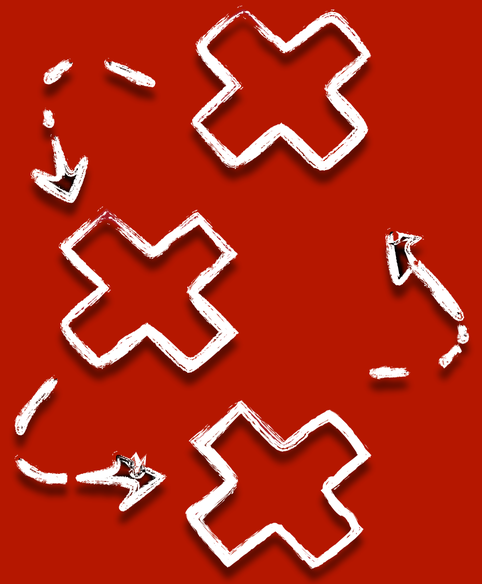


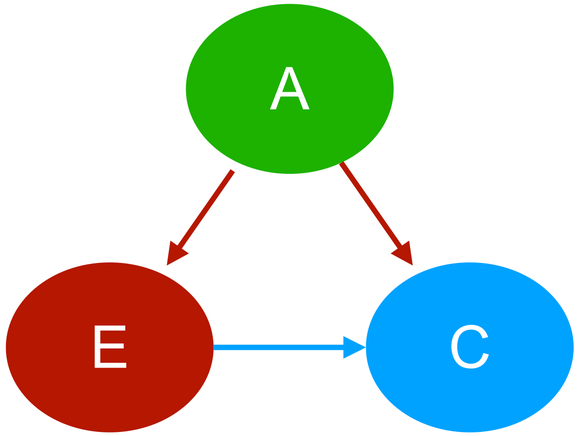
Deep learning 2: Causality & DL

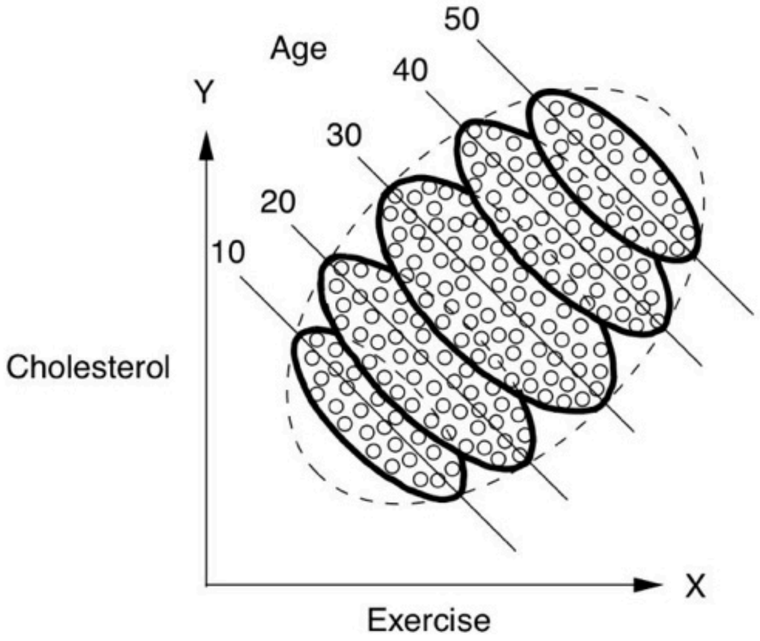
1.2: Graphical models

Lecturer: Sara Magliacane

UvA - Spring 2022

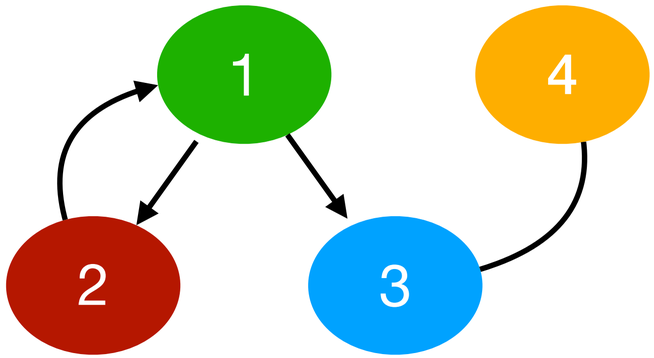
Graphical models





Graph terminology

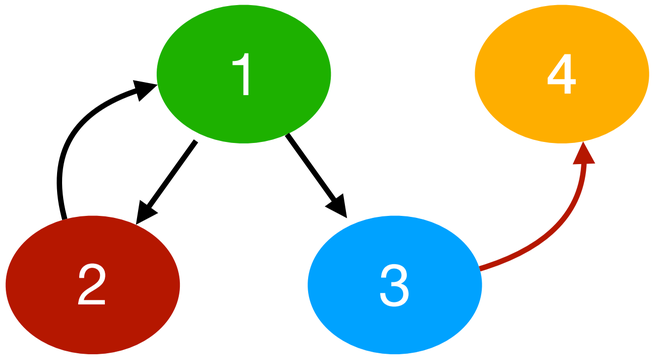
- A graph G is a tuple $G = (\mathbf{V}, \mathbf{E})$:
  - $\mathbf{V}$ is the set of **nodes** (vertices)
  - $\mathbf{E}$ is the set of **edges** between two nodes, i.e. $\mathbf{E} \subseteq \mathbf{V} \times \mathbf{V}$



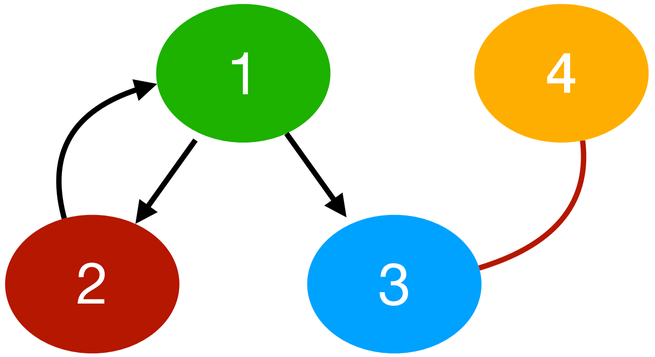
  - Two nodes connected by an edge are **adjacent**

Directed, mixed, undirected graphs

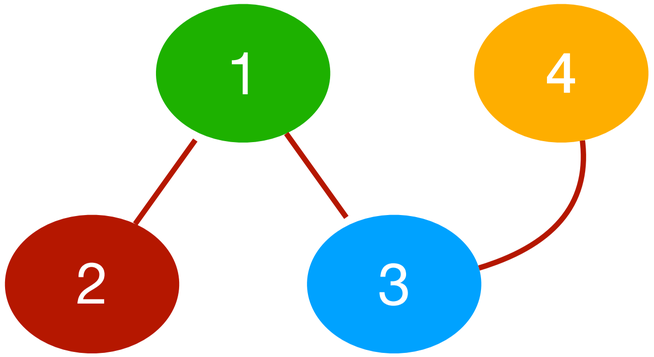
- A graph G is a tuple $G = (\mathbf{V}, \mathbf{E})$:
  - $\mathbf{V}$ is the set of **nodes** (vertices)
  - $\mathbf{E}$ is the set of **edges** between two nodes, i.e. $\mathbf{E} \subseteq \mathbf{V} \times \mathbf{V}$
  - If all edges are **directed** $\rightarrow$ then the graph is **directed**



Directed graph



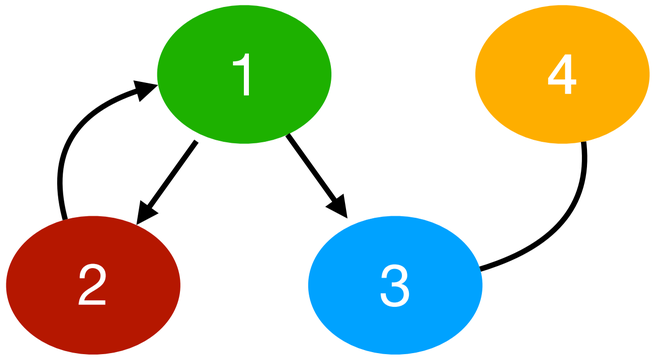
Mixed graph

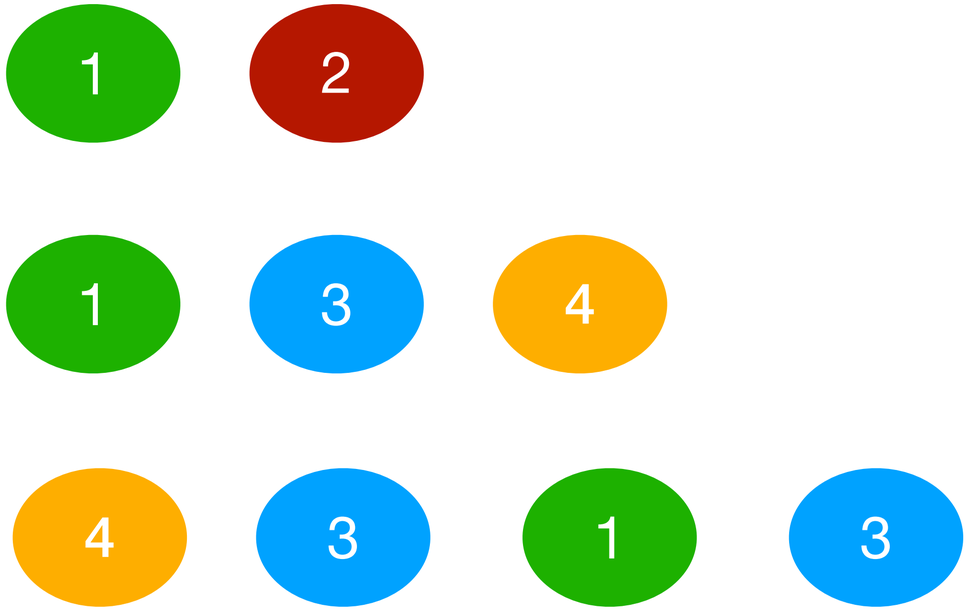


Undirected graph

Graph terminology: paths

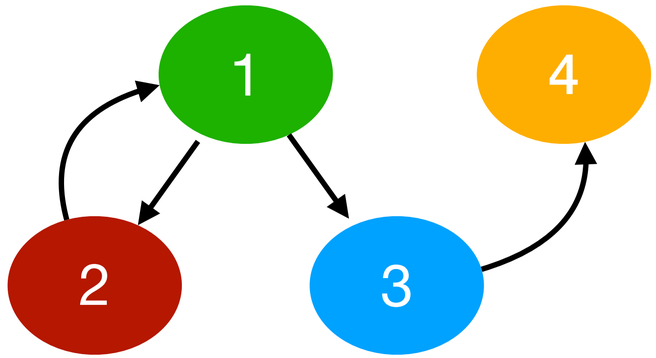
- A **path** between **node i and node j** is a sequence of **distinct nodes** $(i, \ldots, j)$ such that each two **consecutive nodes** are **adjacent**





Directed graphs: paths vs directed paths

- A **path** between **node i and node j** is a sequence of **distinct nodes** $(i, \ldots, j)$ such that each two **consecutive nodes** are **adjacent**
- A **directed path** between **node i and node j** is a path where **all edges point towards j**, i.e. $i \rightarrow \ldots \rightarrow j$

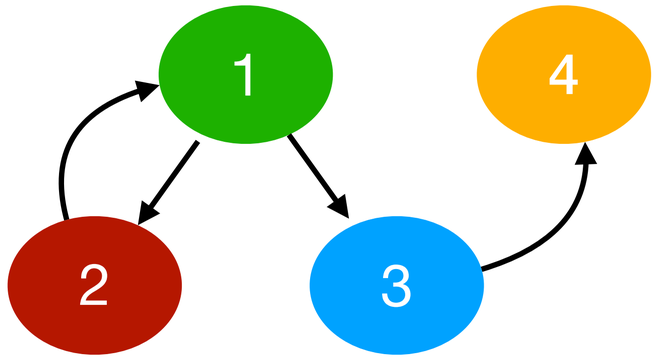


Directed path from 1 to 4

A path from 4 to 1, but not a directed path

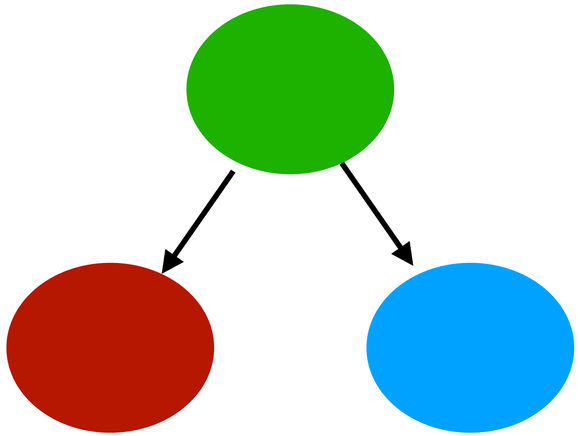
Directed graphs: paths vs directed paths

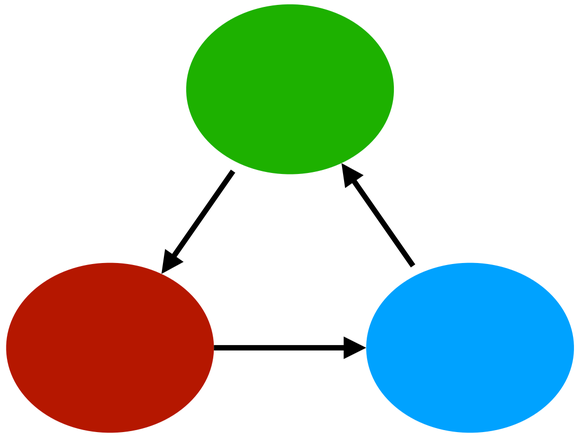
- A **path** between **node i and node j** is a sequence of **distinct nodes** $(i, \ldots, j)$ such that each two **consecutive nodes** are **adjacent**
- A **directed path** between **node i and node j** is a path where **all edges point towards j**, i.e. $i \rightarrow \ldots \rightarrow j$
- A **directed cycle** is a directed path $(i, \ldots, j)$ + (j, i)



Directed Acyclic Graphs (DAGs)

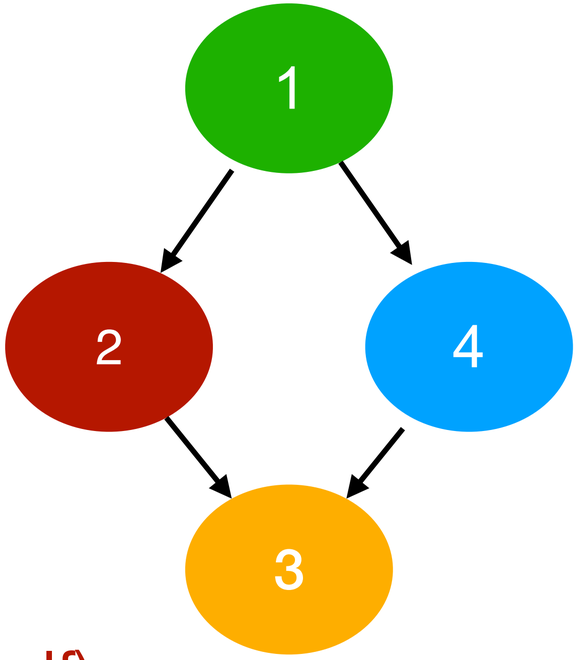
- A DAG is a directed graph $G = (V, E)$:
  - V is the set of **nodes** (vertices)
  - E is the set of **directed edges** between the nodes
  - There are **no directed cycles**





Relationships between nodes in a DAG

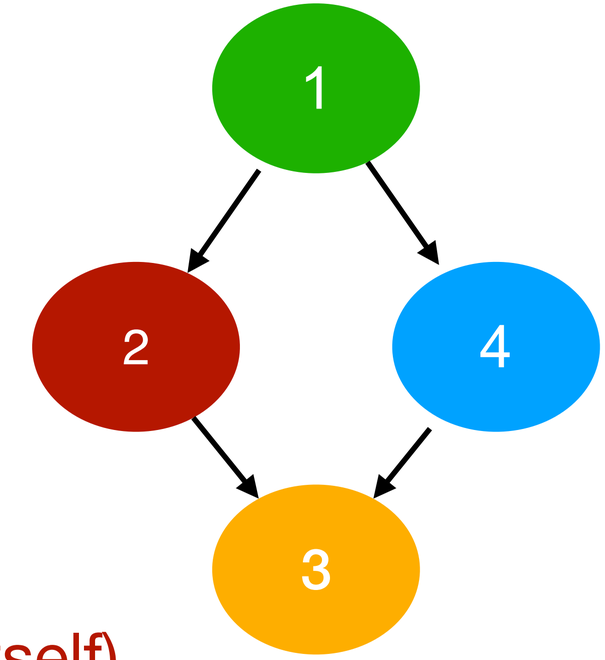
- **Parents** of a node $\mathrm{Pa}_G(V)$
  - Nodes that have an edge pointing to V
- **Children** of a node $\mathrm{Ch}_G(V)$
  - Nodes that have an edge pointing from V
- **Ancestors** of a node $\mathrm{An}_G(V)$
  - Nodes that have a **directed path** to V (including V itself)
- **Descendants** of a node $\mathrm{Desc}_G(V)$
  - Nodes that are reached from V via **directed paths** (including V itself)



Relationships between nodes in a DAG

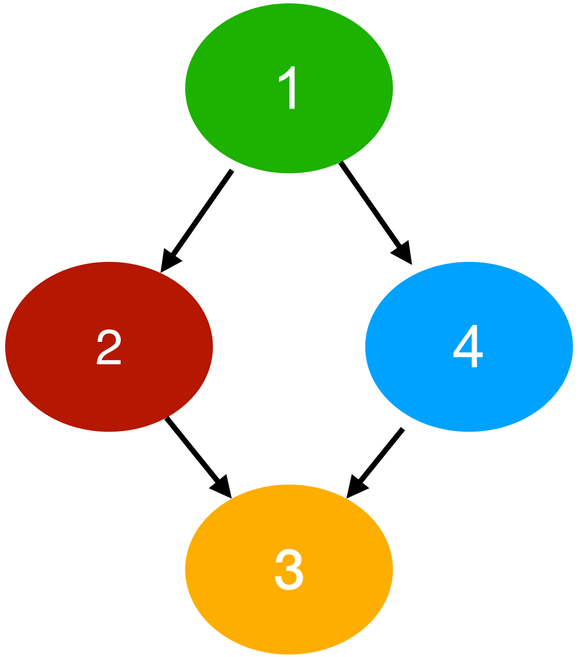
[We omit G, when it is clear from the context]

- **Parents** of a node $\mathrm{Pa}(V)$
  - Nodes that have an edge pointing to V
- **Children** of a node $\mathrm{Ch}(V)$
  - Nodes that have an edge pointing from V
- **Ancestors** of a node $\mathrm{An}(V)$
  - Nodes that have a **directed path** to V (including V itself)
- **Descendants** of a node $\mathrm{Desc}(V)$
  - Nodes that are reached from V via **directed paths** (including V itself)



Kinship relationships for sets of nodes

- We will use **bold** for sets (including sets of nodes)



- **Parents** of a set of nodes $\mathbf{A} \subseteq \mathbf{V}$:

$$\text{Pa}(\mathbf{A}) := \bigcup_{V \in \mathbf{A}} \text{Pa}(V)$$

$$\text{Pa}(\{2,3\}) = \text{Pa}(2) \cup \text{Pa}(3) = \{1\} \cup \{2,4\} = \{1,2,4\}$$

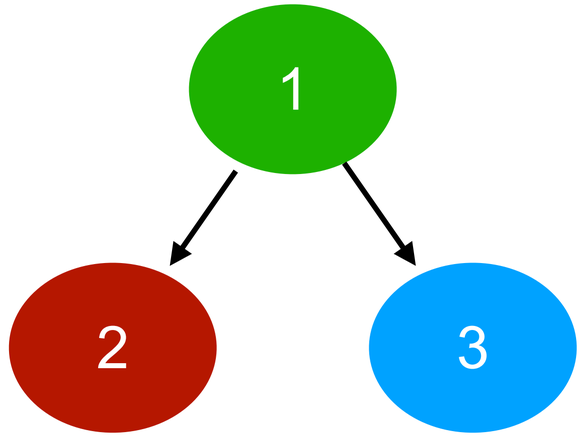
- Similarly for children, ancestors and descendants

DAGs and random variables

- We can represent a factorisation of a joint distribution $p(X_1, \ldots, X_p)$ as a **DAG**
- **Each node $i \in \mathbf{V}$** represents a **random variable X_i**
  - For $\mathbf{A} \subseteq \mathbf{V}$, we can define $X_{\mathbf{A}} := \{X_i : i \in \mathbf{A}\}$

- **Edges** are related to the factors

$$p(X_1, X_2, X_3)$$



Factorizing joint distributions

- A joint distribution can always be factorized in several ways by iterating the **chain rule**

$$p(X_1, X_2) = p(X_1)p(X_2 \mid X_1) = p(X_2)p(X_1 \mid X_2)$$

Factorizing joint distributions

- A joint distribution can always be factorized in several ways by iterating the **chain rule**

$$p(X_1, X_2) = p(X_1)p(X_2 \mid X_1) = p(X_2)p(X_1 \mid X_2)$$

- In general, given any **ordering** of the variables $(X_1, \ldots, X_p)$, we can write:

$$p(X_1, \ldots, X_p) = p(X_1)p(X_2 \mid X_1)\ldots p(X_p \mid X_1, \ldots, X_{p-1})$$

Factorizing joint distributions

- Given any **ordering** of the variables $(X_1, \ldots, X_p)$ we can write:

$$p(X_1, \ldots, X_p) = p(X_1)p(X_2 \mid X_1) \ldots p(X_p \mid X_1, \ldots, X_{p-1})$$

- For example $p(X, Y, Z)$ can be equivalently factorized as:
  - $p(X, Y, Z) =$
  - $p(X, Z, Y) =$
  - $p(Z, Y, X) =$
  - ...

Exploiting conditional independences

- Given any **ordering** of the variables $(X_1, \ldots, X_p)$ we can write:

$$p(X_1, X_2) = p(X_1)p(X_2 \,|\, X_1) = p(X_2)p(X_1 \,|\, X_2)$$

- We can **simplify** the factorisation by using **conditional independences:**

$$X_i \perp\!\!\!\perp X_j \,|\, X_{\mathbf{Z}} \implies p(X_i \,|\, X_j, X_{\mathbf{Z}}) = p(X_i \,|\, X_{\mathbf{Z}})$$

(special case $X_i \perp\!\!\!\perp X_j \implies p(X_i \,|\, X_j) = p(X_i)$)

Quick recap: Independent random variables (discrete case)

- **Definition:** Two discrete random variables X and Y are **independent** iff:

$$\forall x, y : P(X = x, Y = y) = P(X = x)P(Y = y)$$

- We then write $X \perp\!\!\!\perp Y$ (equiv. $Y \perp\!\!\!\perp X$), otherwise $X \not\perp\!\!\!\perp Y$
- This is equivalent to $P(X = x \,|\, Y = y) = P(X = x)$ (and vice versa for Y)
- Intuitively, this means that knowing the value of Y **will not tell us anything** about the distribution of X.

Quick recap: conditional independence (discrete case)

- X is independent of Y **conditioned/given Z** (possibly a set) iff

$$\forall x, y, \mathbf{z} : P(X = x, Y = y \,|\, \mathbf{Z} = \mathbf{z}) = P(X = x \,|\, \mathbf{Z} = \mathbf{z}) P(Y = x \,|\, \mathbf{Z} = \mathbf{z})$$

$$(\text{for } P(\mathbf{Z} = \mathbf{z}) > 0)$$

- We then write $X \perp\!\!\!\perp Y \,|\, Z$ (equiv. $Y \perp\!\!\!\perp X \,|\, Z$), otherwise $X \not\perp\!\!\!\perp Y \,|\, Z$

- This is equivalent to $P(X = x \,|\, Y = y, \mathbf{Z} = \mathbf{z}) = P(X = x \,|\, \mathbf{Z} = \mathbf{z})$

- Intuitively this means that **Y does not add any information** to predict X that isn't already offered by Z

Exploiting conditional independences

- Given any **ordering** of the variables $(X_1, \ldots, X_p)$ we can write:

$$p(X_1, X_2) = p(X_1)p(X_2 \mid X_1) = p(X_2)p(X_1 \mid X_2)$$

- We can **simplify** the factorisation by using **conditional independences:**

$$X_i \perp\!\!\!\perp X_j \mid X_{\mathbf{Z}} \implies p(X_i \mid X_j, X_{\mathbf{Z}}) = p(X_i \mid X_{\mathbf{Z}})$$

(special case $X_i \perp\!\!\!\perp X_j \implies p(X_i \mid X_j) = p(X_i)$)

Factorizing joint distributions

- Given any **ordering** of the variables $(X_1, \ldots, X_p)$ we can write:

$$p(X_1, \ldots, X_p) = p(X_1)p(X_2 \mid X_1) \ldots p(X_p \mid X_1, \ldots, X_{p-1})$$

- For example if $X \perp\!\!\!\perp Y \mid Z$:
  - $p(X, Y, Z) =$
  - $p(X, Z, Y) =$
  - $p(Z, Y, X) =$
  - ...

Bayesian networks

- We have a set of random variables $X_1, \ldots, X_p$ with joint $p(X_1, \ldots, X_p)$
- We have a DAG G, s.t. **each random variable X_i** is represented by **node i**

- We then say $p(X_1, \ldots, X_p)$ **factorizes over G** if

$$p(X_1, \ldots, X_p) = \prod_{i \in V} p(X_i \,|\, \mathbf{X}_{\text{pa}(i)})$$

- A **Bayesian network** (BN) is the tuple (G, p) s.t. p **factorizes over G**

Multiple BN can represent the same distribution

- Given any **ordering** of the variables $(X_1, \ldots, X_p)$ we can write:

$$p(X_1, \ldots, X_p) = p(X_1)p(X_2 \mid X_1) \ldots p(X_p \mid X_1, \ldots, X_{p-1})$$

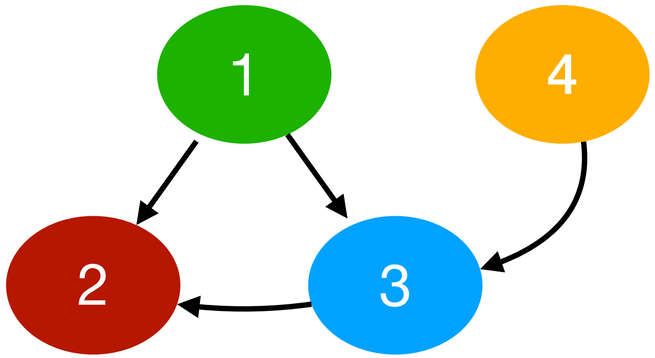
- For example if $X \perp\!\!\!\perp Y \mid Z$:

[Each factorisation can be represented with a DAG]

  - $p(X, Y, Z) = p(X)p(Y)p(Z \mid X, Y)$
  - $p(X, Z, Y) = p(X)p(Z \mid X)p(Y \mid Z)$
  - $p(Z, Y, X) = p(Z)p(Y \mid Z)p(X \mid Z)$
  - ...

Graph terminology: collider on a path

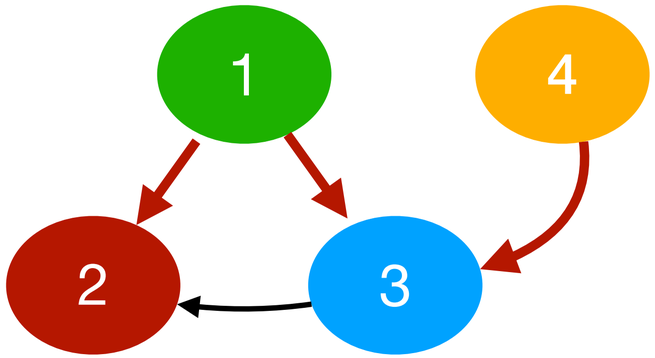
- A **path** between **node i and node j** is a sequence of **distinct nodes** $(i, \ldots, j)$ such that each two **consecutive nodes** are **adjacent**

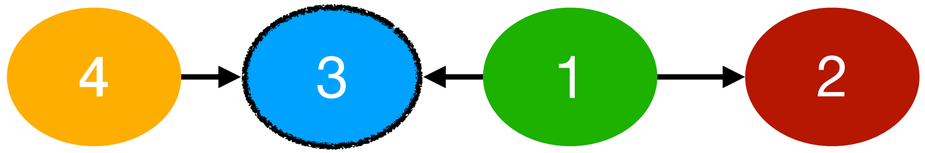


- A **collider** k on a **path** $\pi = (i, \ldots, j)$ is a non-endpoint node $(k \neq i, j)$ s.t. π contains $\rightarrow k \leftarrow$ (other non-endpoint nodes are **non-colliders**)

Graph terminology: collider on a path

- A **path** between **node i and node j** is a sequence of **distinct nodes** $(i, \ldots, j)$ such that each two **consecutive nodes** are **adjacent**



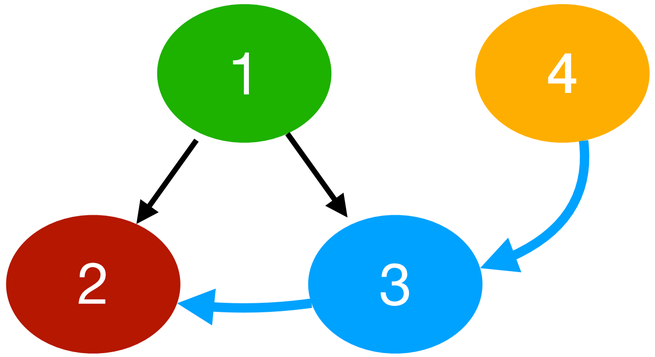


collider

- A **collider** k on a **path** $\pi = (i, \ldots, j)$ is a non-endpoint node $(k \neq i, j)$ s.t. π contains $\rightarrow k \leftarrow$ (other non-endpoint nodes are **non-colliders**)

Graph terminology: collider on a path

- A **path** between **node i and node j** is a sequence of **distinct nodes** $(i, \ldots, j)$ such that each two **consecutive nodes** are **adjacent**



non-collider

- A **collider** k on a **path** $\pi = (i, \ldots, j)$ is a non-endpoint node $(k \neq i, j)$ s.t. π contains $\rightarrow k \leftarrow$ (other non-endpoint nodes are **non-colliders**)

d-separation: blocked paths

- A **path** between **i and j** is **blocked by** $\mathbf{A} \subseteq \mathbf{V} \backslash \{i, j\}$ at least one holds:
  - There is a **non-collider** on the path that is in $\mathbf{A}$**, or**
  - There is a collider k on the path, but $\text{Desc}(k) \cap \mathbf{A} = \varnothing$

d-separation: blocked paths

- A **path** between **i and j** is **blocked by** $\mathbf{A} \subseteq \mathbf{V} \backslash \{i, j\}$ at least one holds:
  - There is a non-collider on the path that is in $\mathbf{A}$**, or**
  - There is a collider k on the path, but $\text{Desc}(k) \cap \mathbf{A} = \varnothing$
- Otherwise it is **active**

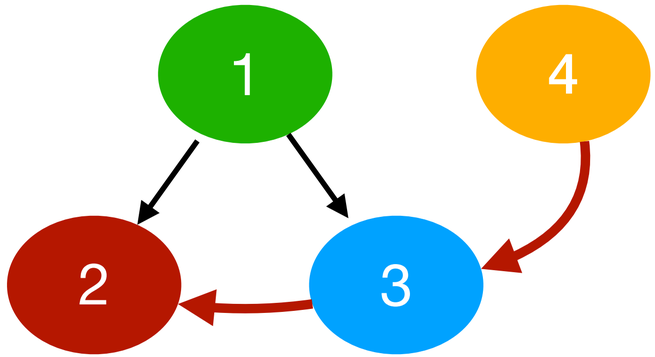
d-separation: blocked paths

- A **path** between **i and j** is **blocked by** $\mathbf{A} \subseteq \mathbf{V} \backslash \{i, j\}$ at least one holds:
  - There is a non-collider on the path that is in $\mathbf{A}$**, or**
  - There is a collider k on the path, but $\text{Desc}(k) \cap \mathbf{A} = \varnothing$
- Otherwise it is **active**

Note: descendants w.r.t. the **whole graph**

d-separation: blocked paths - example 1

- A **path** between **i and j** is **blocked by** $\mathbf{A} \subseteq \mathbf{V} \backslash \{i, j\}$ at least one holds:
  - There is a non-collider on the path that is in $\mathbf{A}$**, or**
  - There is a collider k on the path, but $\text{Desc}(k) \cap \mathbf{A} = \varnothing$
- Otherwise it is **active**



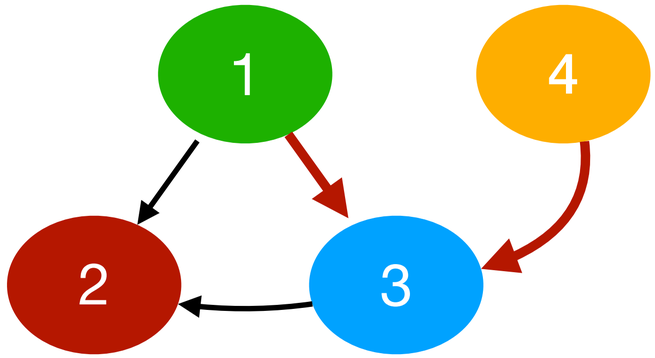
non-collider

If $3 \in \mathbf{A}$, the path is **blocked,**

otherwise it is **active**

d-separation: blocked paths - example 2

- A **path** between **i and j** is **blocked by** $\mathbf{A} \subseteq \mathbf{V} \backslash \{i, j\}$ at least one holds:
  - There is a non-collider on the path that is in $\mathbf{A}$, **or**
  - There is a collider k on the path, but $\text{Desc}(k) \cap \mathbf{A} = \varnothing$
- Otherwise it is **active**

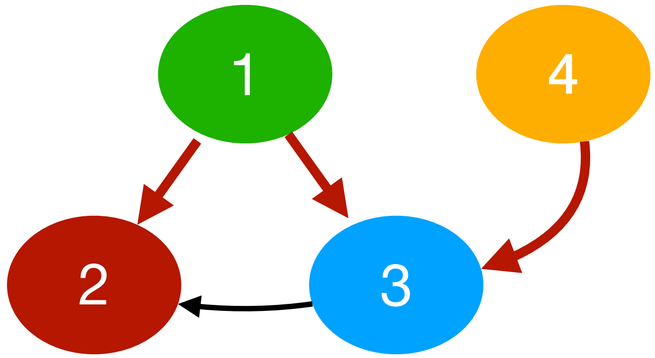


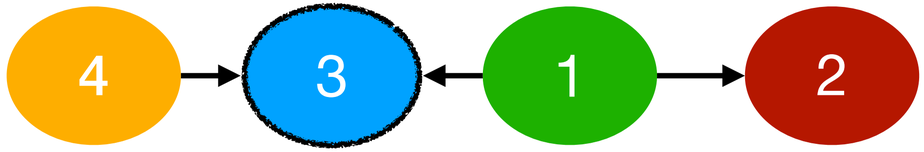
collider

If $3 \notin \mathbf{A}$ and $2 \notin \mathbf{A}$, the path is **blocked,** otherwise it is **active**

d-separation: blocked paths - example 3

- A **path** between **i and j** is **blocked by** $\mathbf{A} \subseteq \mathbf{V} \backslash \{i, j\}$ at least one holds:
  - There is a non-collider on the path that is in $\mathbf{A}$**, or**
  - There is a collider k on the path, but $\text{Desc}(k) \cap \mathbf{A} = \varnothing$
- Otherwise it is **active**





collider non-collider

If $1 \in \mathbf{A}$, the path is **blocked**

OR

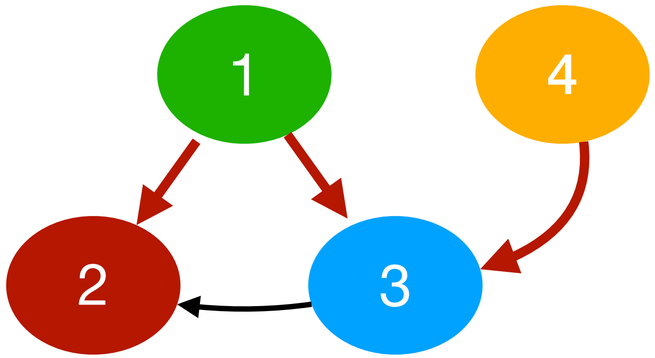
If $3 \notin \mathbf{A}$ and $2 \notin \mathbf{A}$, the path is **blocked**

d-separation

- **i and j** are **d-separated by** $\mathbf{A} \subseteq \mathbf{V} \backslash \{i, j\}$ if all paths between i, j are **blocked by** $\mathbf{A}$
  - We denote d-separation as $i \perp j \mid \mathbf{A}$

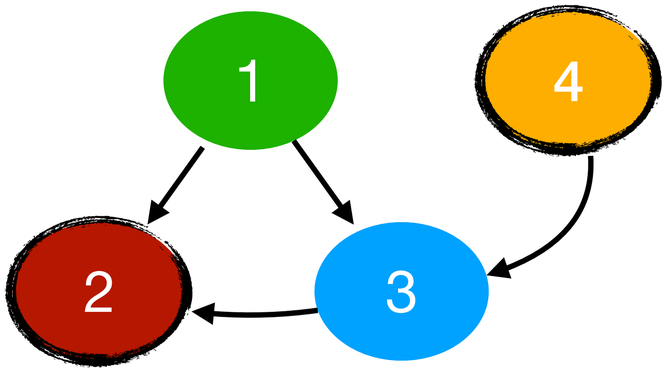
Note: d-separation is symmetric

- Otherwise we say they are **d-connected** $i \not\perp j \mid \mathbf{A}$



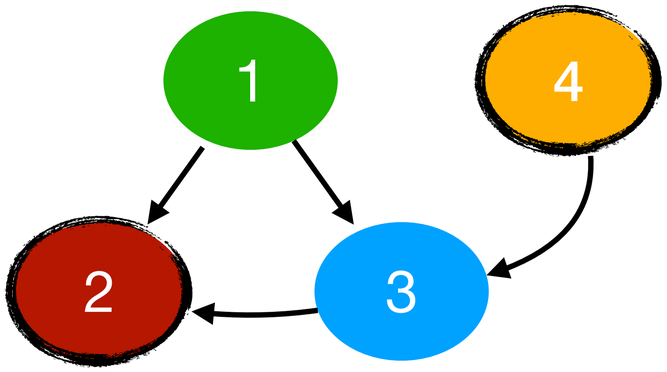
d-separation - complete example

- Which $\mathbf{A} \subseteq \mathbf{V} \backslash \{2,4\}$ d-separate 4 and 2?



d-separation - complete example

- Which $\mathbf{A} \subseteq \mathbf{V} \backslash \{2,4\}$ d-separate 4 and 2?

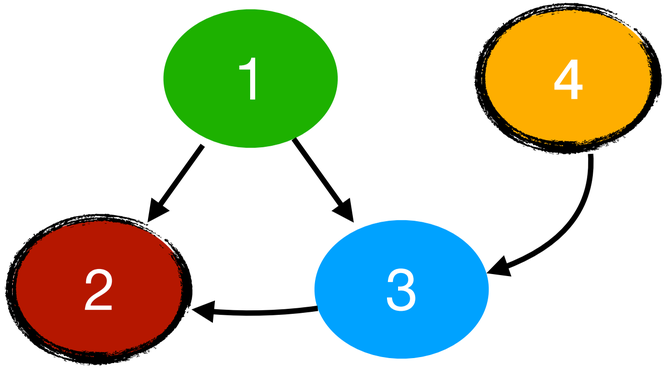


non-collider

If $3 \in \mathbf{A}$, the path is **blocked**

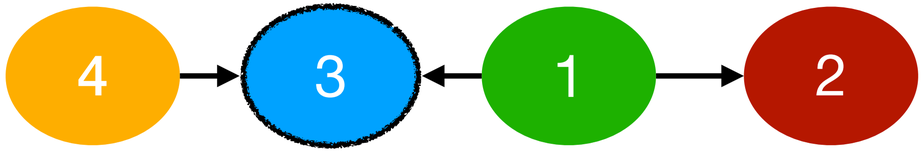
d-separation - complete example

- Which $\mathbf{A} \subseteq \mathbf{V} \backslash \{2,4\}$ d-separate 4 and 2?



non-collider

If $3 \in \mathbf{A}$, the path is **blocked**



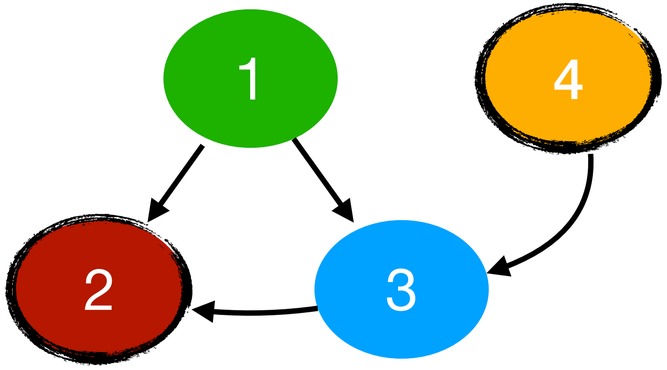
collider non-collider

If $1 \in \mathbf{A}$, the path is **blocked, OR**

If $3 \notin \mathbf{A}$, the path is **blocked**

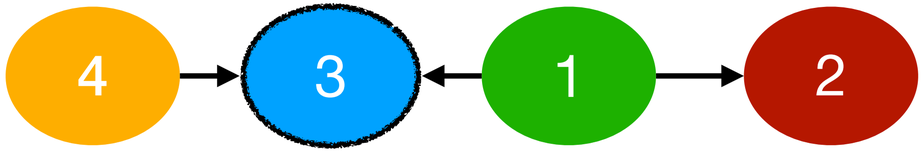
d-separation - complete example

- Which $\mathbf{A} \subseteq \mathbf{V} \backslash \{2,4\}$ d-separate 4 and 2?



non-collider

If $3 \in \mathbf{A}$, the path is **blocked**



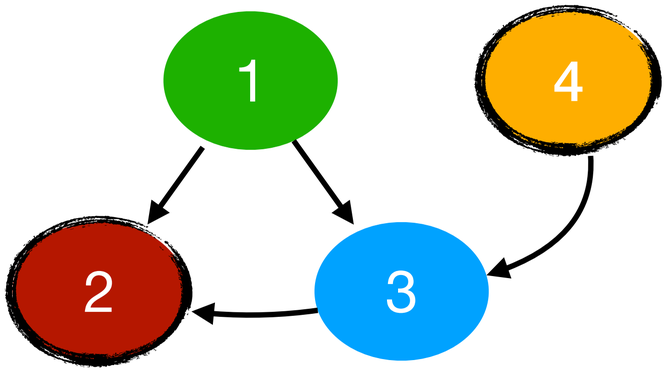
collider non-collider

If $1 \in \mathbf{A}$, the path is **blocked, OR**

~~If $3 \notin \mathbf{A}$, the path is **blocked**~~

d-separation - complete example

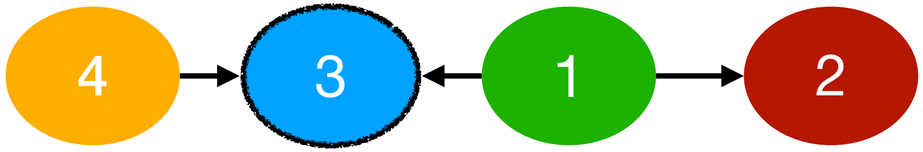
- Which $\mathbf{A} \subseteq \mathbf{V} \backslash \{2,4\}$ d-separate 4 and 2?



non-collider

If $3 \in \mathbf{A}$, the path is **blocked**

$$\mathbf{A} = \{1,3\}$$



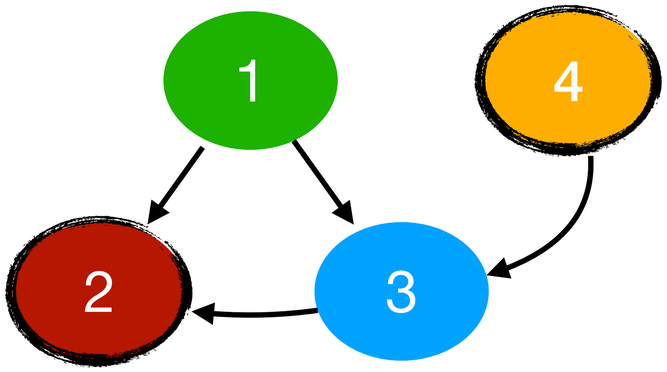
collider non-collider

If $1 \in \mathbf{A}$, the path is **blocked, OR**

~~If $3 \notin \mathbf{A}$, the path is **blocked**~~

d-separation - complete example

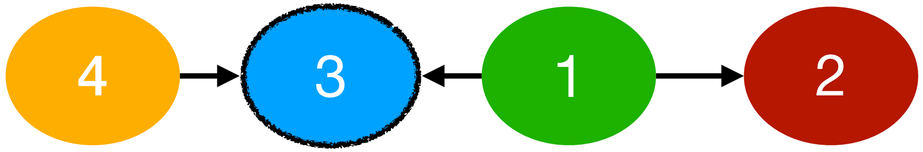
- Which $\mathbf{A} \subseteq \mathbf{V} \backslash \{2,4\}$ d-separate 4 and 2?



$2 \perp 4 \mid \{1,3\}$

non-collider

If $3 \in \mathbf{A}$, the path is **blocked**



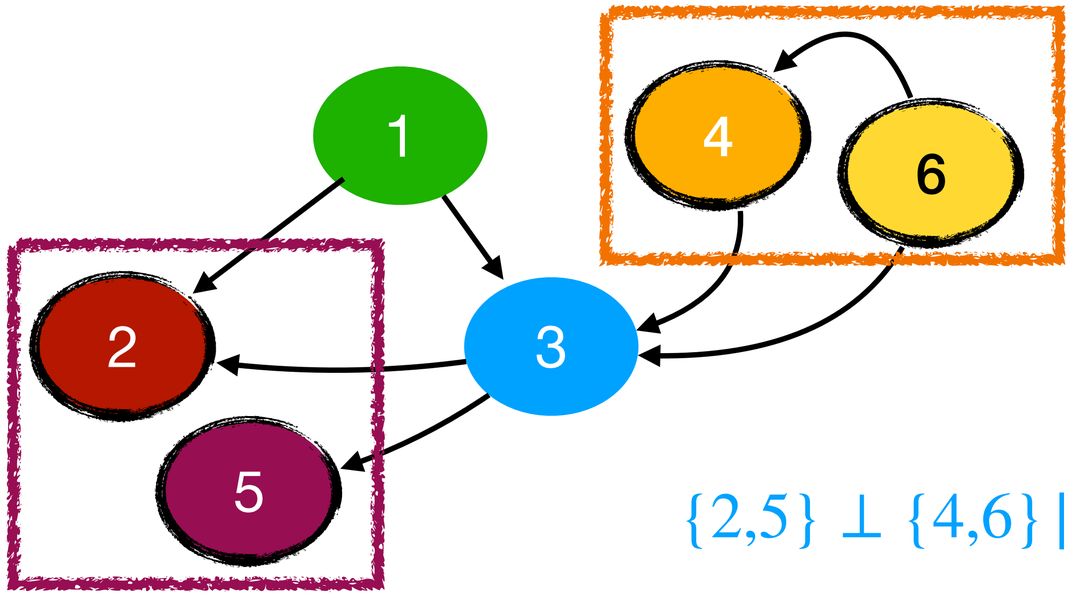
collider non-collider

If $1 \in \mathbf{A}$, the path is **blocked**

$\mathbf{A} = \{1,3\}$

d-separation for sets of nodes

- Sets of nodes $\mathbf{I}$ **and** $\mathbf{J}$ are **d-separated by** $\mathbf{A} \subseteq \mathbf{V} \setminus \mathbf{I} \cup \mathbf{J}$**,** if all paths between $i \in \mathbf{I}$ and $j \in \mathbf{J}$ are **blocked by** $\mathbf{A}$
  - We denote d-separation as $\mathbf{I} \perp \mathbf{J} \mid \mathbf{A}$



$\{2,5\} \perp \{4,6\} \mid \{1,3\}$

Global Markov Property and faithfulness

- If (G, p) is a Bayesian network with a DAG $G = (\mathbf{V}, \mathbf{E})$, i.e. p **factorizes according to G**, then for any disjoint $\mathbf{A}, \mathbf{B}, \mathbf{C} \subseteq \mathbf{V}$:

$$\mathbf{A} \perp_G \mathbf{B} \,|\, \mathbf{C} \implies X_{\mathbf{A}} \perp\!\!\!\perp_p X_{\mathbf{B}} \,|\, X_{\mathbf{C}}$$

Global Markov Property and faithfulness

- If (G, p) is a Bayesian network with a DAG $G = (\mathbf{V}, \mathbf{E})$, i.e. p **factorizes according to G**, then for any disjoint $\mathbf{A}, \mathbf{B}, \mathbf{C} \subseteq \mathbf{V}$:

$$\mathbf{A} \perp_G \mathbf{B} \,|\, \mathbf{C} \implies X_{\mathbf{A}} \perp\!\!\!\perp_p X_{\mathbf{B}} \,|\, X_{\mathbf{C}}$$

- **d-separations** that can be read purely from a graph imply **conditional independences** in the random variables and data generated by the graph

Global Markov Property and faithfulness

- If (G, p) is a Bayesian network with a DAG $G = (\mathbf{V}, \mathbf{E})$, i.e. p **factorizes according to G**, then for any disjoint $\mathbf{A}, \mathbf{B}, \mathbf{C} \subseteq \mathbf{V}$:

$$\mathbf{A} \perp_G \mathbf{B} \,|\, \mathbf{C} \implies X_{\mathbf{A}} \perp\!\!\!\perp_p X_{\mathbf{B}} \,|\, X_{\mathbf{C}}$$

- **d-separations** that can be read purely from a graph imply **conditional independences** in the random variables and data generated by the graph

- The reverse is not true in general, but if it is we say p **is faithful to G** and:

$$\mathbf{A} \perp_G \mathbf{B} \,|\, \mathbf{C} \iff X_{\mathbf{A}} \perp\!\!\!\perp_p X_{\mathbf{B}} \,|\, X_{\mathbf{C}}$$

Why should we care about Bayesian networks?

- We have a set of random variables $X_1, \ldots, X_p$ with joint $p(X_1, \ldots, X_p)$
- We have a DAG G, s.t. **each random variable X_i** is represented by **node i**

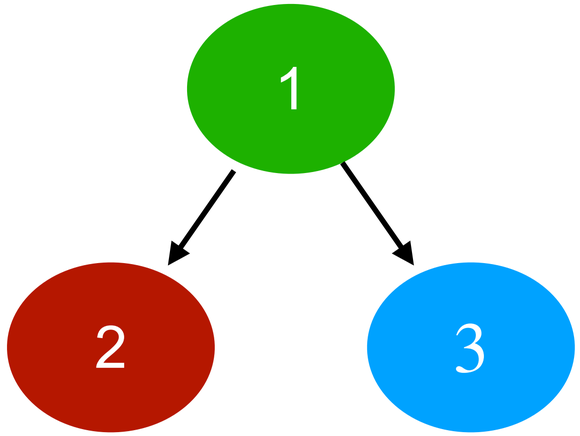
- We then say $p(X_1, \ldots, X_p)$ **factorizes over G** if

$$p(X_1, \ldots, X_p) = \prod_{i \in V} p(X_i \mid \mathbf{X}_{\mathrm{pa}(i)})$$

They can help simplify the factorisation

We can easily read conditional independences (d-separation)

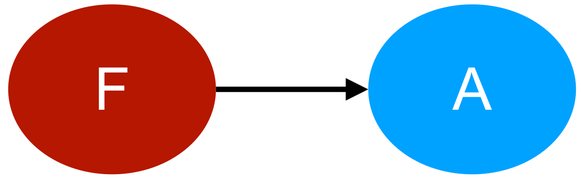
They can represent causal models



BNs vs causal BNs - example

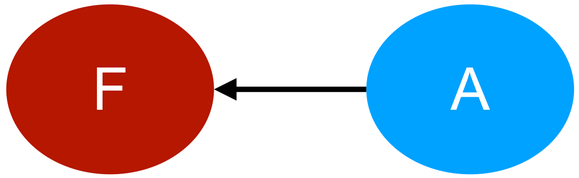
- Fire (F) and Alarm (A) with $p(F, A)$ and $A \not\perp\!\!\!\perp F$ can be factorized as:

$p(F, A) = p(F)\, p(A|F)$



CAUSAL
(lighting a fire triggers alarm)

$p(F, A) = p(A)\, p(F|A)$



NOT-CAUSAL
(triggering alarm does not light a fire)