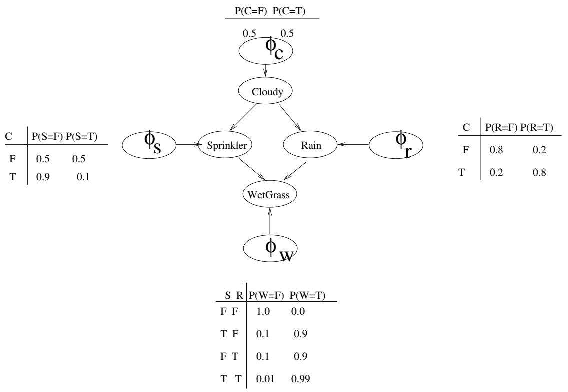
- Inference = estimating hidden quantities from observed.



- Naive method takes $O(2^N)$ time

Review: Learning

- Structure learning (model selection): where does the graph come from?
- Parameter learning (parameter estimation): where do the numbers come from?

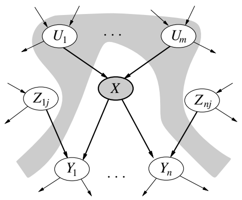


Outline

- Conditional independence properties of DAGs

Local Markov property

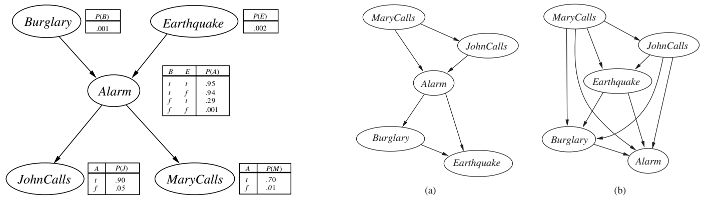
- Node is conditionally independent of its non-descendants given its parents.



$$
\begin{aligned}
P(X_{1:N}) &= P(X_1)P(X_2|X_1)P(X_3|X_1, X_2)\ldots \\
&= \prod_{i=1}^{N} P(X_i|X_{1:i-1}) \\
&= \prod_{i=1}^{N} P(X_i|X_{\pi_i})
\end{aligned}
$$

Topological ordering

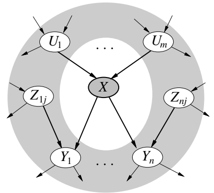
- If we get the ordering wrong, the graph will be more complicated, because the parents may not include the relevant variables to "screen off" the child from its irrelevant ancestors.



(a) (b)

Local Markov property version 2

- A Node is conditionally independent of all others given its Markov blanket.
- The markov blanket is the parents, children, and childrens' parents.



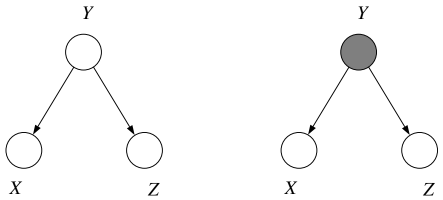
Global Markov properties of DAGs

- By chaining together local independencies, we can infer more global independencies.
- Defn: $X_1 - X_2 \cdots - X_n$ is an *active* path in a DAG G given evidence E if
  1. Whenever we have a v-structure, $X_{i-1} \rightarrow X_i \leftarrow X_{i+1}$, then X_i or one of its descendants is in E; and
  2. no other node along the path is in E
- Defn: X is *d-separated* (directed-separated) from Y given E if there is no active path from any $x \in X$ to any $y \in Y$ given E.
- Theorem: $\mathbf{x}_A \perp \mathbf{x}_B | \mathbf{x}_C$ if every variable in A is d-separated from every variable in B conditioned on all the variables in C.

Chain

- Q: When we condition on $\mathbf{y}$, are $\mathbf{x}$ and $\mathbf{z}$ independent?
$$\mathsf{P}(\mathbf{x},\mathbf{y},\mathbf{z}) = \mathsf{P}(\mathbf{x})\mathsf{P}(\mathbf{y}|\mathbf{x})\mathsf{P}(\mathbf{z}|\mathbf{y})$$
which implies
$$\begin{aligned}\mathsf{P}(\mathbf{x},\mathbf{z}|\mathbf{y}) &= \frac{\mathsf{P}(\mathbf{x})\mathsf{P}(\mathbf{y}|\mathbf{x})\mathsf{P}(\mathbf{z}|\mathbf{y})}{\mathsf{P}(\mathbf{y})}\\ &= \frac{\mathsf{P}(\mathbf{x},\mathbf{y})\mathsf{P}(\mathbf{z}|\mathbf{y})}{\mathsf{P}(\mathbf{y})}\\ &= \mathsf{P}(\mathbf{x}|\mathbf{y})\mathsf{P}(\mathbf{z}|\mathbf{y})\end{aligned}$$
and therefore $\mathbf{x} \perp \mathbf{z}|\mathbf{y}$
- Think of $\mathbf{x}$ as the past, $\mathbf{y}$ as the present and $\mathbf{z}$ as the future.

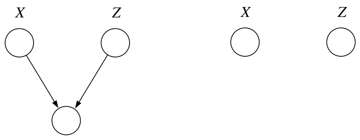
Common Cause



$\mathbf{y}$ is the common cause of the two independent effects $\mathbf{x}$ and $\mathbf{z}$

- Q: When we condition on $\mathbf{y}$, are $\mathbf{x}$ and $\mathbf{z}$ independent?
$$\mathsf{P}(\mathbf{x},\mathbf{y},\mathbf{z}) = \mathsf{P}(\mathbf{y})\mathsf{P}(\mathbf{x}|\mathbf{y})\mathsf{P}(\mathbf{z}|\mathbf{y})$$
which implies
$$\begin{aligned}\mathsf{P}(\mathbf{x},\mathbf{z}|\mathbf{y}) &= \frac{\mathsf{P}(\mathbf{x},\mathbf{y},\mathbf{z})}{\mathsf{P}(\mathbf{y})}\\ &= \frac{\mathsf{P}(\mathbf{y})\mathsf{P}(\mathbf{x}|\mathbf{y})\mathsf{P}(\mathbf{z}|\mathbf{y})}{\mathsf{P}(\mathbf{y})}\\ &= \mathsf{P}(\mathbf{x}|\mathbf{y})\mathsf{P}(\mathbf{z}|\mathbf{y})\end{aligned}$$
and therefore $\mathbf{x} \perp \mathbf{z}|\mathbf{y}$

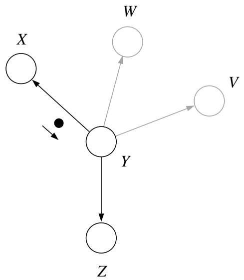
Explaining Away



- Q: When we condition on $\mathbf{y}$, are $\mathbf{x}$ and $\mathbf{z}$ independent?
$$\mathsf{P}(\mathbf{x},\mathbf{y},\mathbf{z}) = \mathsf{P}(\mathbf{x})\mathsf{P}(\mathbf{z})\mathsf{P}(\mathbf{y}|\mathbf{x},\mathbf{z})$$
- $\mathbf{x}$ and $\mathbf{z}$ are *marginally independent*, but given $\mathbf{y}$ they are *conditionally dependent*.
- This important effect is called *explaining away* (Berkson's paradox.)
- For example, flip two coins independently; let $\mathbf{x}$=coin1,$\mathbf{z}$=coin2. Let $\mathbf{y}$=1 if the coins come up the same and $\mathbf{y}$=0 if different.
- $\mathbf{x}$ and $\mathbf{z}$ are independent, but if I tell you $\mathbf{y}$, they become coupled!
- $\mathbf{y}$ is at the bottom of a v-structure, and so the path from $\mathbf{x}$ to $\mathbf{z}$ is active given $\mathbf{y}$ (information flows through).

Bayes Ball Algorithm

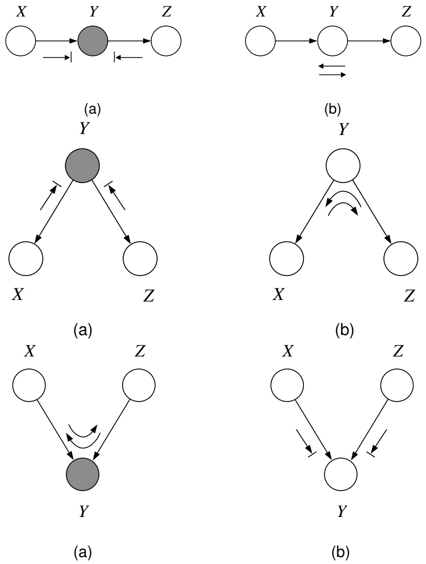
- To check if $\mathbf{x}_A \perp \mathbf{x}_B|\mathbf{x}_C$ we need to check if every variable in A is d-separated from every variable in B conditioned on all vars in C.
- In other words, given that all the nodes in $\mathbf{x}_C$ are clamped, when we wiggle nodes $\mathbf{x}_A$ can we change any of the node $\mathbf{x}_B$?
- The *Bayes-Ball Algorithm* is a such a d-separation test.
We shade all nodes $\mathbf{x}_C$, place balls at each node in $\mathbf{x}_A$ (or $\mathbf{x}_B$), let them bounce around according to some rules, and then ask if any of the balls reach any of the nodes in $\mathbf{x}_B$ (or $\mathbf{x}_A$).



So we need to know what happens when a ball arrives at a node $\mathbf{Y}$ on its way from $\mathbf{X}$ to $\mathbf{Z}$.

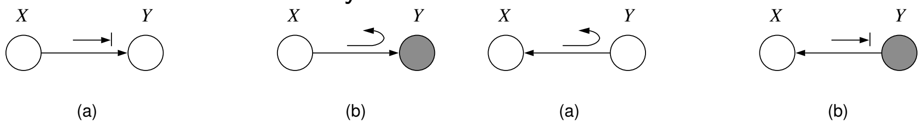
Bayes-Ball Rules

- The three cases we considered tell us rules:

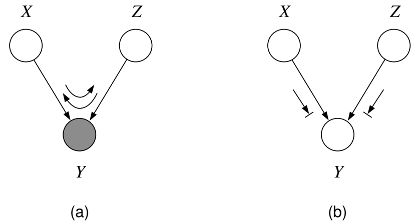


Bayes-Ball Boundary Rules

- We also need the boundary conditions:



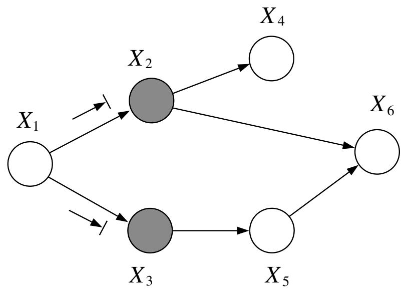
- Here's a trick for the explaining away case:
  If $\mathbf{y}$ *or any of its descendants* is shaded,
  the ball passes through.



- Notice balls can travel opposite to edge directions.

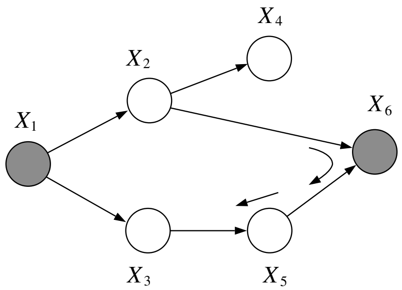
Examples of Bayes-Ball Algorithm

$$\mathbf{x}_1 \perp \mathbf{x}_6 | \{\mathbf{x}_2, \mathbf{x}_3\} \quad ?$$



Examples of Bayes-Ball Algorithm

$$\mathbf{x}_2 \perp \mathbf{x}_3 | \{\mathbf{x}_1, \mathbf{x}_6\} \quad ?$$



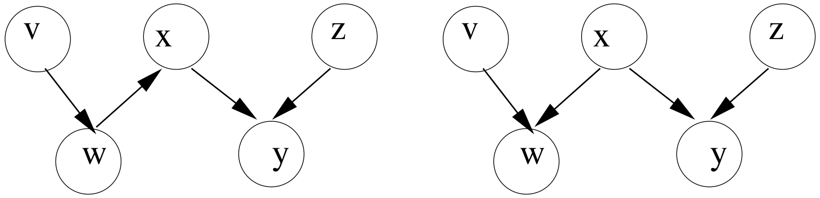
Notice: balls can travel opposite to edge directions.

I-equivalence

- Defn: Let $I(G)$ be the set of conditional independencies encoded by DAG G (for any parameterization of the CPDs):

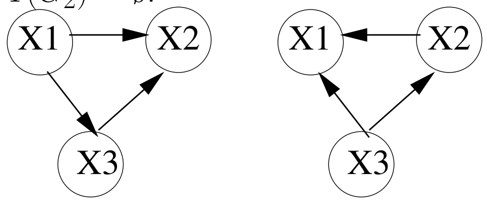
$$I(G) = \{(X \perp Y|Z) : Z \text{ d-separates X from Y}\}$$

- Defn: G_1 and G_2 are *I-equivalent* if $I(G_1) = I(G_2)$
- e.g., $X \rightarrow Y$ is I-equivalent to $X \leftarrow Y$
- Thm: If G_1 and G_2 have the same undirected skeleton and the same set of v-structures, then they are I-equivalent.



I-equivalence

- If G_1 is I-equivalent to G_2, they do not necessarily have the same skeleton and v-structures
- e.g., $I(G_1) = I(G_2) = \emptyset$:



- Corollary: We can only identify graph structure up to I-equivalence, i.e., we cannot always tell the direction of all the arrows from observational data.
- We will return to this issue when we discuss structure learning and causality.