D-Separation and the Bayes Ball Algorithm

Prof. Nicholas Zabaras
Center for Informatics and Computational Science
https://cics.nd.edu/
University of Notre Dame
Notre Dame, IN, USA

Email: nzabaras@gmail.com
URL: http://www.zabaras.com/

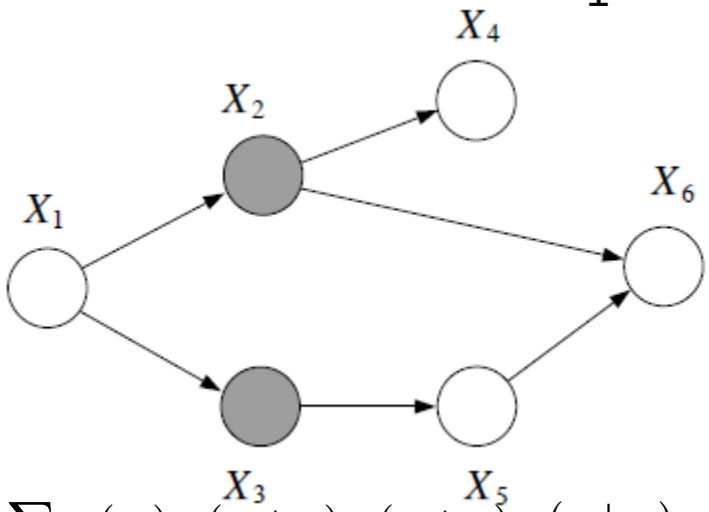
January 22, 2018

Contents

- Kevin Murphy, Machine Learning: A probabilistic Perspective, Chapter 10
- Chris Bishop, Pattern Recognition and Machine Learning, Chapter 8
- Jordan, M. I. (2007). An introduction to probabilistic graphical models. In preparation (Chapter 2) – Also review article entitled `Graphical Models'
- Video Lectures on Machine Learning, Z. Gahramani, C. Bishop and others.

Blocking the Dependency

- As another example, let us show that: $X_1 \perp X_6 \mid X_2, X_3$



$$p(x_6 \mid x_1, x_2, x_3) = \frac{\sum_{x_4, x_5} p(x_1)\, p(x_2 \mid x_1)\, p(x_3 \mid x_1)\, p(x_4 \mid x_2)\, p(x_5 \mid x_3)\, p(x_6 \mid x_2, x_5)}{\sum_{x_4, x_5, x_6} p(x_1)\, p(x_2 \mid x_1)\, p(x_3 \mid x_1)\, p(x_4 \mid x_2)\, p(x_5 \mid x_3)\, p(x_6 \mid x_2, x_5)}$$

$$= \frac{p(x_1)\, p(x_2 \mid x_1)\, p(x_3 \mid x_1) \sum_{x_4} p(x_4 \mid x_2) \sum_{x_5} p(x_5 \mid x_3)\, p(x_6 \mid x_2, x_5)}{p(x_1)\, p(x_2 \mid x_1)\, p(x_3 \mid x_1) \sum_{x_4} p(x_4 \mid x_2) \sum_{x_5} p(x_5 \mid x_3) \sum_{x_6} p(x_6 \mid x_2, x_5)}$$

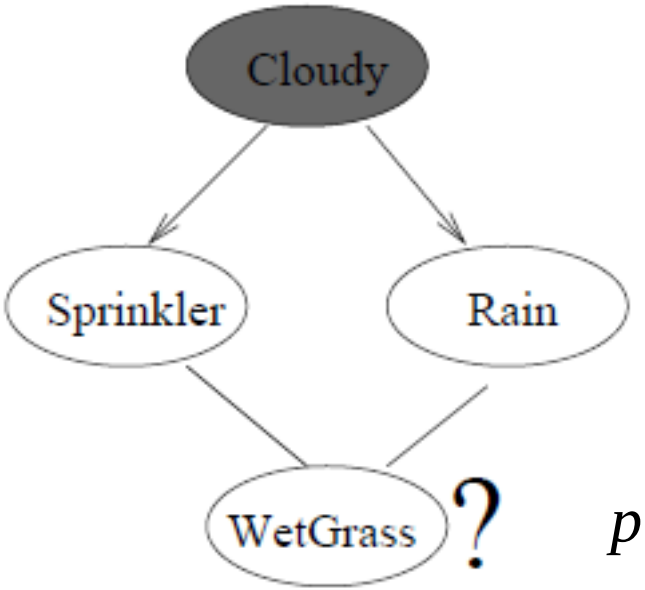
$$= \sum_{x_5} p(x_5 \mid x_3)\, p(x_6 \mid x_2, x_5) = \sum_{x_5} p(x_5 \mid x_2, x_3)\, p(x_6 \mid x_2, x_3, x_5)$$

$$= \sum_{x_5} p(x_6, x_5 \mid x_2, x_3) = p(x_6 \mid x_2, x_3)$$

Causal Reasoning - Prediction

- Given a set of observed variables, we want to estimate the values of hidden variables - this is an inference problem.

- *From causes to effects (Causal Reasoning – Prediction, Uncertainty propagation):* How likely is for the grass to be wet if it is cloudy?

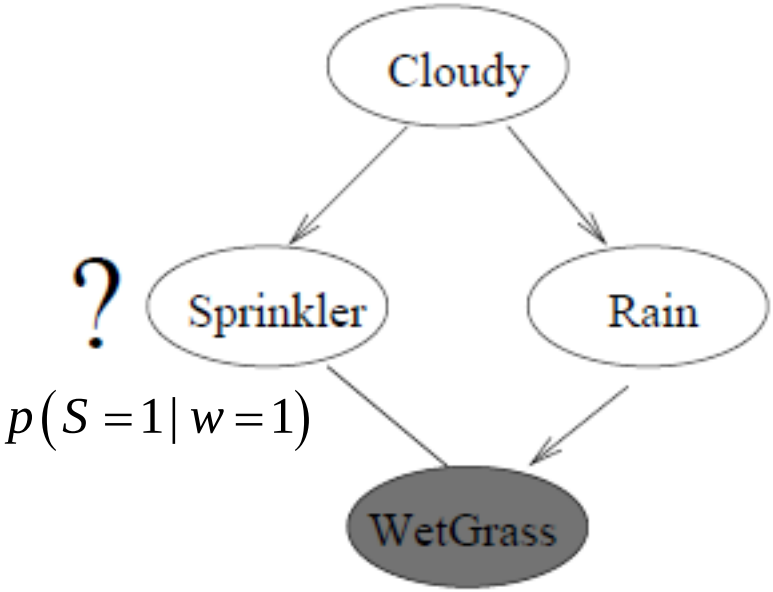


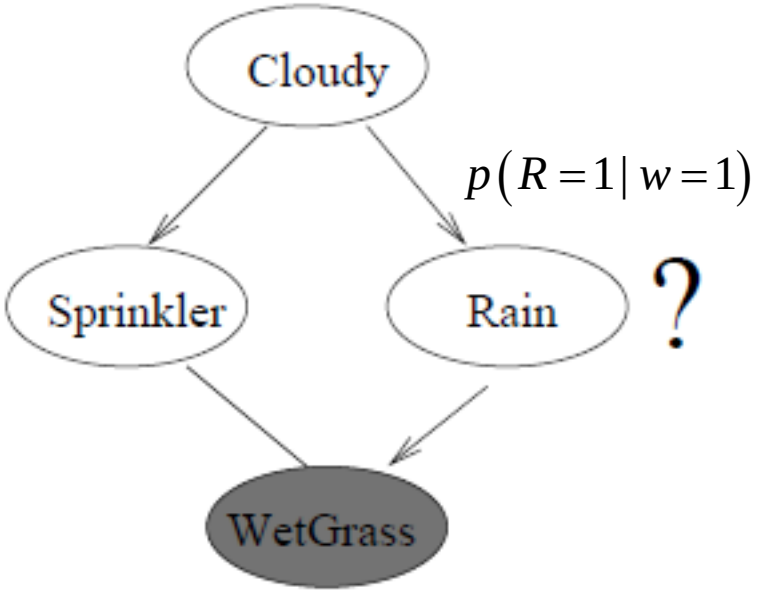
$$p(W=1 \mid c=1)$$

Diagnostic or Evidential Reasoning

- Given a set of observed variables, we want to estimate the values of hidden variables - this is an inference problem.

- *From effects to causes (Diagnostic or Evidential Reasoning - Explanation):* If the grass is wet, how likely is that the sprinkler is on or that it rained?





$$\max_{r,s} p(S=s, R=r \mid w=1) \; or$$

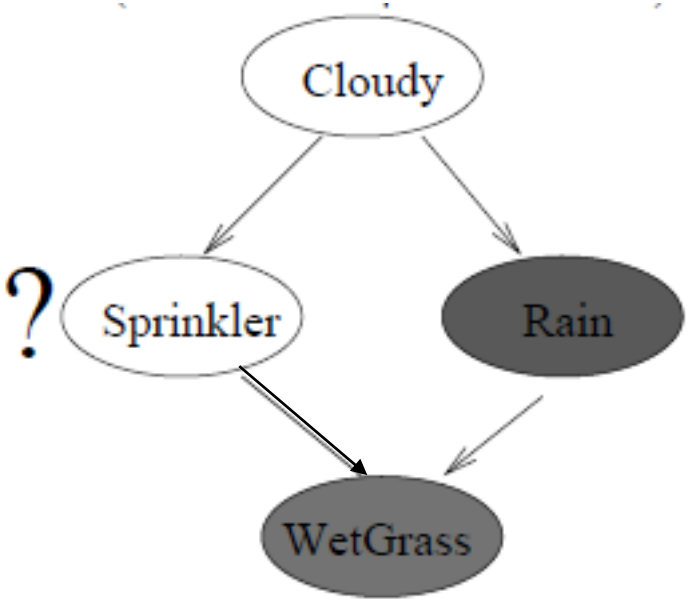
$$\max_{r} p(R=r \mid w=1) \overset{?}{\geq} \max_{s} p(S=s \mid w=1)$$

Explaining Away – Inter-causal Reasoning

- Consider the same example with data pointing that the grass is wet and that it rained. Can we infer the following?

$$P(S = 1|w = 1, r = 1) < P(S = 1|w = 1)$$

- The observation that it has rained has explained away why the grass is wet.
- *S and R become dependent given W even though they are marginally independent.*
- *Different causes of the same effect can interact.*
- One causal factor for the WetGrass variable – Rain – gives us information about another – Sprinkler.

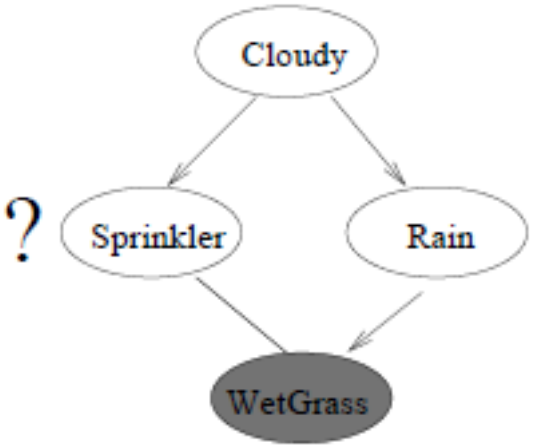


Inference by Marginalizing the Joint

- We can answer an query by marginalization of the joint distribution using the factorization implied by the graph. For example:

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

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

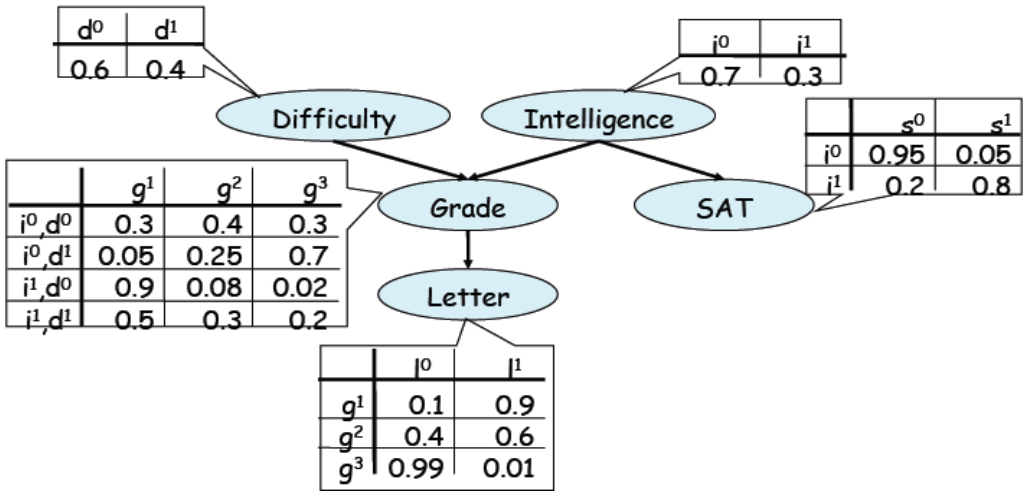


$$p(C,S,R,W) = p(C)\, p(S \mid C)\, p(R \mid C)\, p(W \mid S,R)$$

Local Markov Property

- Local *Markov property: node is conditionally independent of its n**on-descendants** given its parents*

$$\{X_v \perp X_{N(v)}\} | X_{pa(v)}$$



$$L \perp I, D, S \mid G$$

$$S \perp D, G, L \mid I$$

$$G \perp S \mid I, D$$

$$I \perp D$$

$$D \perp I, S$$

- The parents of a variable shield it from probabilistic influence that is causal in nature. *Once we know the value of the parents, no information relating directly or indirectly to its parents or other ancestors can influence my beliefs about it.*
- However, information about its descendants can change my beliefs about it via the evidential reasoning process, e.g. $G \not\perp L \mid I, D$

Inference in DAGs

- DAGs provide a compact way to define joint probability distributions. We can use such a joint distribution to perform **probabilistic inference**.

- This refers to the task of estimating unknown quantities from known quantities.

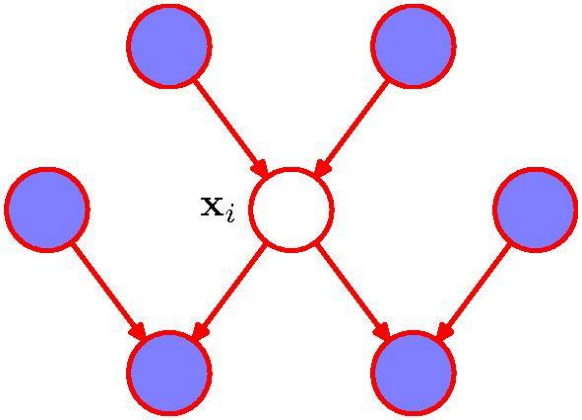
- *For example, in HMMs estimate the hidden states (e.g., words) from the observations (e.g., speech signal).*

- *In the genetic linkage analysis one of the goals is to estimate the likelihood of the data under various DAGs, corresponding to different hypotheses about the location of the disease-causing gene.*

- For posing the inference problem let us suppose we have a set of correlated random variables with joint distribution $p(\boldsymbol{x}_{1:V} | \boldsymbol{\theta})$. (Assume at this point that $\boldsymbol{\theta}$ are known).

Conditioning on Evidence

- The variables in our problem may be hidden (latent variables) or visible (observed data –shown with dark circles)



- Latent variables are usually introduced to allow a richer class of distributions (i.e. add complexity to the model, e.g. in mixture of Gaussians). In other occasions, latent variables may have a physical meaning.
- We usually are interested to compute the posterior of what we want to predict conditional on the given data where we integrate out things of no interest (the latent variables)

Inference in DAGs

- Let us partition this vector into the *visible variables* x_v, which are observed, and *the hidden variables,* x_h, which are unobserved.

- Inference refers to *computing the posterior distribution of the unknowns given the knowns:*

$$p(\boldsymbol{x}_h \mid \boldsymbol{x}_v, \theta) = \frac{p(\boldsymbol{x}_h, \boldsymbol{x}_v \mid \theta)}{p(\boldsymbol{x}_v \mid \theta)} = \frac{p(\boldsymbol{x}_h, \boldsymbol{x}_v \mid \theta)}{\sum_{\boldsymbol{x}_h'} p(\boldsymbol{x}_h', \boldsymbol{x}_v \mid \theta)}$$

- We are conditioning on the data by clamping the visible variables to their observed values, $\mathbf{x}_v$, and then normalizing, to go from $p(\boldsymbol{x}_h, \boldsymbol{x}_v)$ to $p(\boldsymbol{x}_h | \boldsymbol{x}_v)$.

- The normalization constant $p(\boldsymbol{x}_v | \boldsymbol{\theta})$ (likelihood of the data) is also known as the *probability of the evidence.*

Inference in DAGs

- Often some of the hidden variables are of interest. Partition the hidden variables into *query variables,* x_q and *nuisance variables,* x_n. We can marginalize out the nuisance variables:

$$p(\boldsymbol{x}_q \mid \boldsymbol{x}_v, \theta) = \sum_{\boldsymbol{x}_n} p(\boldsymbol{x}_q, \boldsymbol{x}_n \mid \boldsymbol{x}_v, \theta)$$

- We can perform these *operations for a multivariate Gaussian in* $\mathcal{O}(V^3)$ *time, where V is the number of variables.*

- For discrete random variables each with *K* states, if the joint distribution is represented as a multi-dimensional table, we can perform these operations exactly but at $\mathcal{O}(K^V)$ time.

- We can exploit the factorization encoded by the GM to *perform these operations in* $\mathcal{O}(VK^{w+1})$ *time, where w is a quantity known as the treewidth of the graph.*

- *The treewidth measures how "tree-like" the graph is.* If the graph is a tree, *w* = 1, inference takes time linear in V.

- *For general graphs, exact inference takes time exponential in V.*

Learning (of the Parameters θ)

- It is common to distinguish between inference and learning.
- Inference means computing (functions of) $p(\mathbf{x}_h|\mathbf{x}_v, \boldsymbol{\theta})$, where v are the visible nodes, h are the hidden nodes, and $\boldsymbol{\theta}$ are the parameters of the model, assumed to be known.
- *Learning usually means computing a MAP estimate of the parameters given data:*

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{argmax} \left(\sum_{i=1}^{N} logp(\boldsymbol{x}_{i,v}|\boldsymbol{\theta}) + logp(\boldsymbol{\theta}) \right)$$

  $\mathbf{x}_{i,v}$ are the visible variables in case i. For a uniform prior, $p(\boldsymbol{\theta}) \propto 1$, this reduces to the MLE. In a frequentistic approach, this can be seen as some form of penalized maximum likelihood.

- *In a Bayesian setting there is no distinction between inference and learning* – one adds the parameters as nodes to the graph, conditions on the data $\mathcal{D}$ and then infers the values of all the nodes.

Hidden Variables Vs. Parameters

- Main difference between hidden variables and parameters:
  - *the number of hidden variables grows with the amount of training data* (there is usually a set of hidden variables for each observed data case),
  - whereas the number of parameters in usually fixed (in a parametric model).
- Thus *we must integrate out the hidden variables to avoid over fitting, but we may be able to get away with point estimation techniques for parameters which are fewer in number*.

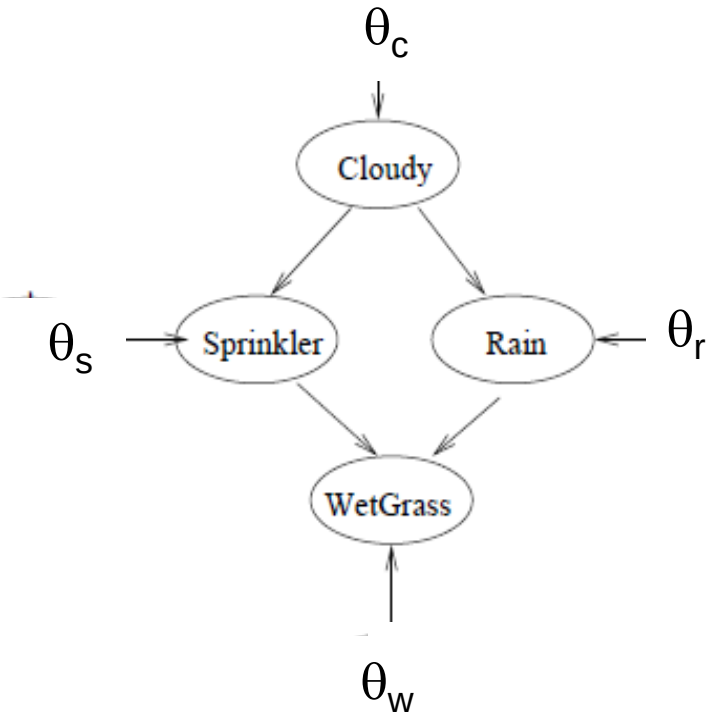
Structure Learning

- We assume that we have iid training data where each node in the graph is fully observed:

$$\mathcal{D} = \left\{ c^i, s^i, r^i, w^i \right\}, i = 1, \ldots, N$$

- In a Bayesian approach, we treat the graph G as a random variable and compute the posterior $p(G|\mathcal{D})$.

- In a frequentist framework, we treat G as unknown constant and find the best estimate through some form of an optimization process such as maximum penalized likelihood:



$$\hat{G} = \underset{G}{argmax}(logp(\mathcal{D}|G) - \lambda C(G))$$

Generative Models and Ancestral Sampling

- Suppose the variables in a DAG have been ordered such that each node has a higher number than any of its parents.
  Our goal is to draw a sample $\hat{x}_1, \ldots, \hat{x}_K$, from the joint distribution.
  The steps are summarized as follows:

  1) We start with the lowest-numbered node and draw a sample from the distribution $p(x_1)$

  2) For node *n* we draw a sample from the conditional distribution $p(x_n|Pa_n)$ where the parents have been set to their sampled values.

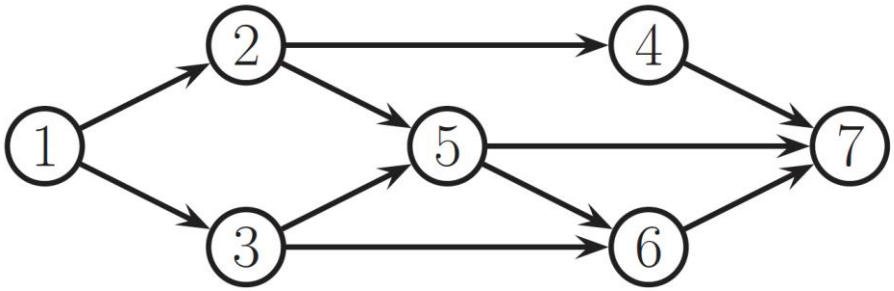
- To obtain a sample from some marginal distribution of a subset of $\hat{x}_1, \ldots, \hat{x}_K$, we simply take the sampled values for the required nodes and ignore the remaining.

Markov blanket and full conditionals

- The set of nodes that renders a node t conditionally independent of all the other nodes in the graph is called t's Markov blanket.

- One can show that Markov blanket of a node in a DGM is equal to the parents, the children, and the co-parents

$$mb(t) \triangleq ch(t) \cup pa(t) \cup copa(t)$$



Other Markov properties of DGMs

- A special case of this property is when we only look at predecessors of a node according to some topological ordering

$$t \perp pred(t) \backslash \mathrm{pa}(t) | \mathrm{pa}(t)$$

This is true because $pred(t) \subseteq nd(t)$.

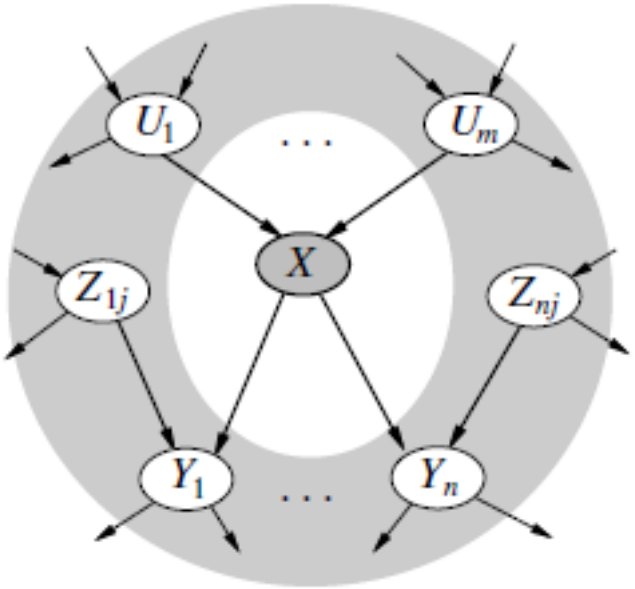
- This is called the ordered Markov property

- Note that, satisfying global Markov property implies satisfaction of local Markov property and topological Markov property. The vice-versa is also true.

Local Markov Property: Markov Blanket

- *A node is conditionally independent of all other nodes given its parents, children and children's parents (co-parents).*

- This is the Markov Blanket. A node is independent of all other nodes given its Markov blanket.



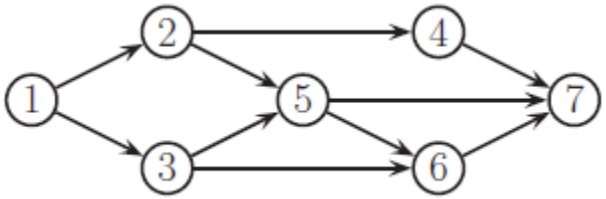
Markov Blanket and Full Conditionals

- The set of nodes that renders a node t conditionally independent of all the other nodes in the graph is called t's **Markov blanket** denoted as mb(t).

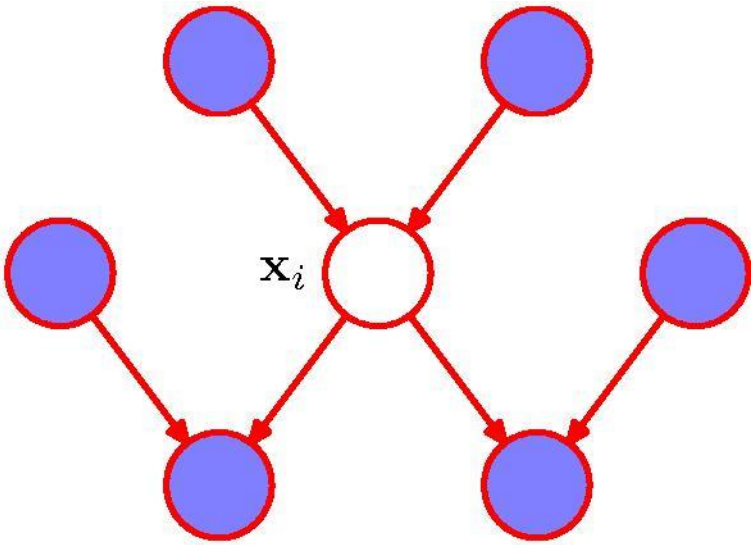
- *One can show that the Markov blanket of a node in a DGM is equal to the parents, the children, and the co-parents (*nodes who are parents of its children):

$$\text{mb}(t) = \text{ch}(t) \cup \text{pa}(t) \cup \text{copa}(t)$$

- In the Figure, mb(5) = *{*6, 7*}* ∪ *{*2, 3*}* ∪ *{*4*}* = *{*2, 3, 4, 6, 7*}* where *4 is a co-parent of 5 because they share a common child, namely 7*.



The Markov Blanket: Directed Graphs



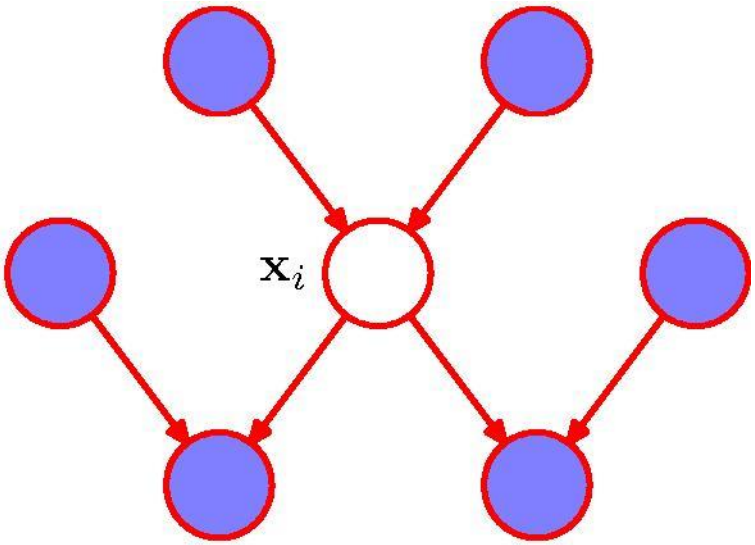
$$p\left(x_i \middle| x_{\{j\neq i\}}\right) = \frac{p\left(x_1,...,x_M\right)}{\int p\left(x_1,...,x_M\right)dx_i}$$

$$= \frac{\prod_k p\left(x_k \middle| Pa_k\right)}{\int \prod_k p\left(x_k \middle| Pa_k\right)dx_i}$$

Factors independent of x_i cancel between numerator and denominator.

- The only factors that remain will be $p(x_i|Pa_i)$, together with the conditional distributions $p(x_k|Pa_k)$ for which x_i is a parent of x_k.
- The conditional $p(x_i|Pa_i)$ will depend on the parents of node x_i, whereas the conditionals $p(x_k|Pa_k)$ will depend on the children of x_i and on the co-parents of x_i , i.e. parents of node x_k other than x_i.

The Markov Blanket: Directed Graphs



$$p\left(x_i \middle| x_{\{j\neq i\}}\right) = \frac{p\left(x_1,\ldots,x_M\right)}{\int p\left(x_1,\ldots,x_M\right)dx_i}$$

$$= \frac{\prod_k p\left(x_k \middle| Pa_k\right)}{\int \prod_k p\left(x_k \middle| Pa_k\right)dx_i}$$

Factors independent of x_i cancel between numerator and denominator.

Markov blanket: include the set of nodes comprising **the parents, the children and the co-parents (other parents of the children)** (explaining away effect).

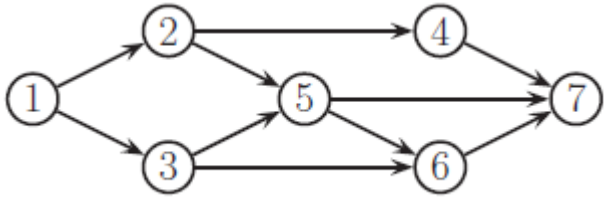
We can think of the Markov blanket of a node x_i as being the minimal set of nodes that isolates x_i from the rest of the graph.

Full Conditional

- To see why the co-parents are in the Markov blanket, note that when we derive $p(x_t|\mathbf{x}_{-t}) = p(x_t, \mathbf{x}_{-t})/p(\mathbf{x}_{-t})$, all the terms that do not involve x_t will cancel out between numerator and denominator, so we are left with a product of CPDs which contain x_t in their **scope**. Hence

$$p(x_t \mid \boldsymbol{x}_{-t}) \propto p(x_t \mid \boldsymbol{x}_{pa(t)}) \prod_{s \in ch(t)} p(x_s \mid \boldsymbol{x}_{pa(s)})$$

- In our example, we have



$$p(x_5 \mid \boldsymbol{x}_{-5}) \propto p(x_5 \mid x_2, x_3)\, p(x_6 \mid x_3, x_5)\, p(x_7 \mid x_4, x_5, x_6)$$

- The resulting expression is called *t's full conditional and it is important when implementing Gibbs sampling.*

Summary of Markov Properties of DGMs

- We have now described up to now two Markov properties for DAGs:

L: Directed Local Markov Property $t \perp nd(t) \setminus pa(t) \mid pa(t)$

O: Ordered Markov Property $t \perp pred(t) \setminus pa(t) \mid pa(t)$

- It is obvious that $L \Rightarrow O$.
- What is less obvious, but nevertheless true, is that $O \Rightarrow L$. Hence *these properties are equivalent*.
- Furthermore, any distribution p that is Markov wrt G can be factorized as;

F: Factorization Property

$$p(\boldsymbol{x}_{1:V} \mid G) = \prod_{t=1}^{V} p(x_t \mid \boldsymbol{x}_{pa(t)})$$

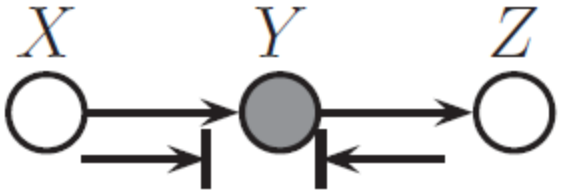
- Clearly $O \Rightarrow F$, but one can show that the converse also holds.

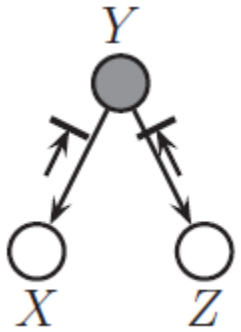
- Koller, D. and N. Friedman (2009). *Probabilistic Graphical Models: Principles and Techniques*. MIT Press.

D-Separation

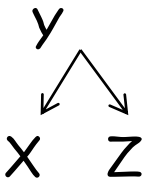
- *An undirected path P is **d-separated** by a set of nodes E (containing the evidence) iff at least one of the following holds:*

1. P contains a chain, $X \rightarrow Y \rightarrow Z$ or $X \leftarrow Y \leftarrow Z$, where $Y \in E$

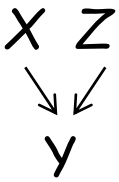


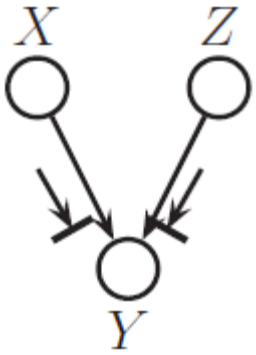


2. P contains a fork (as shown), where $Y \in E$



3. P contains a **v-structure** (as shown) where *Y* is not in *E* and nor is any descendant of *Y*.





- Lauritzen, S. L. and D. J. Spiegelhalter (1988). Local computations with probabalities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society* **50**, 157–224.
- Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems.* Morgan Kaufmann.

D-Separation

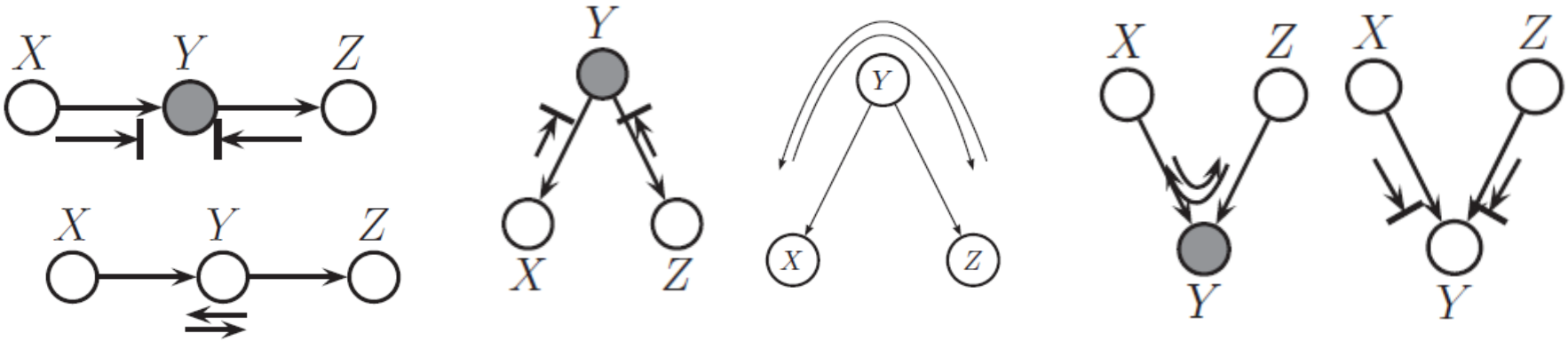
- A *set of nodes A* is d-separated from a different set of nodes *B* given a third observed set *E* iff each undirected path from every node $a \in A$ to every node $b \in B$ is d-separated by *E*.

- Finally, we define the CI properties of a DAG as follows:

$$\mathbf{x}_A \perp_G \mathbf{x}_B | \mathbf{x}_E \iff A \text{ is d-separated from } B \text{ given } E$$

D-Separation and The Bayes Ball Algorithm

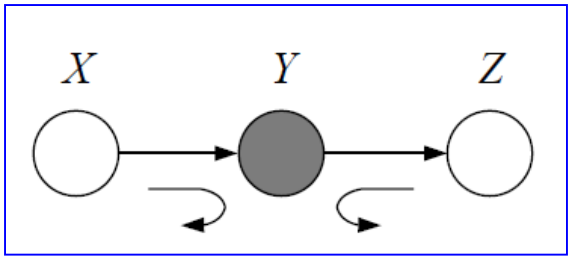
- *The Bayes ball algorithm* is a simple way to see if *A* is d-separated from *B* given *E*.
- We "shade" all nodes in *E* (observed). We then place "balls" at each node in *A*, let them "bounce around" according to some rules, and then ask if any of the balls reach any of the nodes in *B*. Note that *balls can travel opposite to edge directions.*
- We see that a ball can pass through a chain, but not if it is shaded in the middle. Similarly, a ball can pass through a fork, but not if it is shaded in the middle. However, a ball cannot pass through a v-structure, unless it is shaded in the middle.

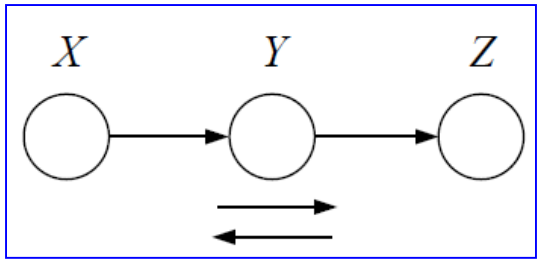


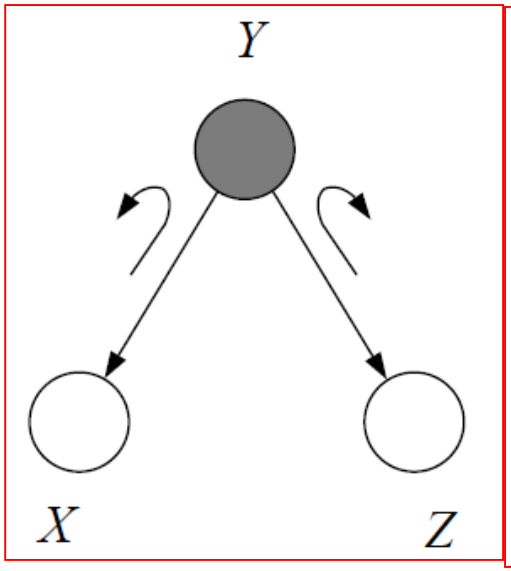
- Shachter, R. (1998). Bayes-ball: The rational pastime (for determining irrelevance and requisite information in belief networks and influence diagrams). In *UAI*.

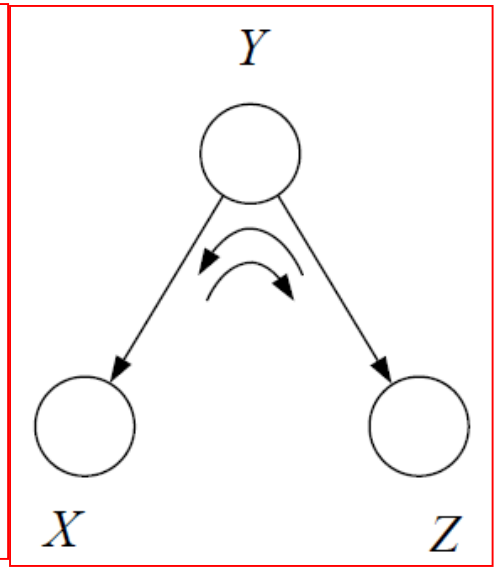
The Bayes Ball Algorithm Rules

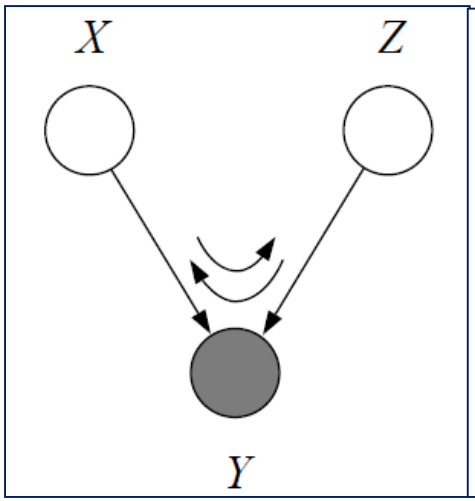
- It provides a set of rules as to what happens when a ball arrives from X to Y en route to Z.

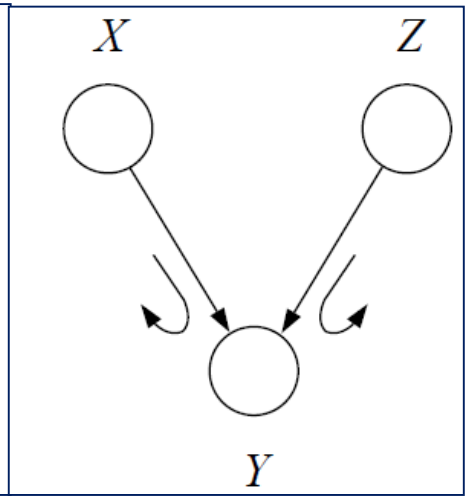




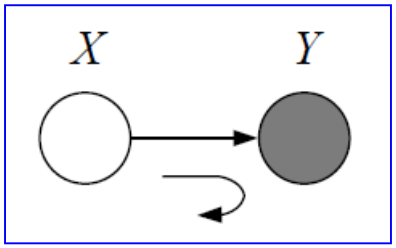


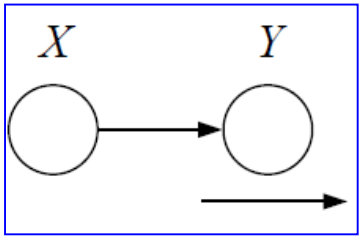






Bayes Ball: Boundary Conditions



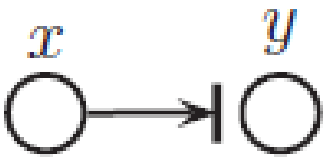


Boundary condition (leaf node)

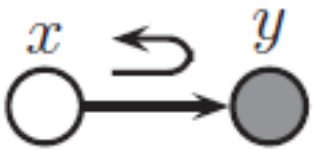
- Balls can travel opposite to edge directions.

Bayes Ball – Boundary Conditions

- We also need the two boundary conditions shown below,

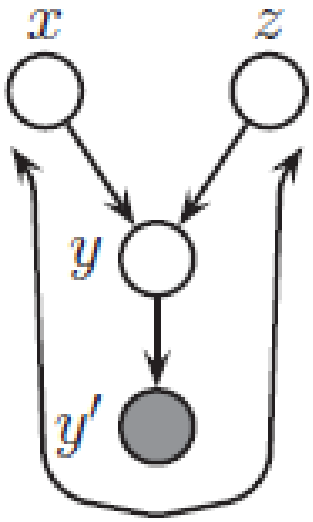


Ball dies



Ball bounces back

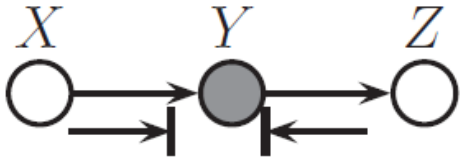
- To understand from where these rules come from, consider Y' is a noise-free copy of Y.
- Then if we observe Y', we effectively observe Y as well, so the parents X and Z have to compete to explain this. So if we send a ball down $X \rightarrow Y \rightarrow Y'$, it should "bounce back" up along $Y' \rightarrow Y \rightarrow Z$.
- However, if Y and all its children are hidden, the ball does not bounce back.



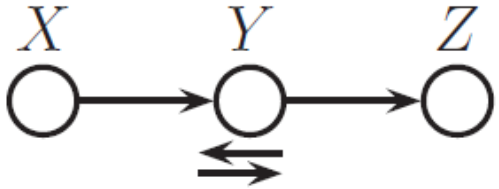
D-Separation : Head-To-Tails (Chain)

- We can justify the 3 rules of Bayes ball as follows. First consider a Markov chain structure $X \rightarrow Y \rightarrow Z$, which encodes

$$p(x, y, z) = p(x)p(y|x)p(z|y)$$



- When we condition on y, are x and z independent?



$$p(x, z \mid y) = \frac{p(x)\,p(y \mid x)\,p(z \mid y)}{p(y)} = \frac{p(x, y)\,p(z \mid y)}{p(y)} = p(x \mid y)\,p(z \mid y)$$

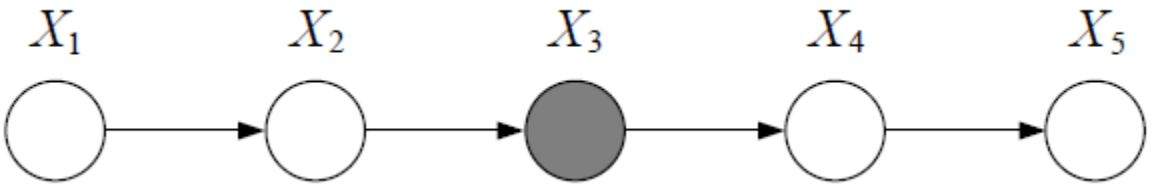
- Therefore $x \perp z|y$. So observing the middle node of chain breaks it in two (as in a Markov chain).
- Information on Y blocks the path from X to Z (graph separation).
- Think of X as the past, Y as the present and Z as the future.

Head-to-tail + observed ⇒ blocked
Head-to-tail + unobserved ⇒ not blocked

Reachability Algorithm

- The algorithm is a reachability algorithm: we shade the nodes X_C, place a ball at each of the nodes X_A, let the balls bounce around the graph according to the set of rules, and ask whether any of the balls reach one of the nodes in X_B. If none of the balls reach X_B, then we assert that $X_A \perp\!\!\!\perp X_B \mid X_C$ If a ball reaches X_B then we assert that $X_A \perp\!\!\!\perp X_B \mid X_C$ is not true.

- Using this algorithm, one can show for example the following independent relations for the Markov Chain below:



$$X_{i+1} \perp\!\!\!\perp \{X_1, X_2, \ldots, X_{i-1}\} \mid X_i$$

- In general, we obtain the conditional independence of any subset of "future" nodes from any subset of "past" nodes given any subset of nodes that separate these subsets.

D-Separation: Tail-To-Tail Nodes

- Now consider the tent structure $X \leftarrow Y \rightarrow Z$ *(hidden cause)* :

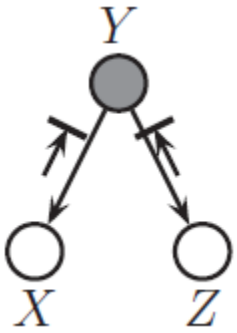
$$p(x, y, z) = p(y)p(x|y)p(z|y)$$

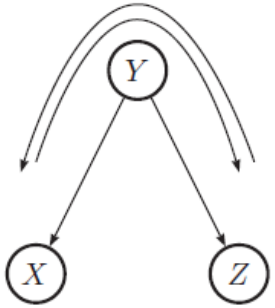
- When we condition on *y*, are *x* and *z* independent? We have

$$p(x, z \mid y) = \frac{p(x,\ y,\ z)}{p(y)} = \frac{p(y)\,p(x \mid y)\,p(z \mid y)}{p(y)} = p(x \mid y)\,p(z \mid y)$$

and therefore $x \perp z|y$.

- So observing a root node separates its children (as in a naive Bayes classifier).





$$p(x, y, z) = p(y)\,p(z|y)\,p(x|y)$$

$$p(x, z) = \sum_y p(z|y)\,p(x|y)\,p(y)$$

$$\neq p(x)\,p(z)$$

- Information on Y blocks the path from X to Z (graph separation)

Tail-to-tail + observed ⇒ blocked

Tail-to-tail + unobserved ⇒ not blocked

Head-To-Head Nodes - Explaining Away

- Finally consider a v-structure $X \rightarrow Y \leftarrow Z$. The joint is

$$p(x, y, z) = p(x)p(z)p(y|x, z)$$

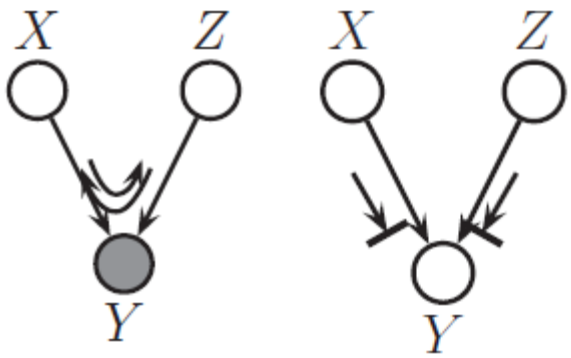
- When we condition on *y*, are *x* and *z* independent? We have

$$p(x, z \mid y) = \frac{p(x)p(z)p(y \mid x, z)}{p(y)} \qquad \text{so } x \not\perp z|y.$$

- However, in the unconditional distribution, we have $p(x, z) = p(x)p(z)$ i.e. x and z are marginally independent.

- Conditioning on a common child at the bottom of a v-structure makes its parents become dependent (the path from x to z is active – information can flow through)

Head-to-head + observed ⇒ not blocked



$$p(x, y, z) = p(x)\,p(z)\,p(y|x, z)$$

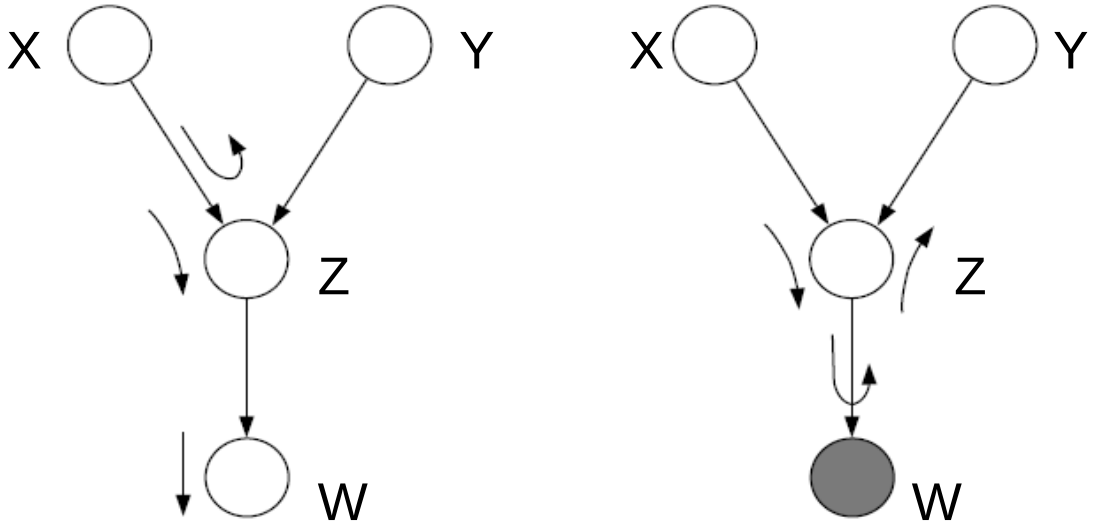
$$p(x, z) = p(x)\,p(z)$$

$$x \perp z$$

Head-to-head + unobserved ⇒ blocked

Head-To-Head Nodes - Explaining Away

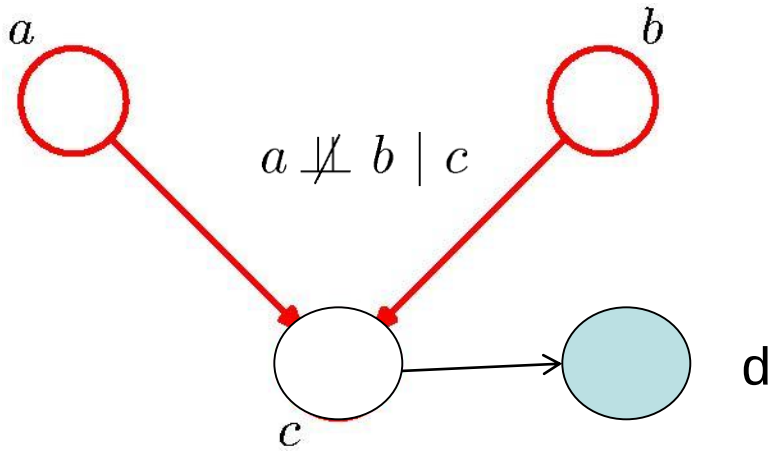
- A ball cannot pass from *X* to *Y* when no observations are made on *Z* or *W*.
- However, a ball can pass from *X* to *Y* when *W* is observed.



- When *Z and W are not observed*, a ball from X cannot reach W and then return to Z and then go to Y since it is killed once it reaches W (boundary condition).

Head-to-head + all descendants unobserved ⇒ blocked
Head-to-head + any descendant observed ⇒ not blocked

Directed Graphs: Head-To-Head Nodes



$$p(a,b|c) \neq p(a|c)\,p(b|c)$$

- Unobserved head-to-head node c blocks the path from a to b
- Once c is observed, the path is unblocked
- Observation of any descendant d of c also unblocks the path
  – Node d propagates evidence to c and that unblocks the path from a to b and a and b become correlated.

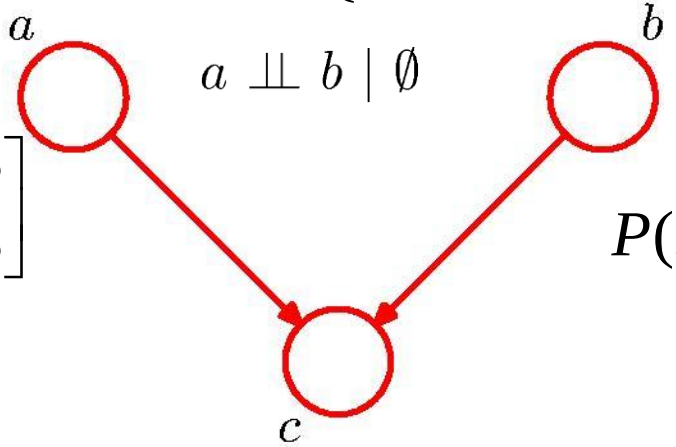
Explaining Away

- Consider the following example of explaining away.
- Suppose we toss two coins representing the binary numbers 0 and 1, and we observe the sum of their values.
- A priori, the coins are independent, but once we observe their sum, they become coupled
  - e.g., if the sum is 1, and the first coin is 0, then we know the second coin is 1

Directed Graphs: Head-To-Head Nodes

- Consider two coins flipped independently of each other.
- Let c be the outcome of the comparison of these two:

$$c = \delta(a,b) = \begin{cases} 1, if\ a = b \\ 0, if\ a \neq b \end{cases}$$



$$P(a): \begin{matrix} a=0 \\ a=1 \end{matrix} \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix} \qquad P(b): \begin{matrix} b=0 \\ b=1 \end{matrix} \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}$$

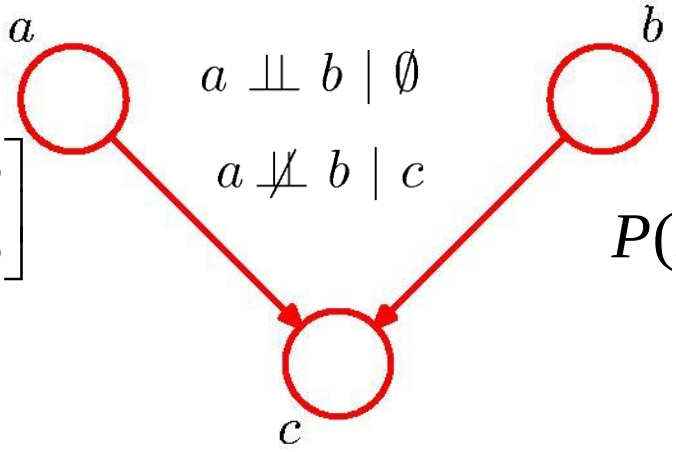
$$ab = 00 \quad ab = 01 \quad ab = 10 \quad ab = 11$$

$$P(c \mid a,b): \begin{matrix} c=0 \\ c=1 \end{matrix} \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}$$

Head-To-Head Nodes: Induced Dependence

- The graph permits us to infer that *a and b are marginally independent.*
- The graph can also be used to infer that *a & b are dependent on c*.

This is what we call induced dependence.

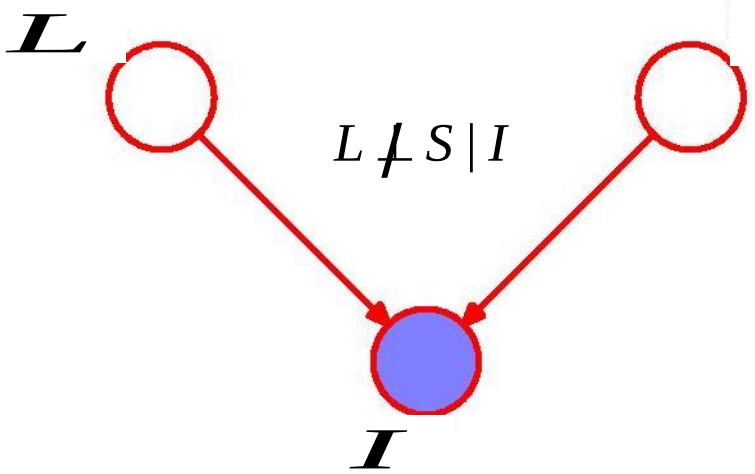


$$P(a): \begin{matrix} a=0 \\ a=1 \end{matrix}\begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix} \qquad P(b): \begin{matrix} b=0 \\ b=1 \end{matrix}\begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}$$

$$P(c \mid a,b): \begin{matrix} c=0 \\ c=1 \end{matrix}\begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}$$

$ab = 00 \quad ab = 01 \quad ab = 10 \quad ab = 11$

Head-to-Head Node: Explaining away



$$p(I,L,S) = p(I|L,S)\,p(L)\,p(S)$$

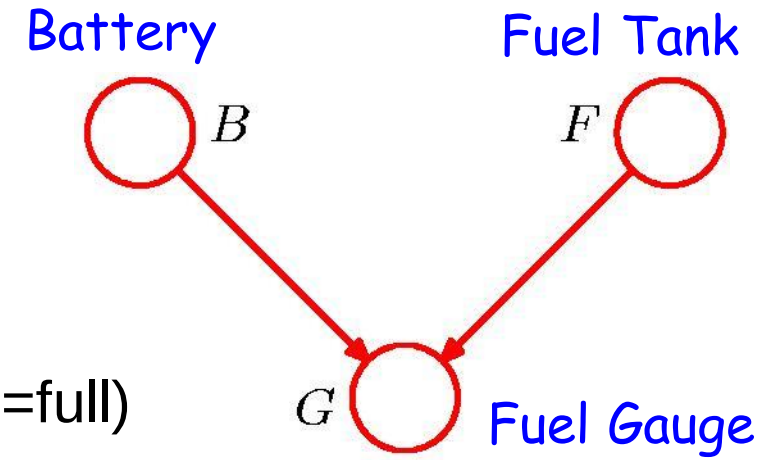
$$p(L,S) = p(L)\,p(S)$$

$$p(L,S \mid I) \neq p(L|I)\,p(S|I)$$

- Consider
  - I to be the image color
  - L the lighting color
  - S the surface color
- If we observe that the pixel I is blue, then we know that if the surface color S is blue, the color of the light L might be blue or white.
- L and S become correlated.

Explaining Away: Another Example

Consider a fuel system problem



B = Battery (0=flat, 1=fully charged)
F = Fuel Tank (0=empty, 1=full)
G = Fuel Gauge Reading (0=empty, 1=full)

Given the state of fuel tank and the battery, the fuel gauge reads full with probabilities given by

$$p(G=1|B=1,F=1) = 0.8$$
$$p(G=1|B=1,F=0) = 0.2$$
$$p(G=1|B=0,F=1) = 0.2$$
$$p(G=1|B=0,F=0) = 0.1$$

Set priors:

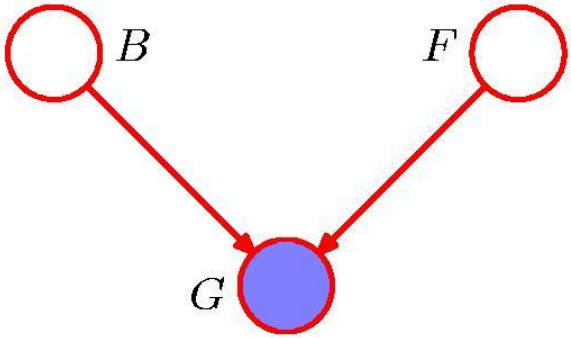
$$p(B=1) = 0.9$$
$$p(F=1) = 0.9$$

Explaining Away: Another Example

$$p(F=0|G=0) = \frac{p(F=0,G=0)}{p(G=0)}$$

$$= \frac{\sum_B p(G=0|B,F=0)p(B)\,p(F=0)}{\sum_B \sum_F p(G=0|B,F)p(B)p(F)} = \frac{0.081}{0.315}$$

$$\simeq 0.257 \;>\; p(F=0)$$



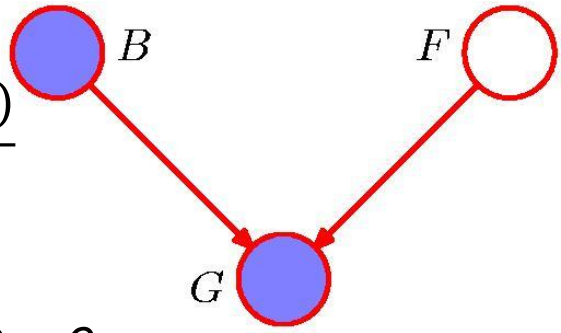
$$p(B,F,G) = p(G|B,F)\,p(B)\,p(F)$$

$$p(F,G) = \sum_B p(G|B,F)\,p(B)\,p(F)$$

Probability of an empty tank increased by observing G = 0.

$$p(F=0|G=0,B=0) = \frac{p(G=0|B=0,F=0)p(F=0)}{\sum_F p(G=0|B=0,F)p(F)}$$

$$\simeq 0.111 \;<\; p(F=0|G=0)$$

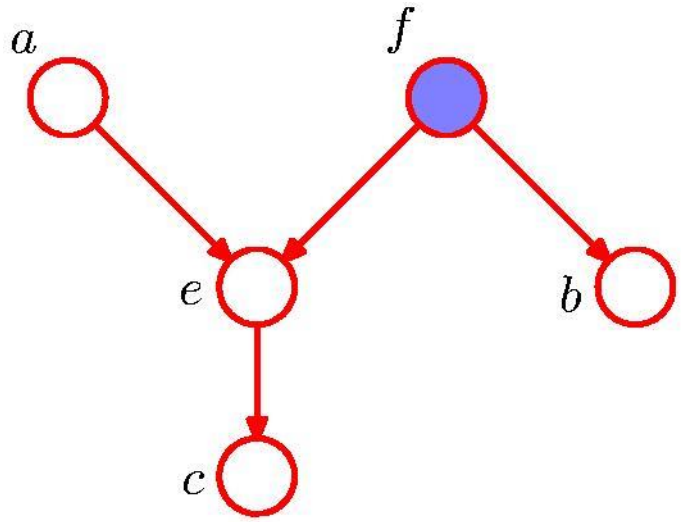


Probability of an empty tank reduced by observing *B = 0*.

Finding that the battery is flat explains away the observation that the fuel gauge reads empty. The state of the fuel tank and that of the battery have become dependent when observing the gauge.

D-Separation: Example

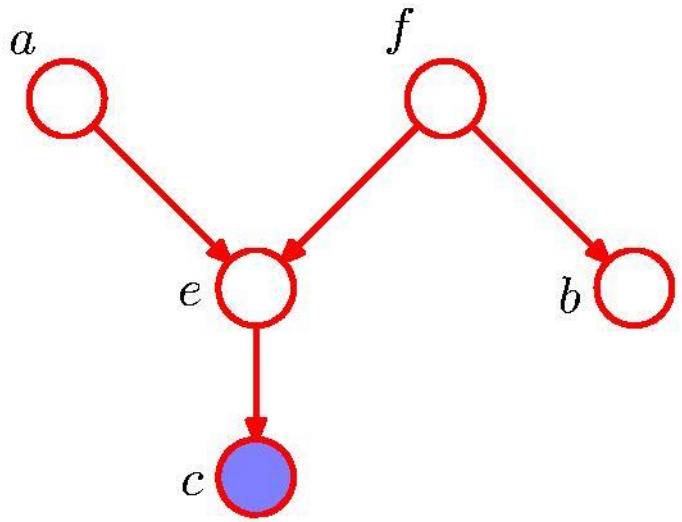
$a \perp\!\!\!\perp b \mid f$



- f is tail-to-tail and observed – so f blocks the path
- e is head-to-head and together with his descendant are un-observed. So they also block the path "a-e-f-b".

D-Separation: Example

$a \not\perp\!\!\!\perp b \mid c$



- Conditional independence if and only if all possible paths are blocked.
- f is tail-to-tail and unobserved – so it does not block the path
- e is head-to-head and has a descendant observed. So it unblocks the path.
- So there is one unblocked path a-e-f-b.

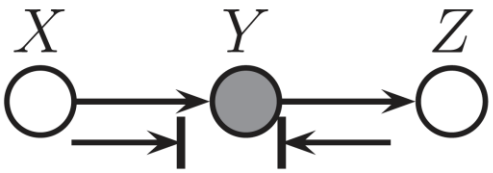
Summary: D-Separation & Bayes Ball

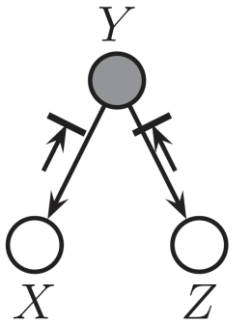
- We say an *undirected path* P is **d-separated** by a set of nodes E if at least one of following holds:
  - P contains a chain, $s \rightarrow m \rightarrow t$ or $s \leftarrow m \leftarrow t$ where $m \in E$
  - P contains a tent or fork, $s \swarrow m \searrow t$ where $m \in E$
  - P contains a collider or v-structure, $s \searrow m \swarrow t$
- We say a *set of nodes* A is d-separated from a different set of nodes B given a third observed set E if each undirected path from each node $a \in A$ to every node $b \in B$ is d-separated by E.

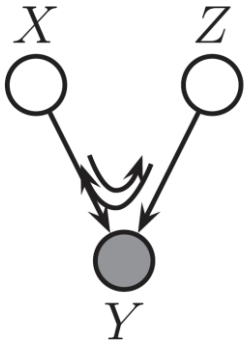
D-Separation & the Bayes Ball Algorithm

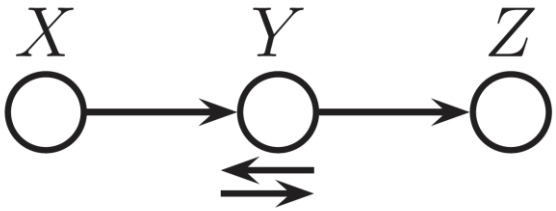
- The Bayes ball algorithm provides a simple way to check if A is d-separated from B given E. The idea is
  - We assume all nodes in E are observed.
  - We then place "balls" at each node in A and let then "bounce around" according to some rule.
  - We then ask if any of the balls reach any of the nodes in B.
- The three main rules to be followed are:
  - balls can travel opposite to edge directions
  - ball can pass through a chain or fork, but not if it is shaded in the middle
  - a ball cannot pass through a v-structure, unless it is shaded in the middle.

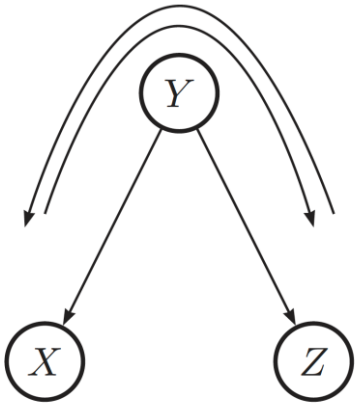
D-Separation & the Bayes Ball Algorithm

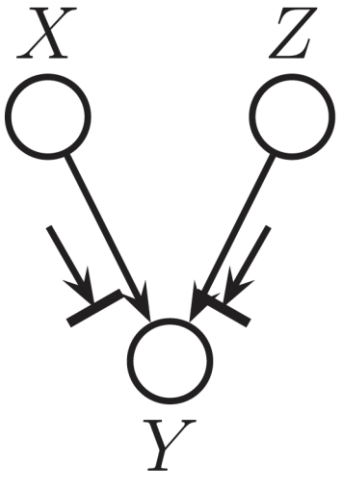












Bayes ball rules

D-Separation & the Bayes Ball Algorithm

- Consider a chain structure $X \rightarrow Y \rightarrow Z$, which encodes

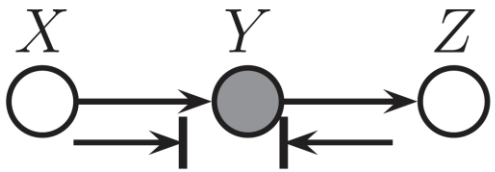
$$p(x, y, z) = p(x)p(y|x)p(z|y)$$

- Now if we condition on y

$$p(x, z|y) = \frac{p(x)p(y|x)p(z|y)}{p(y)} = \frac{p(x, y)p(z|y)}{p(y)} = p(x|y)p(z|y)$$

- Therefore,

$$x \perp z|y$$



D-Separation & the Bayes Ball Algorithm

- Consider a tent structure $X \leftarrow Y \rightarrow Z$, which encodes

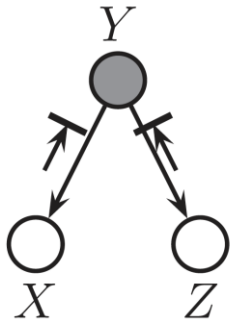
$$p(x, y, z) = p(y)p(x|y)p(z|y)$$

- Now if we condition on y

$$p(x, z|y) = \frac{p(y)p(x|y)p(z|y)}{p(y)} = p(x|y)p(z|y)$$

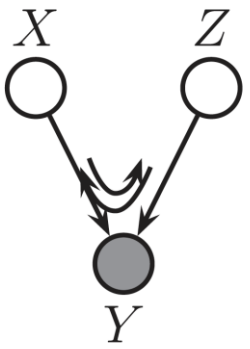
- Therefore,

$$x \perp z|y$$



D-Separation & the Bayes Ball Algorithm

- Consider a v structure $X \rightarrow Y \leftarrow Z$, which encodes



$$p(x, y, z) = p(x)p(z)p(y|x, z)$$

- Now if we condition on y

$$p(x, z|y) = \frac{p(x)p(z)p(y|x, z)}{p(y)} \neq p(x|y)p(z|y)$$

- Therefore, x and z are not conditionally independent given y. However,

$$p(x, z) = p(x)p(z)$$

- Hence,

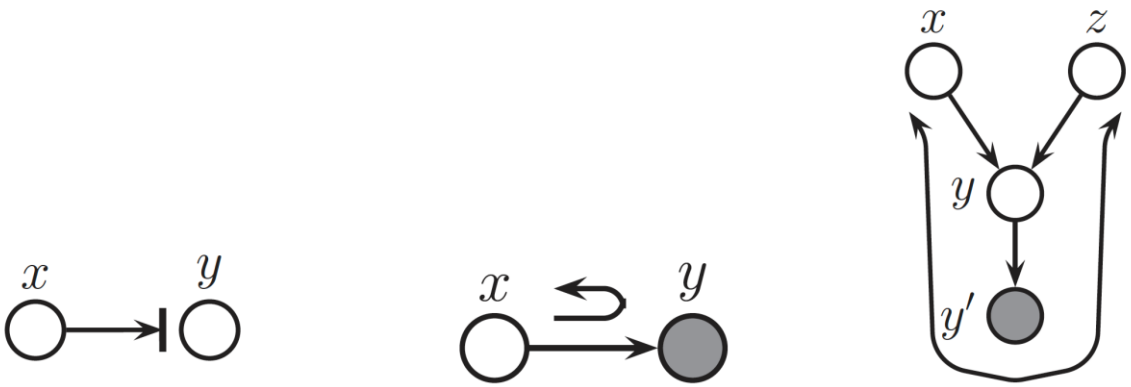
$$x \perp z$$

D-Separation & the Bayes Ball Algorithm

- We see that conditioning on a common child at the bottom of a v-structure makes its parents become dependent.
- This important effect is called explaining away, inter-causal reasoning , or Berkson's paradox
- Suppose we toss two coins, representing the binary numbers 0 and 1, and we observe the "sum" of their values
- A priori, the tosses are independent.
- But once we observe their sum, they become coupled
  - If the sum is 1, and the first coin is 0, then we know the second coin is 1

D-Separation & the Bayes Ball Algorithm

- Finally, Bayes Ball also needs the "boundary conditions"



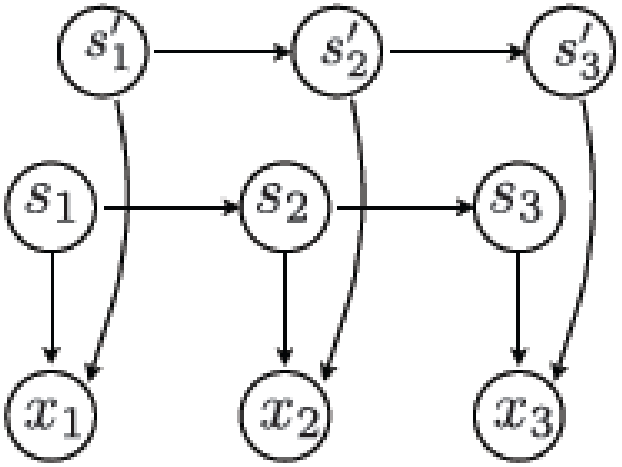
- Suppose, y' is a noise-free copy of y, which means we effectively observe y is we observe y'.
  - Parents x and z will compete to explain this.
  - So, if we send a ball $x \rightarrow y \rightarrow y'$, it should bounce back as $y' \rightarrow y \rightarrow z$.

Global Markov Properties of DAGs

- By chaining together local independencies, we can infer more global independencies.
- Definition: $X_1 - X_2 \cdots - X_n$ 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
- Definition: 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: $x_A \perp x_B | x_C$ if every variable in A is d-separated from every variable in B conditioned on all the variables in C.

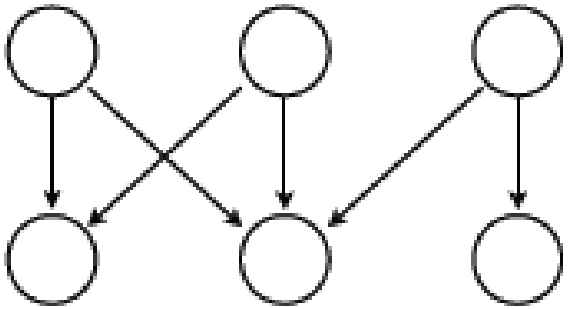
Marginal Independence and Induced Dependence

- Both *marginal independence and induced dependence is very common on factorial Hidden Markov Models where observations depend on two Markov chains running in parallel.*



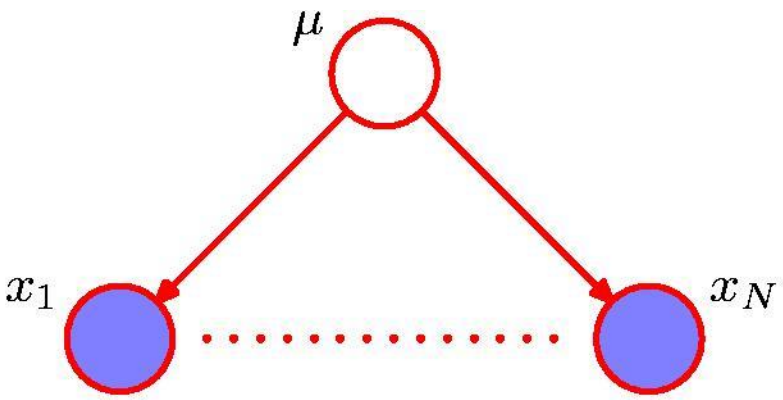
- Similarly in cases of 2 layer diagnostic models

Latent variables



Findings/test outcomes

D-Separation: I.I.D. Data



- Given μ, using d-separation, x_i is independent of any other $x_j, j \neq i$

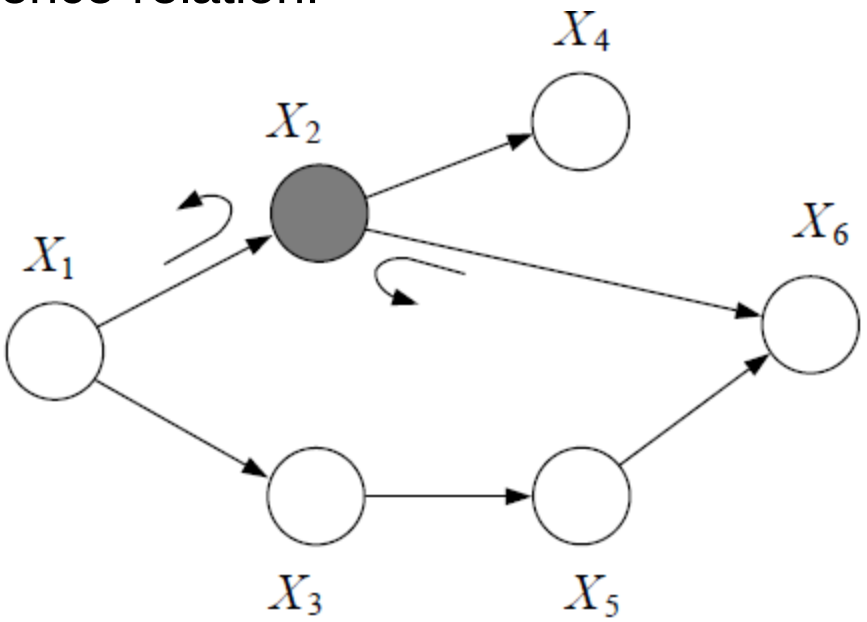
$$p(\mathcal{D}|\mu) = \prod_{n=1}^{N} p(x_n|\mu)$$

- However, if we integrate over μ, the observations are in general no longer independent

$$p(\mathcal{D}) = \int_{-\infty}^{\infty} p(\mathcal{D}|\mu)p(\mu)d\mu \neq \prod_{n=1}^{N} p(x_n)$$

The Bayes Ball Algorithm

- Using this algorithm, one can show for example the following independence relation:



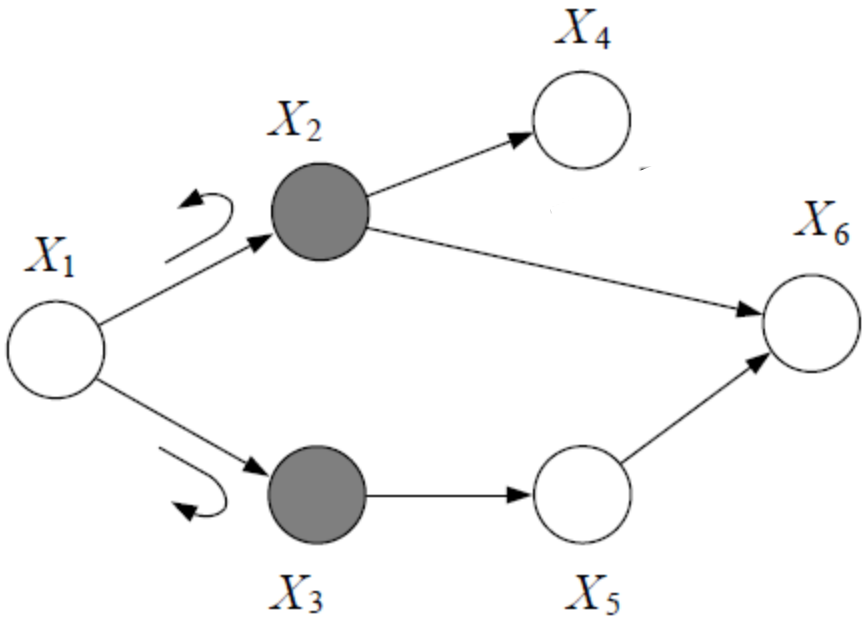
$$X_4 \perp\!\!\!\perp \{X_1, X_6\} \mid X_2$$

- Let us consider whether it is possible for a ball to arrive at node X_4 from either node X_1 or node X_6, given that X_2 is shaded.

- One path originates at X_1 and the other originates at X_6. In both cases, a ball arriving at X_2 would bounce back.

Reachability

- One can also show the following CI relation:



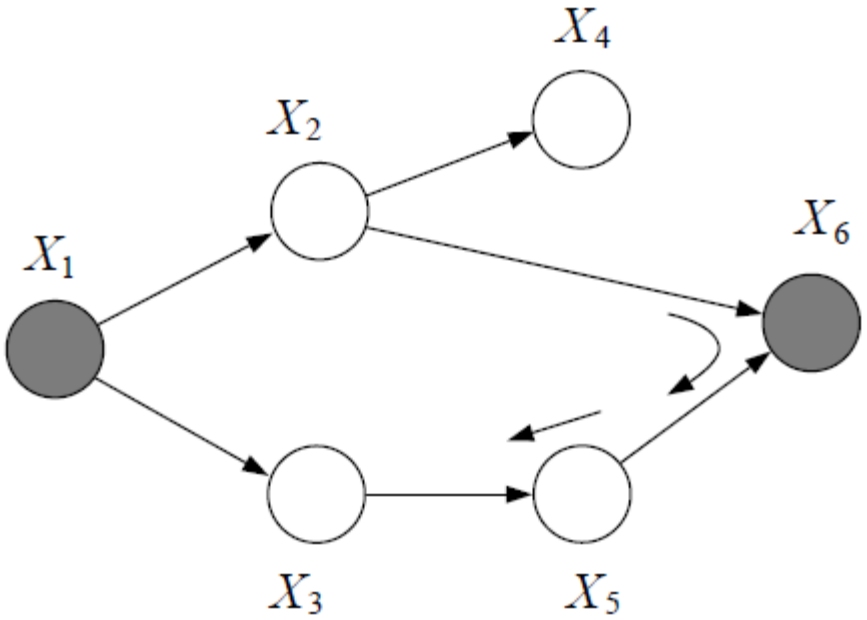
$$X_1 \perp\!\!\!\perp X_6 \mid \{X_2, X_3\}$$

- Consider a ball starting at X_1 and traveling to X_3. This ball cannot pass through to X_5.

- Similarly a ball cannot pass from X_1 to X_6 through X_2.

Reachability

- Using this algorithm, one can show the following reachability condition.



$$X_2 \not\perp X_3 \mid X_1, X_6$$

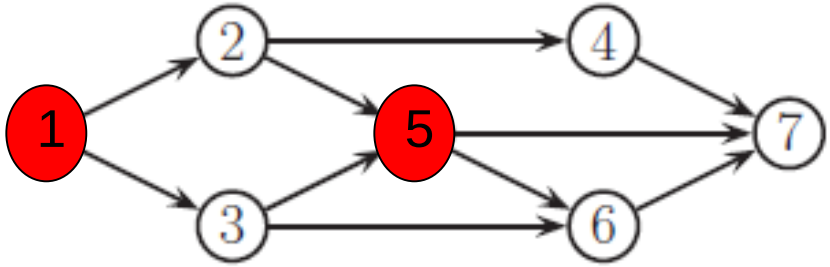
- A ball can pass from X_2 via X_6 to X_5. The ball then passes through X_5 and arrives at X_3. The observation of X_6 implies an *"explaining-away"* dependency between X_2 and X_5.

- *The ball from X_2 to X_1 is finally blocked at X_1 (tail to tail and observed)*

The Bayes Ball Algorithm

- In the DAG shown, we see that $x_2 \perp x_6 | \{x_5, x_1\}$ since
  - the $2 \rightarrow 5 \rightarrow 6$ path is blocked by x_5 (which is observed),
  - the $2 \rightarrow 4 \rightarrow 7 \rightarrow 6$ path is blocked by x_7 (which is hidden),
  - and the $2 \rightarrow 1 \rightarrow 3 \rightarrow 6$ path is blocked by x_1 (which is observed).

- However, we also see that $x_2 \not\perp x_6 | x_5, x_7$, since now the $2 \rightarrow 4 \rightarrow 7 \rightarrow 6$ path is no longer blocked by x_7 (which is observed).



D-Separation and Non-Reachability

- Traditionally, non-reachability of X_C from X_A given X_B is known as ***D-separation*** of X_A and X_C given X_B.
- D-separation requires blocking of *all* paths between *every* node in X_A and *every* node in X_C.
- The implied CI assertions must hold for *every member* of the family of distributions defined by *G*
- Some members of *G* may have additional CI relations.

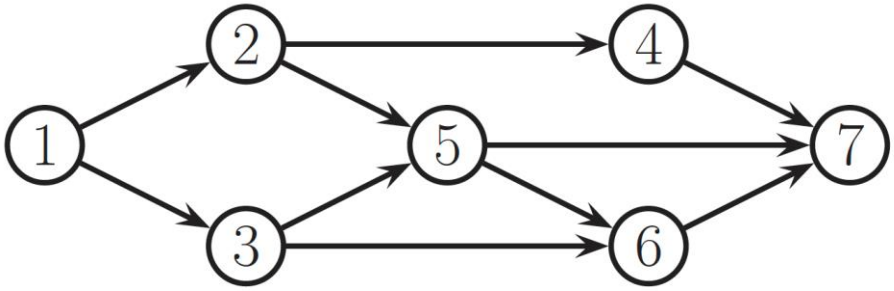
- J. Pearl, Probabilistic Reasoning in Intelligence Systems, Morgan Kaufmann, San Mateo, CA, 1988.

Other Markov Properties of DGMs

- Based on the d-separation criterion,

$$t \perp nd(t) \backslash \mathrm{pa}(t) | \mathrm{pa}(t)$$

where non-descendants of a node $nd(t)$ are all nodes except for its descendants, $nd(t) = \mathcal{V} \backslash \{t \cup desc(t)\}$



- For above graph, we have $nd(3) = \{2,4\}$ and $\mathrm{pa}(3) = 1$.

- Hence,

$$3 \perp 2,4 | 1$$

This is called the **directed local Markov property**

Other Markov Properties of DGMs

- A special case of this property is when we only look at predecessors of a node according to some topological ordering

$$t \perp pred(t) \backslash \mathrm{pa}(t) | \mathrm{pa}(t)$$

This is true because $pred(t) \subseteq nd(t)$.

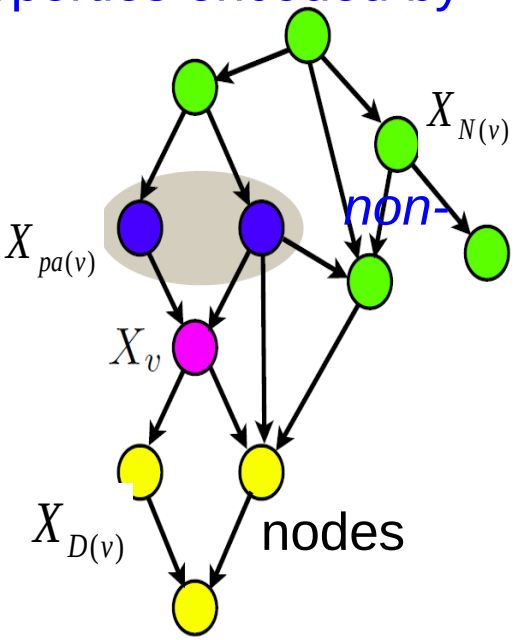
- This is called the ordered Markov property

- Note that, satisfying global Markov property implies satisfaction of local Markov property and topological Markov property. The vice-versa is also true.

Local Markov Property

- Definition: let $I_l(G)$ be the set of independence properties encoded by DAG G, namely the following local Markov property:

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



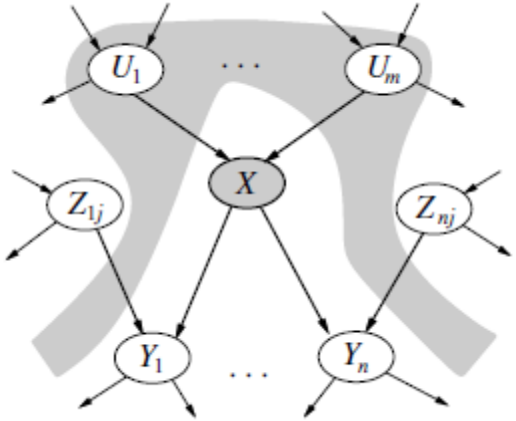
$$\{X_v \perp X_{N(v)}\} | X_{pa(v)}$$

where $X_{N(v)}$ are the (non-descendants) that appear earlier than X_v in I *not including the parents pa(v) of X_v*

- Here I is the ordering of the nodes. We define an ordering *I* of the nodes in a graph G to be topological if for every node X_v the parents of the node appear before X_v in *I*.

Local Markov Property

$\{X_v \perp X_{N(v)}\} | X_{pa(v)}$



- *Definition: X is an* ancestor *of Y in G=($\mathcal{X}$,$\mathcal{E}$), and Y is a* descendant *of X, if there exists a directed path $X_1,\ldots,X_k$ with $X_1 = X$ and $X_k = Y$.*
- *We use* Descendants_x *to denote X's descendants,* Ancestors_x *to denote X's ancestors.*
- NonDescendants_x *to denote the set of nodes in* $\mathcal{X}$- Descendants_x, i.e.
- Ancestors(X) ⊆ NonDescendants(X).

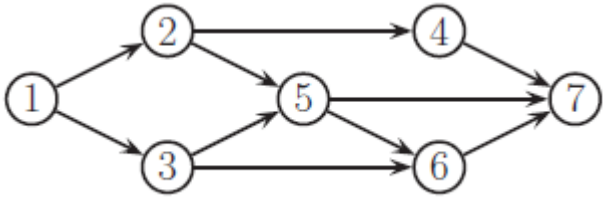
Other Markov Properties of DGMs

- From the d-separation criterion, one can conclude that

  Directed Local Markov property $t \perp nd(t) \setminus pa(t) \mid pa(t)$

  where the **non-descendants** of a node nd(*t*) are all the nodes except for its descendants, $nd(t) = V \setminus \{t \cup desc(t)\}$.

- E.g., in the Figure shown, we have nd(3) = {1,2,4}, and pa(3) = 1, so $3 \perp 2, 4 \mid 1$.



- A special case of this property is when we only *look at predecessors of a node according to some topological ordering*:

  Ordered Markov property $t \perp pred(t) \setminus pa(t) \mid pa(t)$

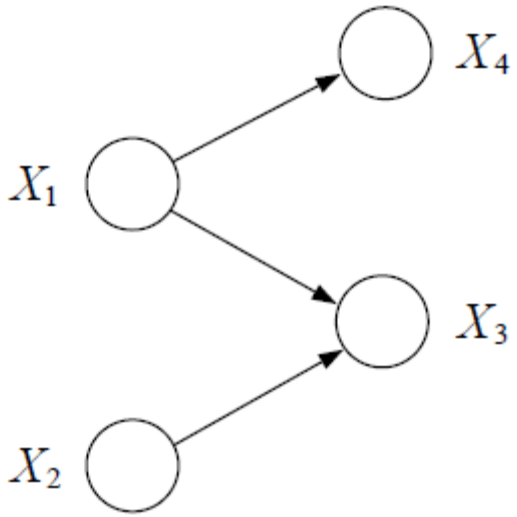
- This follows since pred(*t*) $\subseteq$ nd(*t*). E.g. in the Figure, with ordering 1, 2, . . . , 7. we find pred(3) = {1, 2} and pa(3) = 1, so $3 \perp 2 \mid 1$.

Characterization of Directed Graphs

- Consider $\mathcal{D}_1$ being the joint distribution of the directed graph (for any values of the conditional Tables)

$$p(x_1, \ldots, x_n) \triangleq \prod_{i=1}^{n} p(x_i \mid x_{\pi_i})$$

- *There is a family $\mathcal{D}_2$* of distributions associated with G that includes all $p(x_1,\ldots,x_n)$ that satisfy every CI relation associated with the graph.



$$X_1 \perp\!\!\!\perp X_2$$

$$X_2 \perp\!\!\!\perp X_4$$

$$X_2 \perp\!\!\!\perp X_4 \mid X_1$$

$$X_3 \perp\!\!\!\perp X_4 \mid X_1$$

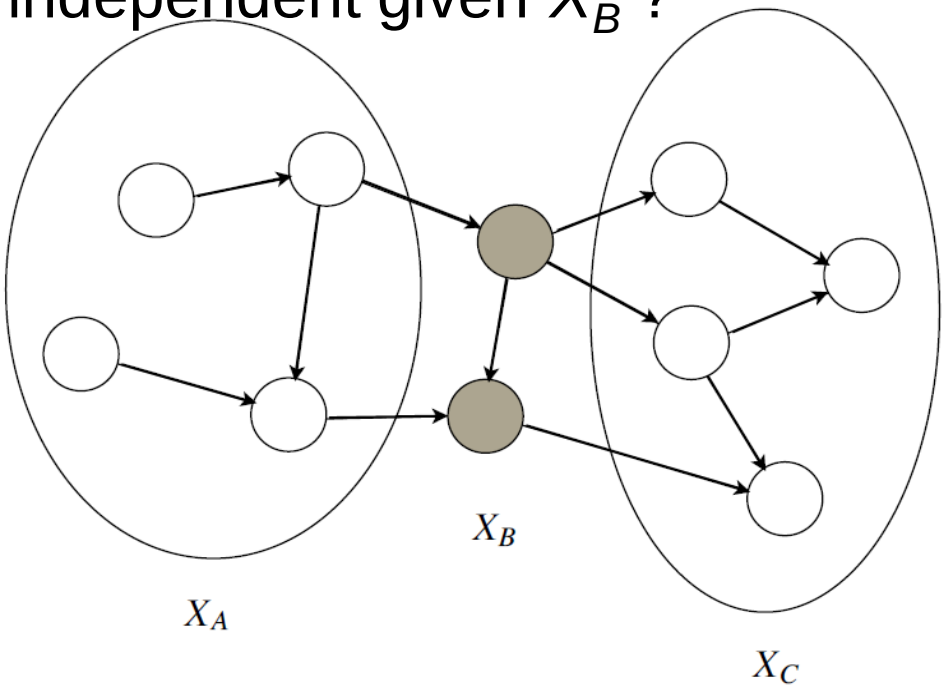
$$X_2 \perp\!\!\!\perp X_4 \mid \{X_1, X_3\}$$

$$\{X_2, X_3\} \perp\!\!\!\perp X_4 \mid X_1$$

- *Theorem: The two distributions $\mathcal{D}_1$ and $\mathcal{D}_2$ are identical.*

General Conditional Independence

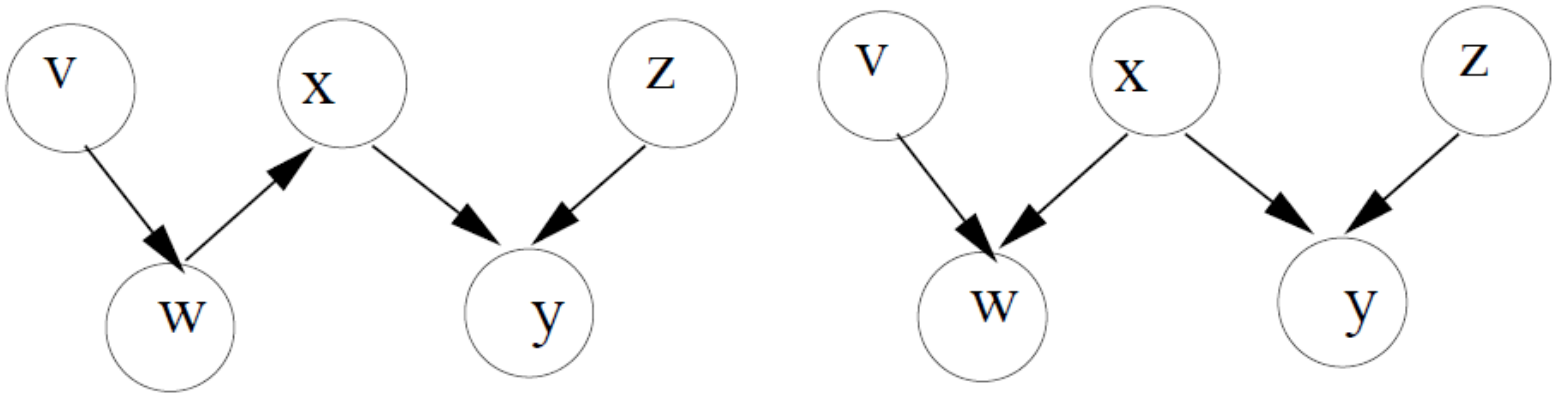
- For any given (nonoverlapping) sets of nodes X_A, X_B, and X_C, and a given graph (factorization) G, are X_A and X_C conditionally independent given X_B ?



- *What is the set **of all** CI assertions for graph G ?*
- *D-Separation* and the *Bayes Ball Algorithm* provide the answer.

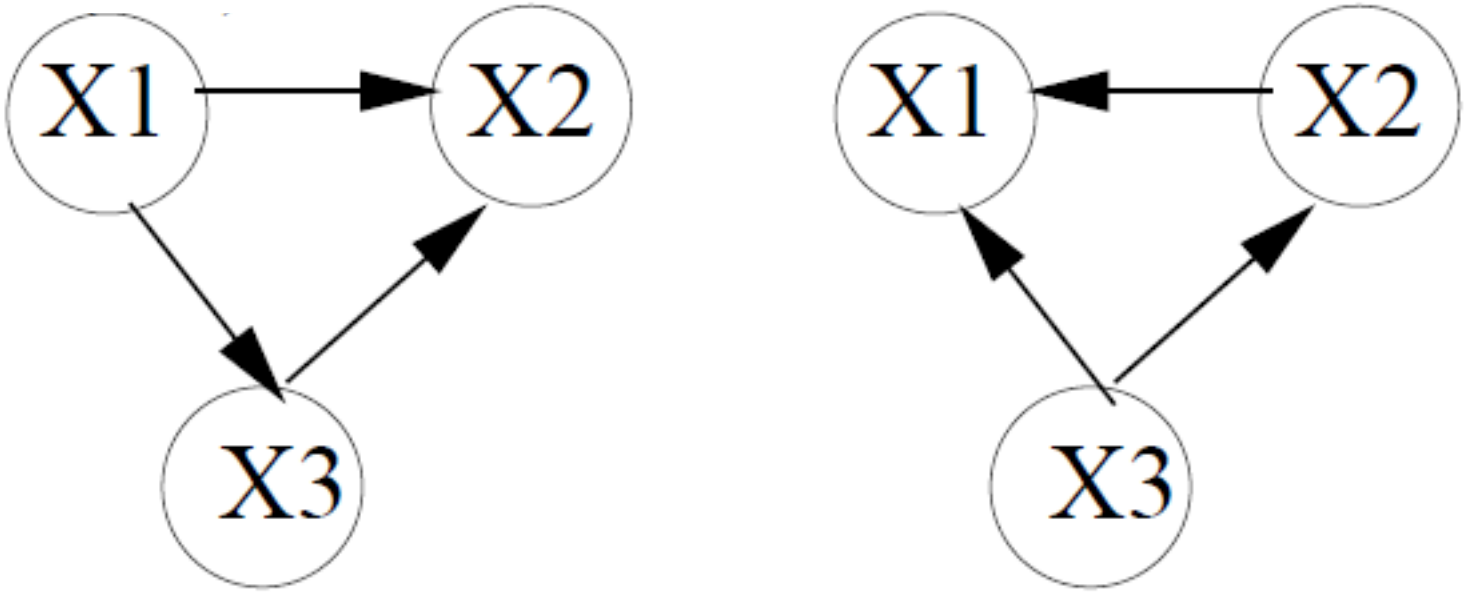
I-Equivalence

- We write $\mathbf{x}_A \perp_G \mathbf{x}_B | \mathbf{x}_C$ if A is independent of B given C in the graph G.
- Let $I(G)$ *be the set of all such CI statements encoded by the graph*.
- Definition: G_1 and G_2 are I-equivalent if $I(G_1) = I(G_2)$
- For example, $X \quad Y$ is I-equivalent to $X \quad Y$
- Theorem: *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, the two graphs do not necessarily have the same skeleton and v-structures.*
- *For example, the two graphs below have $I(G_1)=I(G_2)=0$.*



- *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.*
- This is important in the context of graph structure learning and causality.

Conditional Independence & I-map

- At the heart of any *directed* graphical model is a set of conditional independence (CI) assumptions

- We write $\boldsymbol{x}_A \perp_G \boldsymbol{x}_B | \boldsymbol{x}_C$ if A is independent of B given C in the graph G.

- Let $I(G)$ be the set of all such CI statements encoded by the graph

- We say $I(G)$ is an I-map for p if $I(G) \subseteq I(p)$, where $I(p)$ is the set of all conditional independent statements encoded by a graph.

  - This allows us to use G as a safe proxy for p.

Independence Properties of Distributions

- Definition: let I(P) be the set of independence properties of the form X ⊥ Y |Z that hold in distribution P.

- Consider a discrete distribution defined as follows:

X	Y	$p(X,Y)$
0	0	0.08
0	1	0.32
1	0	0.12
1	1	0.48

- Note from this Table that:

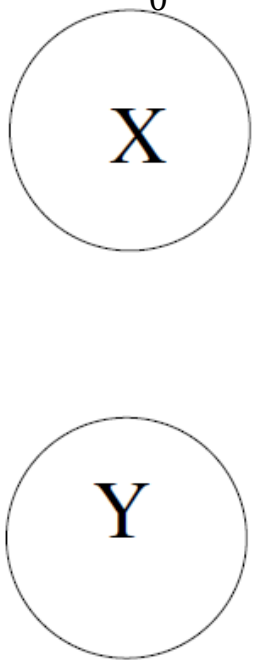
$$P(X = 1) = 0.48 + 0.12 = 0.6, \quad P(Y = 1) = 0.32 + 0.48 = 0.8$$
$$P(X = 1, Y = 1) = 0.48 = 0.6 \times 0.8 \Rightarrow P(X=x, Y=y) = P(X=x)P(Y=y) \ \forall x, y$$

$$\Rightarrow (X \perp Y) \in I(P) \quad \text{or } P \models (X \perp Y)$$

Local Independence Properties $I_\ell(G)$ of G

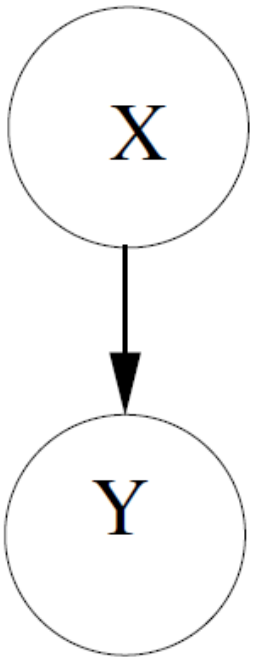
- Let $I_l(G)$ *be the set of all CI statements encoded by the graphs shown*. By inspection, we can write:

G_0



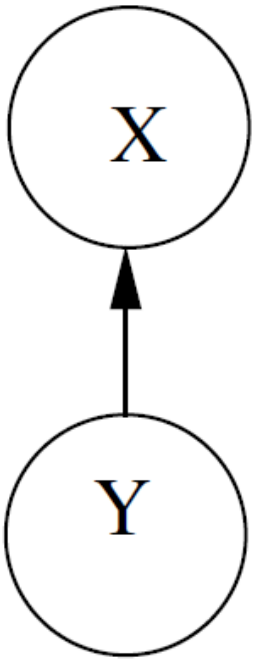
$$I_\ell(G_0) = \{(X \perp Y)\}$$

$G_{X \to Y}$



$$I_\ell(G_{X \to Y}) = 0$$

$G_{Y \to X}$



$$I_\ell(G_{Y \to X}) = 0$$

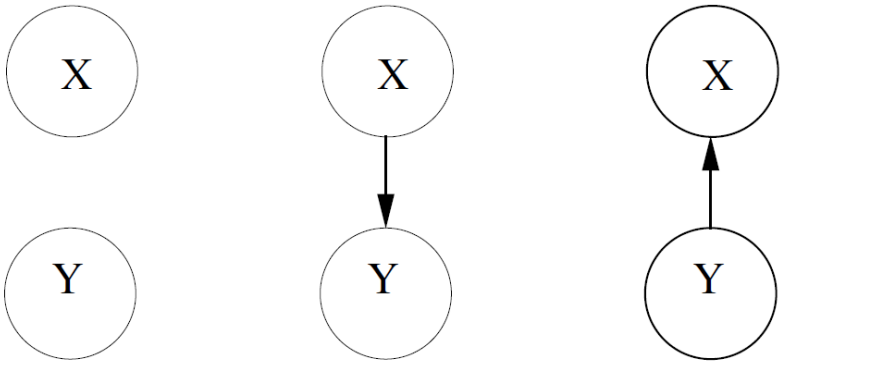
I-Maps

- Definition: A DAG G is an I-map (independency-map) of P if

$$I_l(G) \subseteq I(P)$$

i.e. P satisfies all local independencies associated with G. However, P may have additional independencies not reflected by G.

- From previous example



$I_\ell(G_0) = \{(X \perp Y)\}$ $I_\ell(G_{X \to Y}) = 0$ $I_\ell(G_{Y \to X}) = 0$

X	Y	$p(X,Y)$
0	0	0.08
0	1	0.32
1	0	0.12
1	1	0.48

I(P)={(X ⊥ Y)}

- Hence all three graphs are I-maps of P.

I-Maps

- Let *$I_I(G)$ be the set of all such CI statements encoded by the graph*.
- We say *that G is an **I-map** (independence map) for p, or **that p is Markov** wrt G, iff $I_I(G) \subseteq I(p)$,* where $I(p)$ is the set of all CI statements that hold for distribution p.
- In other words, *the graph is an I-map if it does not make any assertions of CI that are not true of the distribution.*
- This allows us to ***use the graph as a safe proxy for p when reasoning about p's CI properties*.**
- This is helpful for designing algorithms that work for large classes of distributions, regardless of their specific numerical parameters $\boldsymbol{\theta}$.

From I-MAP to Factorization

- Definition: P factorizes according to G if P can be written as

$$p(\boldsymbol{x}_{1:V} \mid G) = \prod_{t=1}^{V} p(x_t \mid \boldsymbol{x}_{pa(t)})$$

- Theorem: If G is an I-map of P, then P factorizes according to G.
  The proof can be seen as follows. Assume a topological ordering

$$p(\boldsymbol{X}_{1:V} \mid G) = p(X_1)\, p\left(X_2 \mid X_1\right) p(X_3 \mid X_1, X_2)\ldots$$

$$= \prod_{i=1}^{V} p(x_i \mid \boldsymbol{x}_{1:i-1})$$

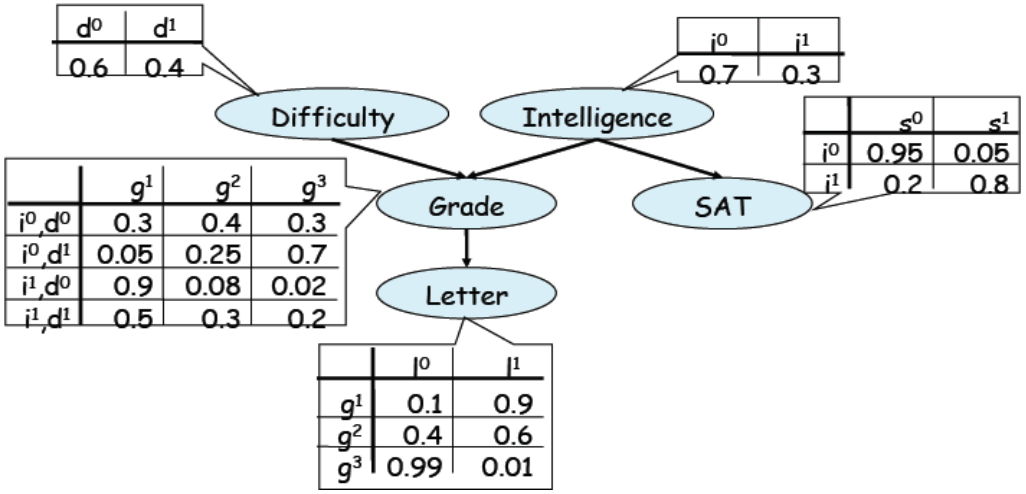
$$= \prod_{i=1}^{V} p\left(X_i \mid \boldsymbol{X}_{pa(i)}, \boldsymbol{Z}\right) = \prod_{i=1}^{V} p(X_i \mid \boldsymbol{X}_{pa(i)}) \left(where: \boldsymbol{Z} \subseteq \boldsymbol{X}_{NonDescendants(i)}\right)$$

where the last step follows since G is an I-Map of p.

Example

- In the graph shown below, the factorization of P is as follows:

$$p(D,I,G,S,L) = p(D)\,p(I)\,p(G \mid D,I)\,p(S \mid I)\,p(L \mid G)$$



$$L \perp I, D, S \mid G$$

$$S \perp D, G, L \mid I$$

$$G \perp S \mid I, D$$

$$I \perp D$$

$$D \perp I, S$$

- The number of independent parameters is now 1+1+8+2+3=15.
- Specification of the full joint requires 48 -1 = 47 parameters.
- In general, the number of parameters is less than V 2^k, where k=# parents (exponentially smaller than the number of parameters needed in the joint 2^V-1).

Compact Representation of the Joint Distribution

- Theorem: If G is an I-map of P, then P factorizes according to G.

$$p(\boldsymbol{x}_{1:V} \mid G) = \prod_{t=1}^{V} p(x_t \mid \boldsymbol{x}_{pa(t)})$$

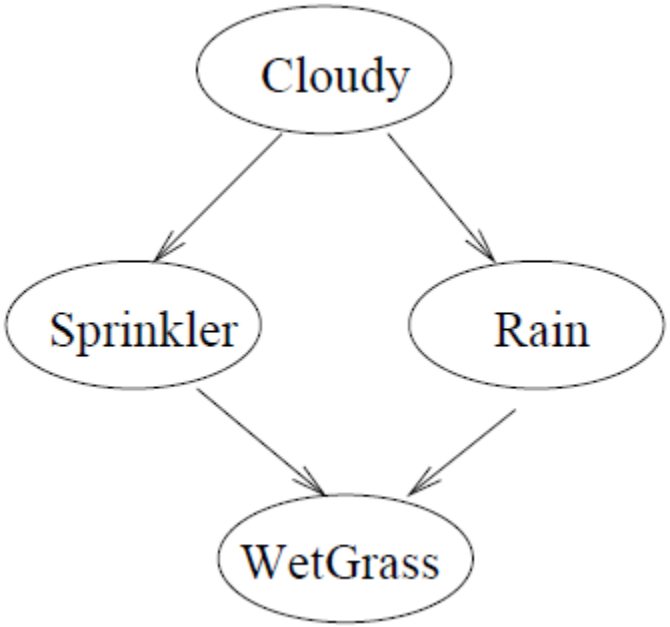
- Corollary: *If G is an I-map of P, then we can represent P using G and a set of conditional probability distributions (CPDs), $P(X_i|Pa(X_i))$, one per node.*

- Definition: *A Bayesian network (aka belief network) representing distribution P is an I-map of P and a set of CPDs.*

- For binary random variables, the Bayes net takes $\mathcal{O}(V2^K)$ parameters (K = max. num. parents), whereas full joint takes $\mathcal{O}(2^V)$ parameters.

- Factored representation is easier to understand, easier to learn and supports more efficient inference.

Compact Representation of the Joint Distribution



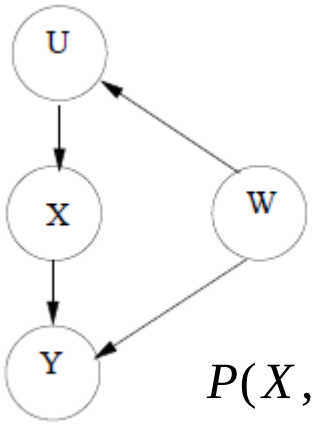
$$P(C, S,R,W) = P(C)P(S|C)P(R|C)P(W|S,R)$$

From Factorization to I-Map

- *Theorem: If P factorizes according to G, then G is an I-Map of P.*

- We show the proof with an example. Consider a distribution P that factorizes according to G as:

$$P(X,W,U,Y) = p(W)\,p(U \mid W)\,p(Y \mid X,W)\,p(X \mid U)$$



- From this factorization we can derive

$$X \perp W \mid U$$

$$P(X,W \mid U) = \frac{\sum_Y P(X,W,U,Y)}{P(U)} = \frac{p(W)\,p(U \mid W)\,p(X \mid U)\sum_Y p(Y \mid X,W)}{P(U)}$$

$$= \frac{p(U,W)\,p(X \mid U)}{p(U)} \Rightarrow P(X,W \mid U) = P(X \mid U)P(W \mid U)$$

Minimal I-Maps and Bayesian Nets

- Let G be a fully connected DAG. Then $I_l(G) = \emptyset \subseteq I(P)$ for any P.
- Hence the complete graph is an I-map for any distribution.
- The fully connected graph is an I-map of all distributions, since it makes no CI assertions at all (since it is not missing any edges).
- Definition: *A DAG G is a minimal I-map for P if it is an I-map for P, and if the removal of even a single edge from G renders it not an I-map.*
- We therefore say G is a **minimal I-map** of p if G is an I-map of p, and if there is no G '$\subseteq$ G which is an I-map of p.

Minimal I-Maps and Bayesian Nets

- Construction: pick a node ordering, then let the parents of node X_i be the minimal subset of $U \subseteq \{X_1, \ldots, X_{i-1}\}$ such that:

$$X_i \perp \{X_1, \ldots, X_{i-1}\} \setminus U | U.$$

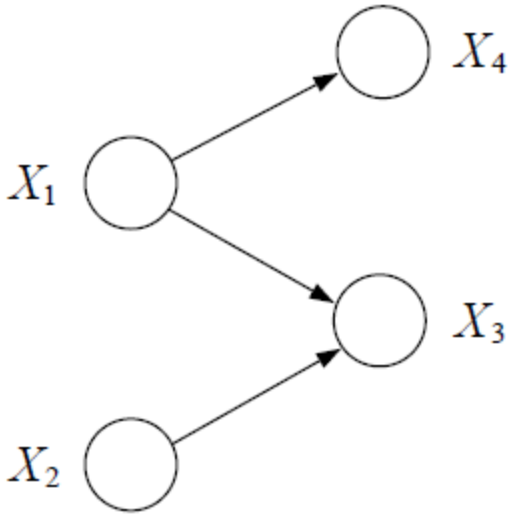
- Updated Definition: *A Bayesian network (aka belief network) representing distribution P is a minimal I-map of P and a set of CPDs.*

Conditional Independence Properties of DGMs

- A fully connected graph is an I-map of all distributions, since it makes no CI assertions at all
- We therefore say G is a minimal I-map of p is there is no $G' \subset G$ which is an I-map of p.
- But how to determine if $\boldsymbol{x}_A \perp_G \boldsymbol{x}_B | \boldsymbol{x}_C$?
  - For undirected graph, determining unconditional independencies is straightforward based on simple graph separation.
  - However for directed graphical model, we need to take into account the directions of the edges as well (explaining away)

Characterization of Directed Graphs

- Consider two distributions: One $\mathcal{D}_1$ being the joint distribution of the directed graph (for any values of the conditional Tables) and the other $\mathcal{D}_2$ defined from all the independent relations between the random variables in the graph.



$$p(x_1, \ldots, x_n) \triangleq \prod_{i=1}^{n} p(x_i \mid x_{\pi_i})$$

$$X_1 \perp\!\!\!\perp X_2$$

$$X_2 \perp\!\!\!\perp X_4$$

$$X_2 \perp\!\!\!\perp X_4 \mid X_1$$

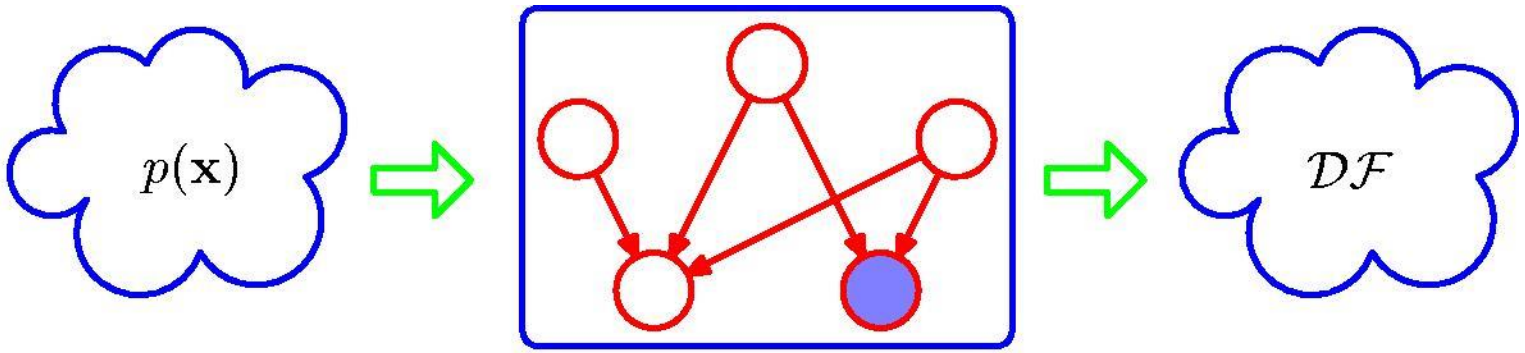
$$X_3 \perp\!\!\!\perp X_4 \mid X_1$$

$$X_2 \perp\!\!\!\perp X_4 \mid \{X_1, X_3\}$$

$$\{X_2, X_3\} \perp\!\!\!\perp X_4 \mid X_1$$

- *Theorem: The two distributions $\mathcal{D}_1$ and $\mathcal{D}_2$ are identical.*

Directed Graphs as Distribution Filters

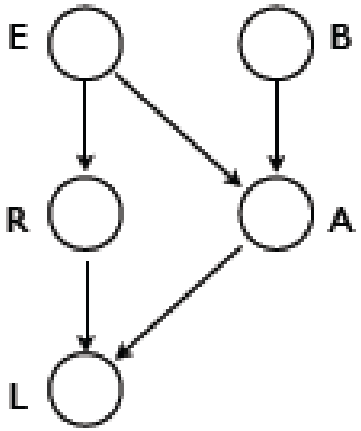


We can view a graphical model (in this case a directed graph) as a filter in which *a probability distribution p(x) is allowed through the filter if, and only if, it satisfies the directed factorization property.* The set of all possible probability distributions *p(x)* that pass through the filter is denoted $\mathcal{DF}$.

Note that for any given graph, *the set of distributions $\mathcal{DF}$ will include any distributions that have additional independence properties beyond those described by the graph.*

Directed Graphs and Distributions

- The probability distribution associated with the graph needs to be consistent with all the independence relations encoded in the graph.



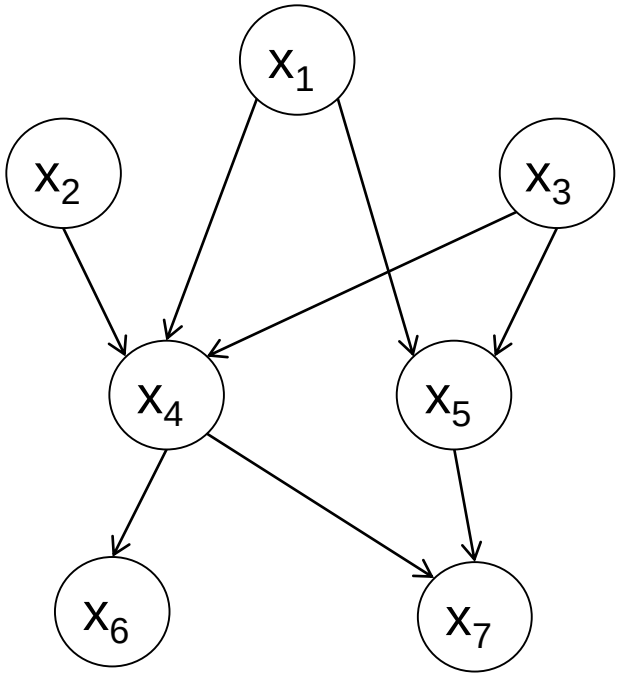
$$p(A,B,E,R,L) = p(E)\,p(B)\,p(R\,|\,E)\,p(A\,|\,B,E)\,p(L\,|\,A,R)$$

- However, a distribution that is consistent with the graph may satisfy additional independence properties not encoded in the graph, e.g.

$$p(A,B,E,R,L) = p(E)\,p(B)\,p(R\,|\,E)\,p(A\,|\,E)\,p(L\,|\,R)$$

$$p(A,B,E,R,L) = p(E)\,p(B)\,p(R\,|\,E)\,p(A)\,p(L)$$

DAGs and Probability Distributions



$$p(x_1,\ldots,x_7) = p(x_1)\,p(x_2)\,p(x_3)\,p(x_4|x_1,x_2,x_3)$$
$$p(x_5|x_1,x_3)\,p(x_6|x_4)\,p(x_7|x_4,x_5)$$

General Factorization

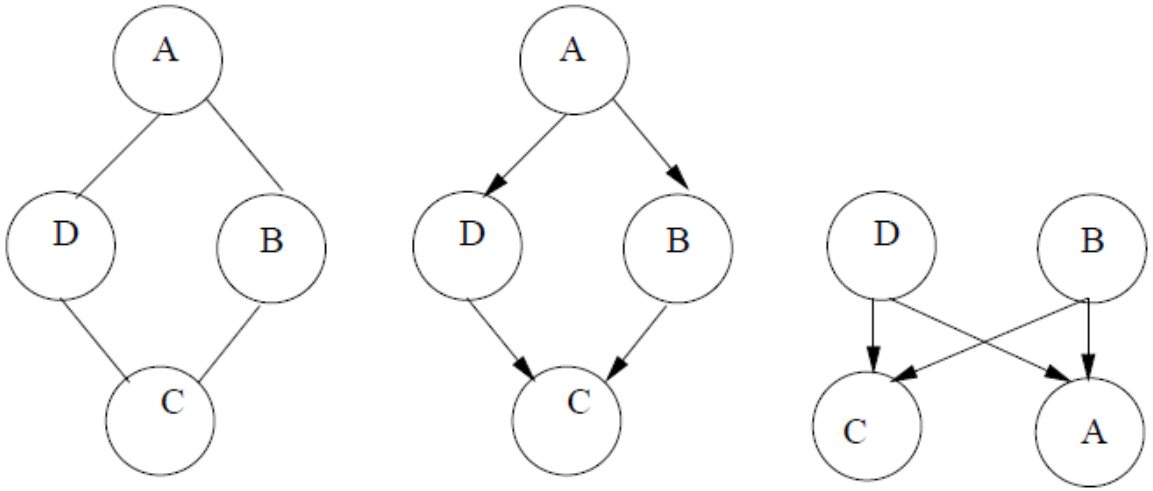
$$p(\mathbf{x}) = \prod_{k=1}^{K} p(x_k|x_{Pa_k})$$

$$x_{Pa_k} = \{x_j,\ j \in Pa_k\}$$

Any distribution $p(x_1,x_2,..,x_K)$ consistent with all the independence statements implied by a DAG via D-separation can be written as a product of local conditional distributions of a variable given its parents

P-Map (Perfect-Map)

- Can we find a graph that captures all the independencies in an arbitrary distribution (and no more)?
- Defn: A DAG G is a perfect map (P-map) for a distribution P if $I(P) = I(G)$.
- Thm: *not every distribution has a perfect map.*
- Proof by counterexample. Suppose we have a model where A⊥C|{B,D}, and B ⊥ D|{A,C}. This cannot be represented by any Bayes net.
- In the example, BN1 wrongly says B ⊥ D|A, BN2 wrongly says B ⊥ D.



Global Markov Properties of DAGS

- By chaining together local independencies, we can infer more global independencies.

- Defn: X is d-separated (directed-separated) from Y given Z if along every undirected path between X and Y there is a node w s.t. either

  - W has converging arrows ($\rightarrow w \leftarrow$) and neither W nor its descendants are in z; or

  - W does not have converging arrows and $W \in Z$.

- Definition: *I(G) = all independence properties that correspond to d-separation*

$$I(G) = \{(X \perp Y \mid Z) : d - sep_G(X; Y \mid Z)\}$$

Soundness of d-Separation

- Theorem: If P factorizes according to G, then I(G) ⊆ I(P).

- This means that any independence claim made by the graph is satisfied by all distributions P that factorize according to G (no false claims of independence).

- The proof of the theorem is easier highlighted for undirected graphs where d-separation is a simple graph separation (see Koller and Friedman, Chapter 4).

Completeness of d-Separation

- Theorem (Completeness) v1: For any distribution P that factorizes over G, if $(X \perp Y \mid Z) \in I(P)$, then $dsep_G(X; Y \mid Z)$.
- Contrapositive rule: $(A \Rightarrow B) \Leftrightarrow (\neg B \Rightarrow \neg A)$.
- Theorem (Completeness, contrapositive form) v1. If X and Y are not d-separated given Z, then X and Y are dependent in all distributions P that factorize over G.
- This definition of completeness is too strong since P may have conditional independencies that are not evident from the graph.
- eg. Let G be the graph $X \rightarrow Y$, where P(Y |X) is

X	$Y=0$	$Y=1$
0	0.4	0.6
1	0.4	0.6

- G is I-map of P since $I(G) = \emptyset \subseteq I(P) = \{(X \perp Y)\}$.
- But the CPD encodes $X \perp Y$ which is not evident in the graph.

Completeness of d-Separation

- Theorem (Completeness) v2*: If $(X \perp Y \mid Z)$ in **all** distributions P that factorize over G, then $dsep_G(X; Y \mid Z)$.*

- Theorem (Completeness, contrapositive form) v2: *If X and Y are not d-separated given Z, then X and Y are dependent in some distribution P that factorizes over G.*

- Theorem: *d-separation is complete.*

- Proof: See Koller & Friedman, Theorem 3.5, p73.

- Hence d-separation captures as many of the independencies as possible (without reference to the particular CPDs) for all distributions that factorize over some DAG.

D-Separation ⇔ Factorization

- Consider a DAG G with nodes (variables) $X_1, \ldots, X_V$.
- Consider the set $\mathcal{U}$ of all (families of) joint distributions for the same variables
- A subset of distributions, $\mathcal{DI} \subseteq \mathcal{U}$, maintain the CI assertions implied by D-separation in G
- Another subset of distributions, $\mathcal{DF} \subseteq \mathcal{U}$, can be factored according to G
- It turns out that $\mathcal{DI} = \mathcal{DF}$

Summary of Markov Properties of DGMs

- We have now described three Markov properties for DAGs:

G: Directed Global Markov Property $\mathbf{x}_A \perp_G \mathbf{x}_B | \mathbf{x}_E \iff$ A is d-separated from B given E

L: Directed Local Markov Property $t \perp \text{nd}(t) \setminus \text{pa}(t) \mid \text{pa}(t)$

O: Ordered Markov Property $t \perp \text{pred}(t) \setminus \text{pa}(t) \mid \text{pa}(t)$

- It is obvious that $G \Rightarrow L \Rightarrow O$.
- What is less obvious, but nevertheless true, is that $O \Rightarrow L \Rightarrow G$. Hence all *these properties are equivalent*.
- Furthermore, any distribution p that is Markov wrt G can be factorized as;

F: Factorization Property

$$p(\mathbf{x}_{1:V} \mid G) = \prod_{t=1}^{V} p(x_t \mid \mathbf{x}_{pa(t)})$$

- Clearly $O \Rightarrow F$, but one can show that the converse also holds.

- Koller, D. and N. Friedman (2009). *Probabilistic Graphical Models: Principles and Techniques*. MIT Press.