Bayesian networks: Basics

Fabio Stella
Department of Informatics, Systems and Communication
University of Milan-Bicocca
fabio.stella@unimib.it

OUTLOOK

- BAYESIAN AND CAUSAL NETWORKS
- CHAIN RULE AND FACTORIZATION
- CHAIN, FORK AND COLLIDER
- D-SEPARATION

In the previous lecture we used **CAUSAL GRAPHS** aid intuition, while in the next lectures, we introduce the formalisms that underlie this intuition.

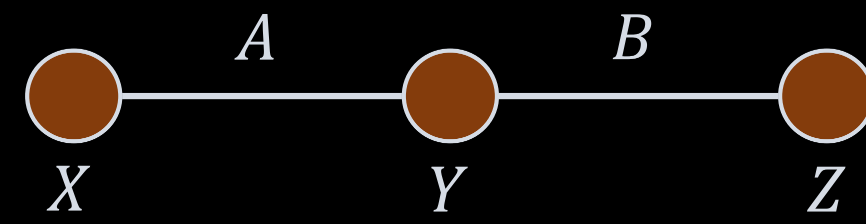


Figure 3.1

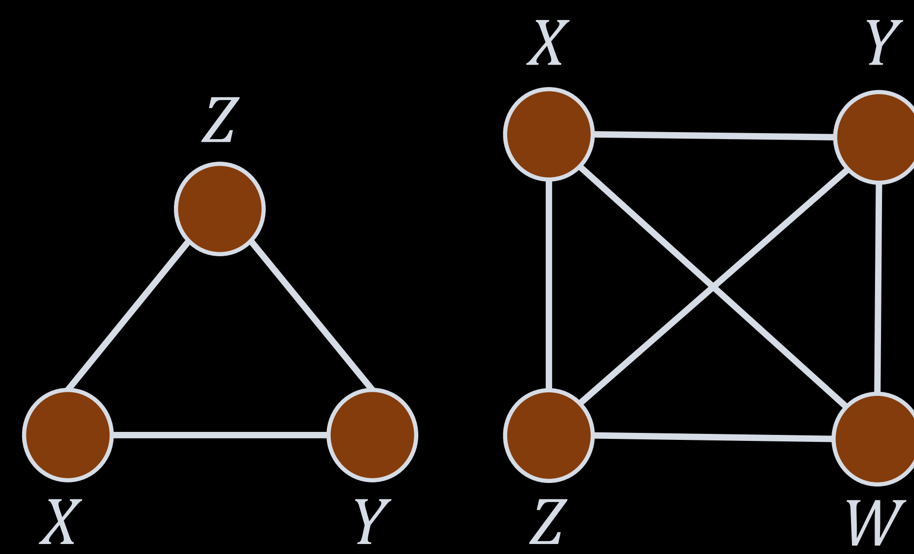
GRAPH; consists of a collection of **NODES** (vertices) and **EDGES**.

ADJACENT NODES; if there is an edge between them.

X and Y as well as Y and Z are adjacent

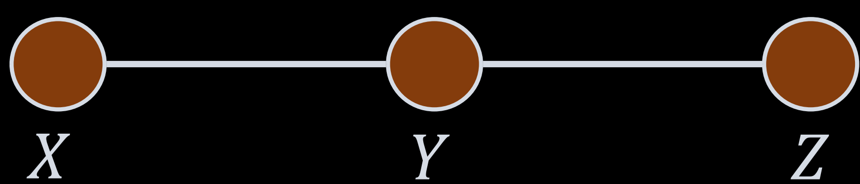
X and Z are not adjacent

COMPLETE GRAPH; if there is an edge between every pair of nodes.

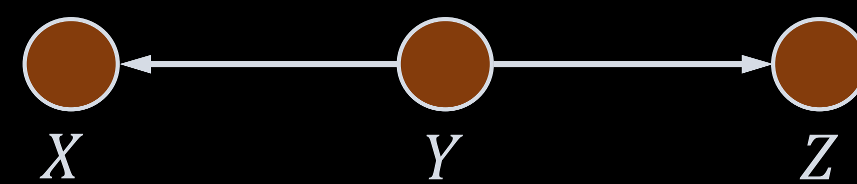


COMPLETE GRAPHS

DIRECTED AND UN-DIRECTED GRAPHS; a graph whose edges are all directed is said to be a directed graph while a graph whose edges are all un-directed is said to be an un-directed graph.



UN-DIRECTED GRAPH



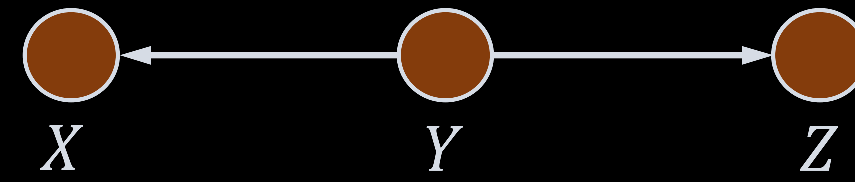
DIRECTED GRAPH

In a **DIRECTED GRAPH** we let

- $pa(X)$ be the **PARENT SET** of node X
- $ch(Y)$ be the **CHILD SET** of node Y

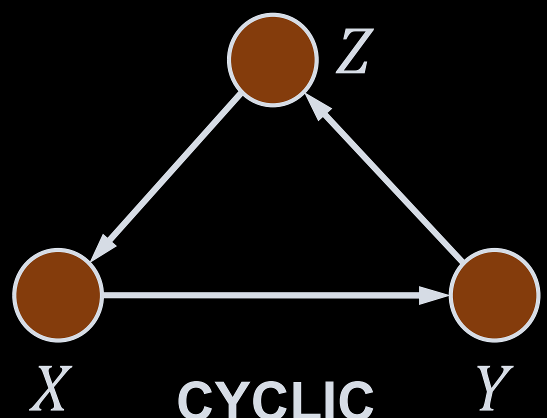
$$pa(X) = Y$$
$$pa(Y) = \emptyset$$

$$ch(Y) = \{X, Z\}$$
$$ch(X) = \emptyset$$

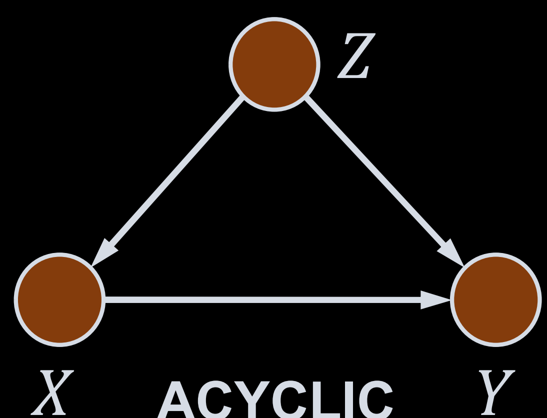


PATH

$X \longleftarrow Y \longrightarrow Z$

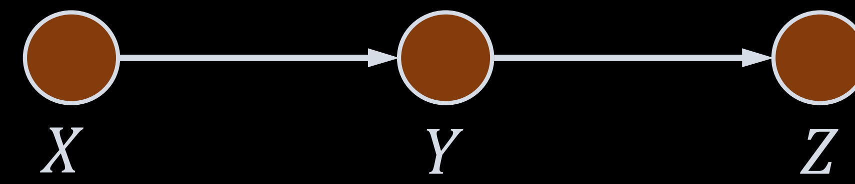


CYCLIC



ACYCLIC

An **PATH** is any sequence of adjacent nodes, regardless of the direction of the edges that join them.

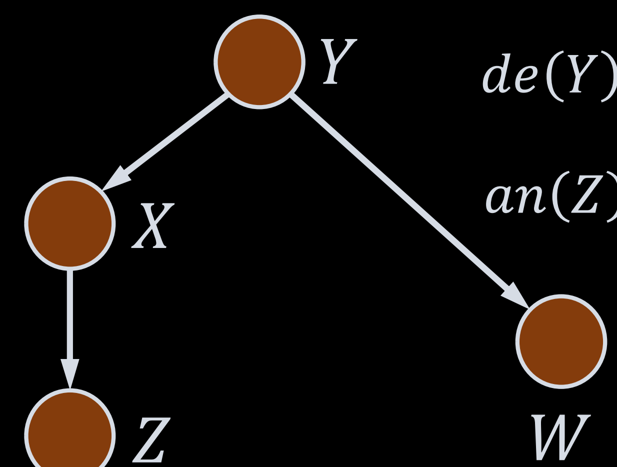


A path between two nodes is a **DIRECTED PATH** if can be traced along the arrows, that is, if no node on the path has two edges on the path directed into it, or two edges directed out of it.

DIRECTED PATH

$X \longrightarrow Y \longrightarrow Z$

- $de(Y)$ is the set of **DESCENDANTS** of node Y, i.e., the set of nodes which can be reached by Y.
- $an(Z)$ is the set of **ANCESTORS** of node Z, i.e., the set of nodes from which Z can be reached.



$$de(Y) = \{X, Z, W\}$$

$$an(Z) = \{X, Y\}$$

Probabilistic Graphical Models (PGMs)

(statistical models)

BAYESIAN NETWORKS (BNs)

(main models)

Causal Graphical Models (CGMs)

(causal models)

CAUSAL NETWORKS (CNs)

	X	
Y	0	1
0	0,7	0,1
1	0,3	0.9

PROBABILITY TABLE

Assume we only care about modeling **ASSOCIATION**, without any causal modeling.

We want to model the **DATA DISTRIBUTION**

$$P(X_1, X_2, \dots, X_n)$$

However, if we were to model discrete random variables, i.e., by using **PROBABILITY TABLES**, it would take an exponential number of parameters!!!

$$P(X_1, X_2, \dots, X_n) = \overbrace{P(X_1)\, P(X_2|X_1)\, P(X_3|X_1, X_2) \cdots P(X_n|X_1, \dots, X_{n-1})}^{\text{CHAIN RULE}} = P(X_1) \prod_{i=2}^{n} P(X_i|X_1, \dots, X_{i-1})$$

Assume each X_i to be binary and consider how we would model

$$P(X_n|X_1, \dots, X_{n-1})$$

X_1	X_2	X_3	$P(X_4 \mid X_1, X_2, X_3)$
0	0	0	α_1
0	0	1	α_2
0	1	0	α_3
0	1	1	α_4
1	0	0	α_5
1	0	1	α_6
1	1	0	α_7
1	1	1	α_8

Table 3.1

2^{n-1} parameters

$2^{4-1} = 8$

n	2^(n-1)
2	2
3	4
10	512
20	524.288
21	1.048.576

quickly intractable

?

model local dependencies

$$P(X_4|X_1, X_2, X_3) = P(X_4|X_3)$$

X_4 locally depends on X_3 only

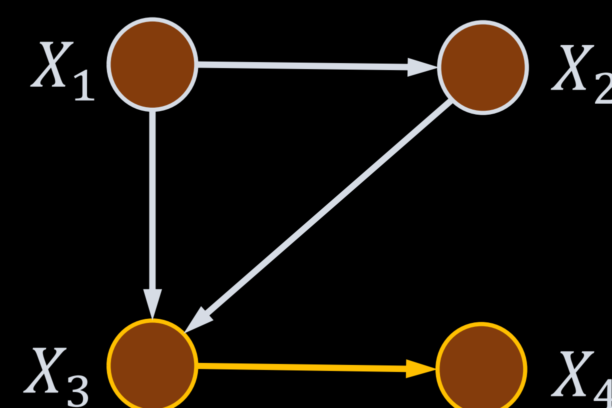


Figure 3.2

In the case of $P(X_1, X_2, X_3, X_4)$, the **CHAIN RULE** allows us to write as follows

$$P(X_1, X_2, X_3, X_4) = P(X_1)P(X_2|X_1)P(X_3|X_1, X_2)P(X_4|X_1, X_2, X_3)$$

LOCAL MARKOV ASSUMPTION

Given its parents $pa(X)$ in the DAG, a node X is independent of all its non-descendants.

BAYESIAN NETWORK FACTORIZATION

Given a probability distribution P and a DAG $\mathcal{G}$, P factorizes according to $\mathcal{G}$ if

$$P(X_1, X_2, \dots, X_n) = \prod_{i=1}^{n} P\big(X_i | pa(X_i)\big)$$

$$P(X_1, X_2, X_3, X_4) = P(X_1)P(X_2|X_1)P(X_3|X_1, X_2)P(X_4|X_3)$$

MARKOV PROBABILITY DISTRIBUTION

A probability distribution P is said to be (locally) Markov with respect to a DAG $\mathcal{G}$ if all nodes X satisfy the local Markov assumption.

As important as the local Markov assumption is, it only gives us information about the independencies in P that a DAG $\mathcal{G}$ implies.

It does not even tell us that if X and Y are adjacent in the DAG $\mathcal{G}$, then X and Y are dependent.

This additional information is very commonly assumed in causal DAGs.

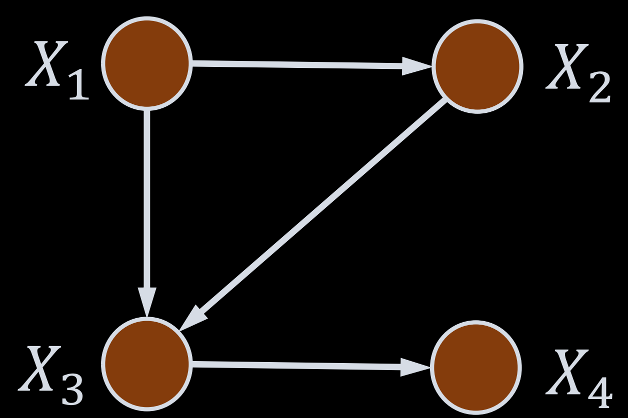


Figure 3.2

To get this guaranteed dependence between adjacent nodes, we will generally assume a slightly stronger assumption than the **LOCAL MARKOV ASSUMPTION**.

MINIMALITY ASSUMPTION

> Given its parents $pa(X)$ in the DAG $\mathcal{G}$, a node X is independent of all its non-descendants (local Markov assumption).
>
> Adjacent nodes in the DAG $\mathcal{G}$ are dependent.

If P is Markov with respect to a DAG $\mathcal{G}$, then we know that:

- P satisfies a set of independencies that are specific to the structure of $\mathcal{G}$.
- If P and $\mathcal{G}$ also satisfy minimality, then this set of independencies is minimal in the sense that P does not satisfy any additional independencies in $\mathcal{G}$.
  This is equivalent to saying that adjacent nodes are dependent.

If the DAG $\mathcal{G}$ simply consists of two connected nodes, X and Y, as in Figure 3.3, the **LOCAL MARKOV ASSUMPTION** would tell

Figure 3.3

LOCAL MARKOV ASSUMPTION

$$P(X,Y) = P(X)P(Y|X)$$

$$P(X,Y) = P(X)P(Y)$$

X and Y are independent

The minimality assumption does not allow this additional independence.

The **MINIMALITY ASSUMPTION** would tell us to factorize $P(X,Y)$ as $P(X)P(Y|X)$, and it would tell us that no additional independencies $(X \quad Y)$ exist in P that are minimal with respect to Figure 3.3.

To get this guaranteed dependence between adjacent nodes, we will generally assume a slightly stronger assumption than the **LOCAL MARKOV ASSUMPTION.**

MINIMALITY ASSUMPTION

> Given its parents $pa(X)$ in the DAG $\mathcal{G}$, a node X is independent of all its non-descendants (local Markov assumption).
>
> Adjacent nodes in the DAG $\mathcal{G}$ are dependent.

Because removing edges in a Bayesian network is equivalent to adding independencies, the **MINIMALITY ASSUMPTION** is equivalent to saying that we can't remove any more edges from the graph $\mathcal{G}$.

In other words, we can say that every edge is "active."

Concretely, consider that P and $\mathcal{G}$ are **MARKOV COMPATIBLE** i.e, P factorizes according to $\mathcal{G}$, and that $\mathcal{G}$' is what we get when we remove some edge from $\mathcal{G}$. If P is also Markov with respect to $\mathcal{G}$', then P is not minimal with respect to $\mathcal{G}$.

If the DAG $\mathcal{G}$ simply consists of two connected nodes, X and Y, as in Figure 3.3, the **LOCAL MARKOV ASSUMPTION** would tell

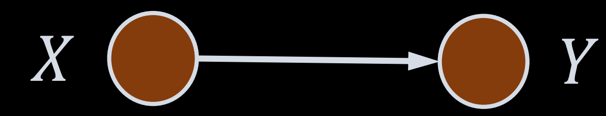


Figure 3.3

Consider the DAG $\mathcal{G}$ in Figure 3.3.

Assume P and $\mathcal{G}$ are Markov compatible.

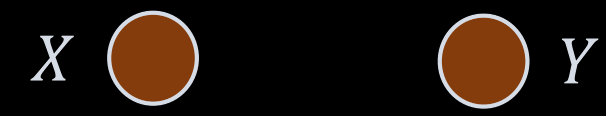


Figure 3.4

Let $\mathcal{G}$' (Figure 3.4) be what we get when we remove the edge from $\mathcal{G}$ (Figure 3.3).

If P is also Markov with respect to $\mathcal{G}$', then P is not minimal with respect to $\mathcal{G}$.

To get this guaranteed dependence between adjacent nodes, we will generally assume a slightly stronger assumption than the **LOCAL MARKOV ASSUMPTION**.

MINIMALITY ASSUMPTION

1. Given its parents $pa(X)$ in the DAG $\mathcal{G}$, a node X is independent of all its non-descendants (local Markov assumption).
2. Adjacent nodes in the DAG $\mathcal{G}$ are dependent.

Now that we know about the **MINIMALITY ASSUMPTION** and what it implies about how distributions P factorize when they are Markov with respect to some DAG $\mathcal{G}$ (**LOCAL MARKOV ASSUMPTION**), we can discuss the flow of association in DAGs.

However, everything in this section is purely statistical, thus to be ready to discuss the flow of causation in DAGs, we first need to make causal assumptions.

If the DAG $\mathcal{G}$ simply consists of two connected nodes, X and Y, as in Figure 3.3, the **LOCAL MARKOV ASSUMPTION** would tell

Figure 3.3

Consider the DAG $\mathcal{G}$ in Figure 3.3.

Assume P and $\mathcal{G}$ are Markov compatible.

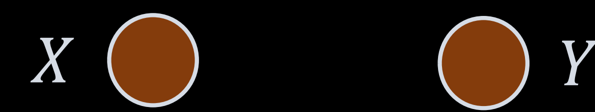


Figure 3.4

Let $\mathcal{G}$' (Figure 3.4) be what we get when we remove the edge from $\mathcal{G}$ (Figure 3.3).

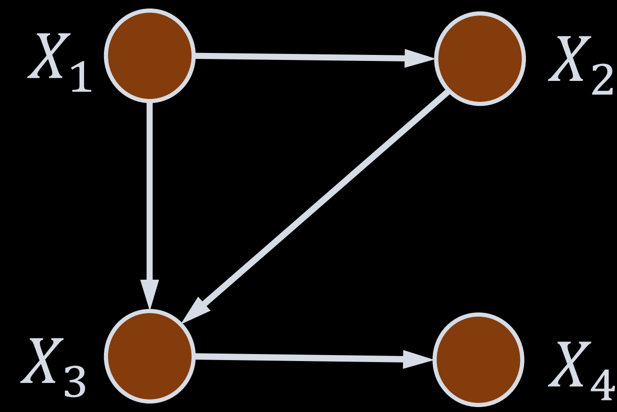
If P is also Markov with respect to $\mathcal{G}$', then P is not minimal with respect to $\mathcal{G}$.

Up to now all we presented was about statistical models and modeling association.

We now need to introduce some causal assumptions, turn them into causal models for allowing the study of causation.

In order to introduce causal assumptions, we must first understand what it means for X to be a cause of Y.

Figure 3.2



WHAT IS A CAUSE?

A variable X is said to be a cause of a variable Y if Y can change in response to changes in X.

X_1 is a direct cause of X_2, X_3.

X_2 is a direct cause of X_3.

X_3 is a direct cause of X_4.

Another phrase commonly used to describe this primitive is that Y "**LISTENS**" to X.

(STRICT) CAUSAL EDGES ASSUMPTION

In a DAG, every parent is a direct cause of all its children.

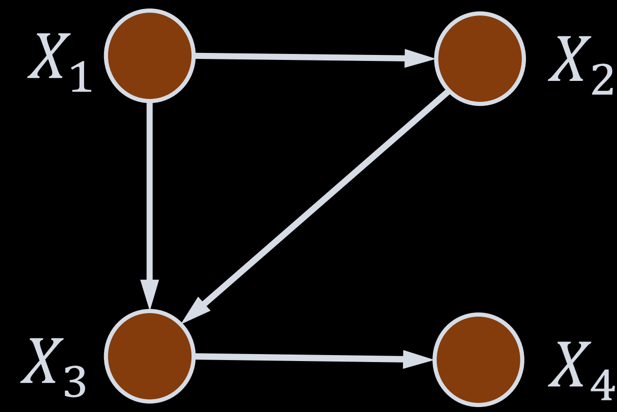
CAUSAL GRAPH

A causal graph is a DAG that satisfies the strict causal edges assumption.

Adding the **CAUSAL EDGES ASSUMPTION**, implies that **DIRECTED PATHS** in the DAG take on a very special meaning; they correspond to causation.

This is in contrast to other **PATHS** in the graph, which association may flow along, but causation certainly may not.

Figure 3.2



X_1 is a cause of X_2, X_3 and X_4.

X_1 is a direct cause of X_2 and X_3.

X_1 is an indirect cause of X_4.

Association flows from X_2 to X_3 through the path $X_2 \longleftarrow X_1 \longrightarrow X_3$, causation not.

WHAT IS A CAUSE?

A variable X is said to be a cause of a variable Y if Y can change in response to changes in X.

Another phrase commonly used to describe this primitive is that Y "**LISTENS**" to X.

(STRICT) CAUSAL EDGES ASSUMPTION

In a DAG, every parent is a direct cause of all its children.

CAUSAL GRAPH

A causal graph is a DAG that satisfies the strict causal edges assumption.

Moving forward, we will now think of the edges of graphs as causal, in order to describe concepts intuitively with causal language.

However, all of the associational claims about statistical independence will still hold, even when the edges do not have causal meaning like in the vanilla Bayesian networks.

The main assumptions that we need for causal graphical models to tell us how association and causation flow between variables are the following:

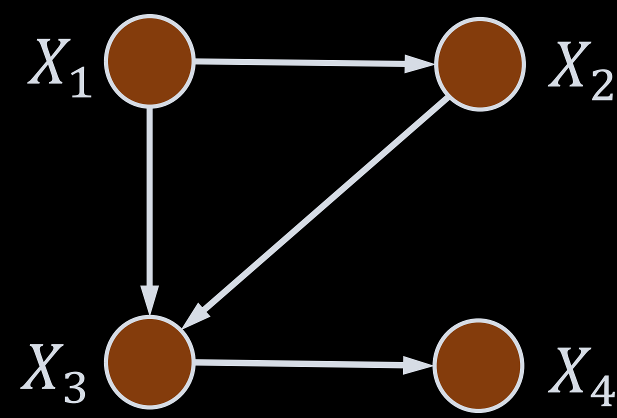
LOCAL MARKOV ASSUMPTION

Given its parents $pa(X)$ in the DAG, a node X is independent of all its non-descendants.

(STRICT) CAUSAL EDGES ASSUMPTION

In a DAG, every parent is a direct cause of all its children.

Figure 3.2



X_1 is a cause of X_2, X_3 and X_4.

X_1 is a direct cause of X_2 and X_3.

X_1 is an indirect cause of X_4.

Association flows from X_2 to X_3 through the path $X_2 \longleftarrow X_1 \longrightarrow X_3$, causation not.

CAUSAL GRAPH

A causal graph is a DAG that satisfies the strict causal edges assumption.

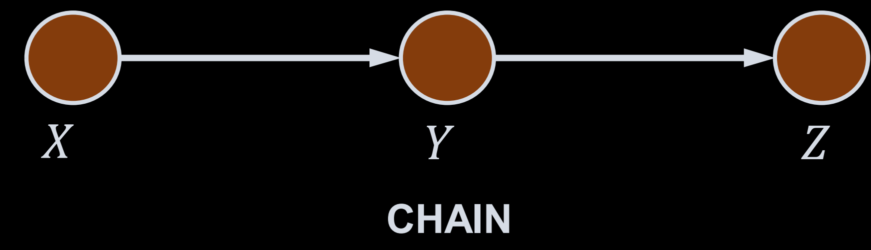
To understand the difference between association flow and causal flow in DAGs, we need the following minimal building blocks

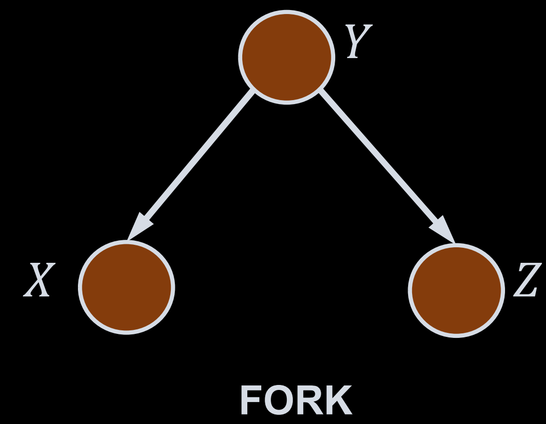
- chain
- fork
- collider
- two un-connected nodes
- two connected nodes

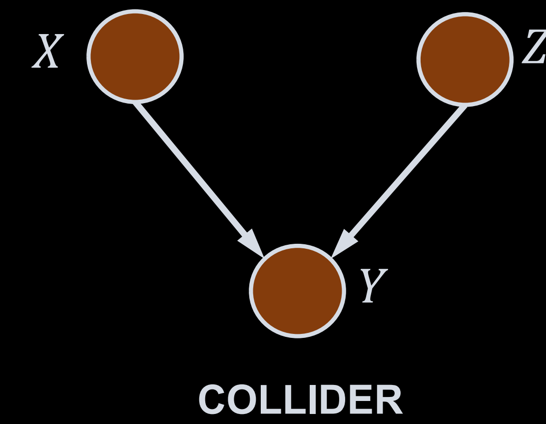
By “**FLOW OF ASSOCIATION**,” we mean whether any two nodes in a graph are **ASSOCIATED** or **NOT ASSOCIATED**.

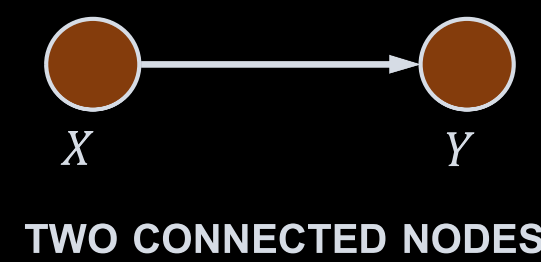
In other terms, we want to know whether two nodes are (statistically) **DEPENDENT** or (statistically) **INDEPENDENT**.

However, we will also study whether two nodes are **CONDITIONALLY INDEPENDENT OR NOT**.









Given a graph consisting of just two unconnected nodes, these nodes are not associated, because there is no edge between them.

To show this, consider the factorization of the joint probability

$$P(X, Y)$$

that the Bayesian network factorization gives us:

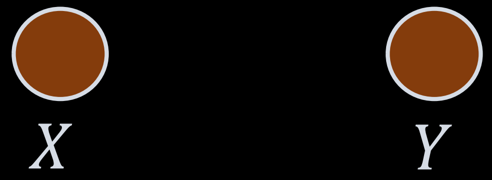
BAYESIAN NETWORK FACTORIZATION

Given a probability distribution P and a DAG $\mathcal{G}$, P factorizes according to $\mathcal{G}$ if

$$P(X_1, X_2, \dots, X_n) = \prod_{i=1}^{n} P(X_i | pa(X_i))$$

$$P(X, Y) = P(X)P(Y)$$

Thus, applying the Bayesian network factorization immediately gives us a proof that the two nodes X and Y are unassociated (independent). (we assume that P is Markov w.r.t. the graph $\mathcal{G}$).



TWO UN-CONNECTED NODES

On the contrary, if there is an edge between the two nodes, then the two nodes are associated.

We exploit the causal edges assumption

(STRICT) CAUSAL EDGES ASSUMPTION

> In a DAG, every parent is a direct cause of all its children.

which means that X is a cause of Y.

Now, since X is a cause of Y, by definition

WHAT IS A CAUSE?

> A variable X is said to be a cause of a variable Y if Y can change in response to changes in X.

Y must be able to change in response to changes in X, so Y and X are associated.

In general, any time two nodes are adjacent in a causal graph, they are associated.

TWO CONNECTED NODES

CHAINS and **FORKS** share the same set of **DEPENDENCIES**:

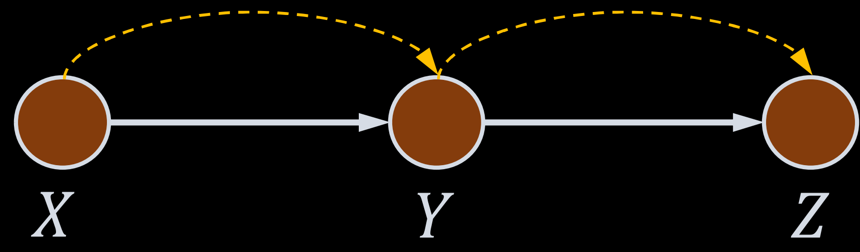
- X and Y are dependent
- Y and Z are dependent

for the same reason that we just discussed (i.e., two connected nodes)

Adjacent nodes are always dependent when we make the causal edges assumption.

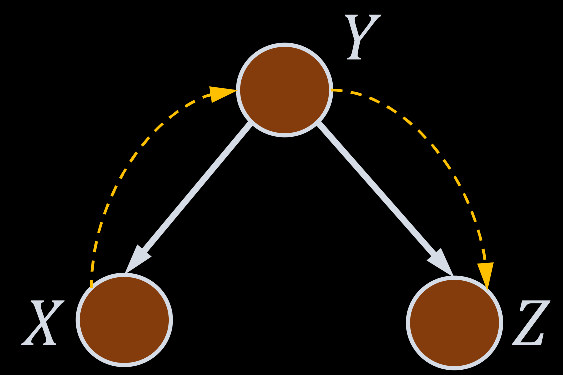
(STRICT) CAUSAL EDGES ASSUMPTION

In a DAG, every parent is a direct cause of all its children.



CHAIN

CHAIN: X causes changes in Y which then causes changes in Z.



FORK

FORK: the same value that Y takes on is used to determine both the value that X takes on and the value that Z takes on.

In other words, X and Z are associated through their (shared) common cause Y. (mind likely dependent!!!)

What about X and Z?

Does association flow from X to Z through Y in chains and forks?

Usually, yes, X and Z are associated in both chains and forks.

CHAINS and **FORKS** also share the same set of **INDEPENDENCIES**.

When we condition on Y in both graphs, it blocks the flow of association from X to Z.

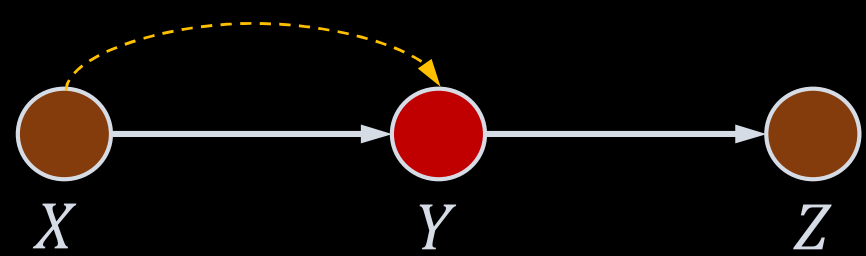
This is because of the

LOCAL MARKOV ASSUMPTION

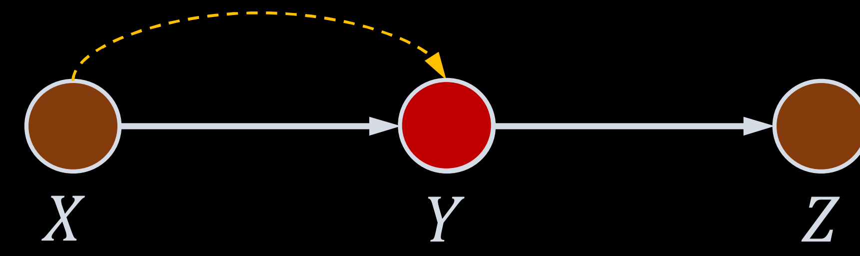
> Given its parents $pa(X)$ in the DAG, a node X is independent of all its non-descendants.

Therefore, when we condition on Y (Z's parent in both graphs), Z becomes independent of X (and viceversa).

This independence is an instance of a **BLOCKED PATH**.

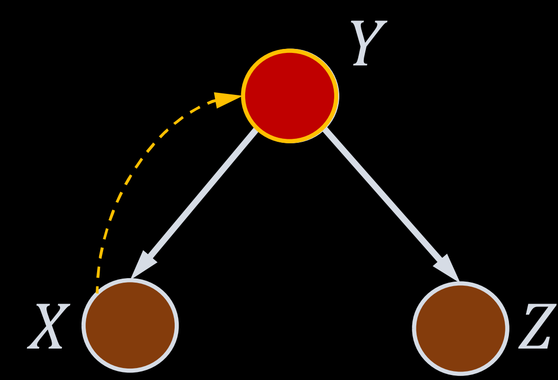


BLOCKED PATH



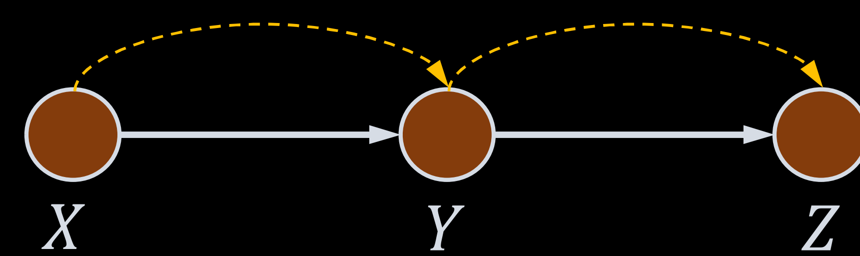
CHAIN

CHAIN: Y blocks the flow of association from X to Z.



FORK

FORK: Y blocks the flow of association from X to Z.



UN-BLOCKED PATH

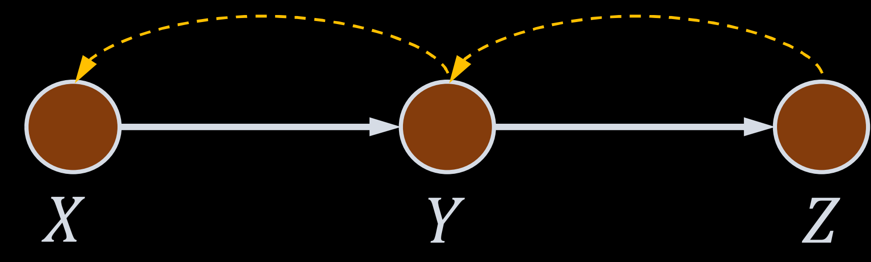
It is worth noticing that association flows also from Z to X by the same paths.

In general, the **FLOW OF ASSOCIATION IS SYMMETRIC**.

Therefore,

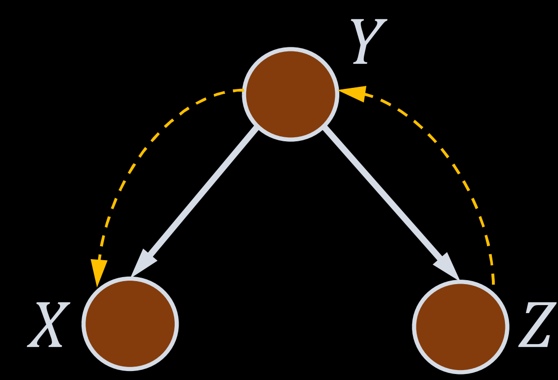
- not conditioning on Y: association flows from X to Z and flows from Z to X.
- conditioning on Y: association does not flow from X to Z and does not flow from Z to X, i.e., Y blocks the path from X to Z as well as it blocks the path from Z to X.

THIS IS JUST INTUITION!!!



CHAIN

CHAIN: association flows from Z to X.



FORK

FORK: association flows from Z to X.

To prove that $X \perp\!\!\!\perp Z \mid Y$, we need to show that

$$P(X,Z|Y) = P(X|Y)P(Z|Y)$$

we exploit ⇨

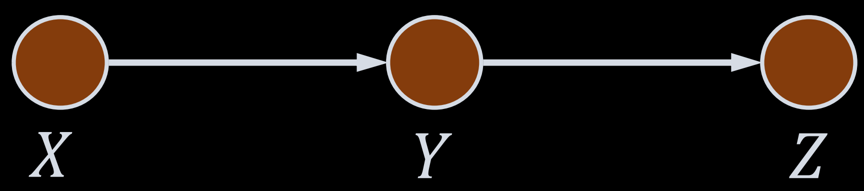
LOCAL MARKOV ASSUMPTION

Given its parents $pa(X)$ in the DAG, a node X is independent of all its non-descendants.

To prove that $X \perp\!\!\!\perp Z \mid Y$, we need to show that

$$P(X, Z|Y) = P(X|Y)P(Z|Y)$$

We give the proof for chains.



BAYESIAN NETWORK FACTORIZATION

Given a probability distribution P and a DAG $\mathcal{G}$, P factorizes according to $\mathcal{G}$ if

$$P(X_1, X_2, \dots, X_n) = \prod_{i=1}^{n} P\big(X_i|pa(X_i)\big)$$

$$P(X, Y, Z) = P(X)P(Y|X)P(Z|Y)$$

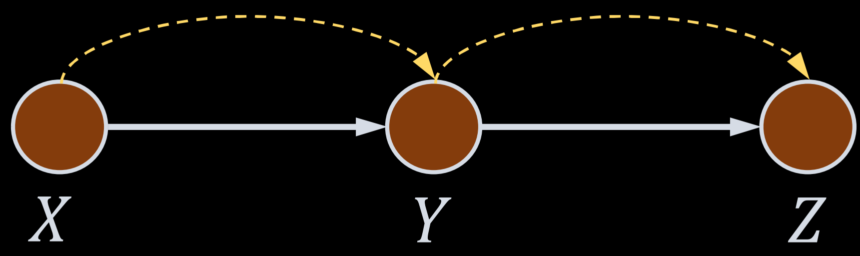
$$P(X, Z|Y) = \frac{P(X, Y, Z)}{P(Y)} \quad \text{(Bayes Theorem)}$$

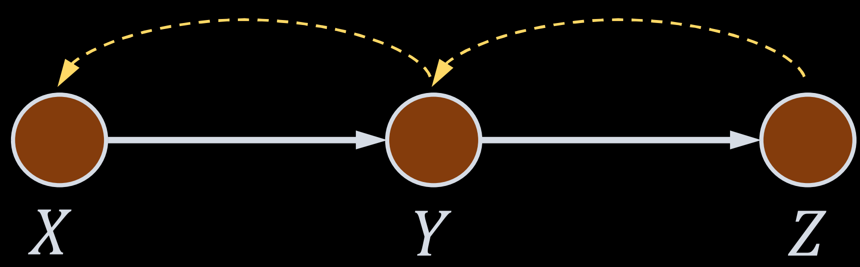
$$= \frac{P(X)P(Y|X)P(Z|Y)}{P(Y)}$$

$$= \frac{P(X, Y)P(Z|Y)}{P(Y)}$$

$$= P(X|Y)P(Z|Y)$$

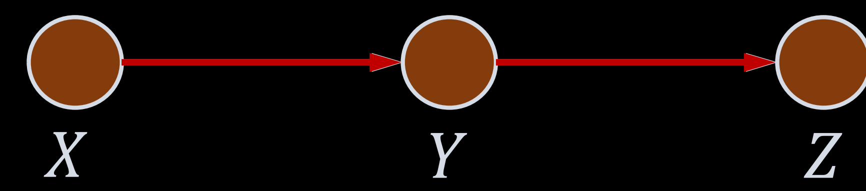
Try to prove the same for forks!!!





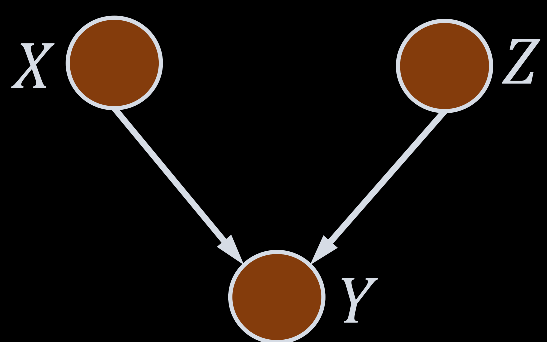
(STRICT) CAUSAL EDGES ASSUMPTION

In a DAG, every parent is a direct cause of all its children.



Flow of association is symmetric.

Association flows along any path that does not contain an **IMMORALITY** (**COLLIDER**).



Flow of causation is not symmetric.

Causation flows in a single direction, only along directed paths.

A **COLLIDER** is a node with two or more parents.

In a **COLLIDER** X and Z are independent, i.e., $X \perp\!\!\!\perp Z$

How could it be that X and Z are associated?

- X isn't the descendant of Z, and Z isn't the descendant of X, like in chains.
- X and Z don't share a common cause, like in forks.

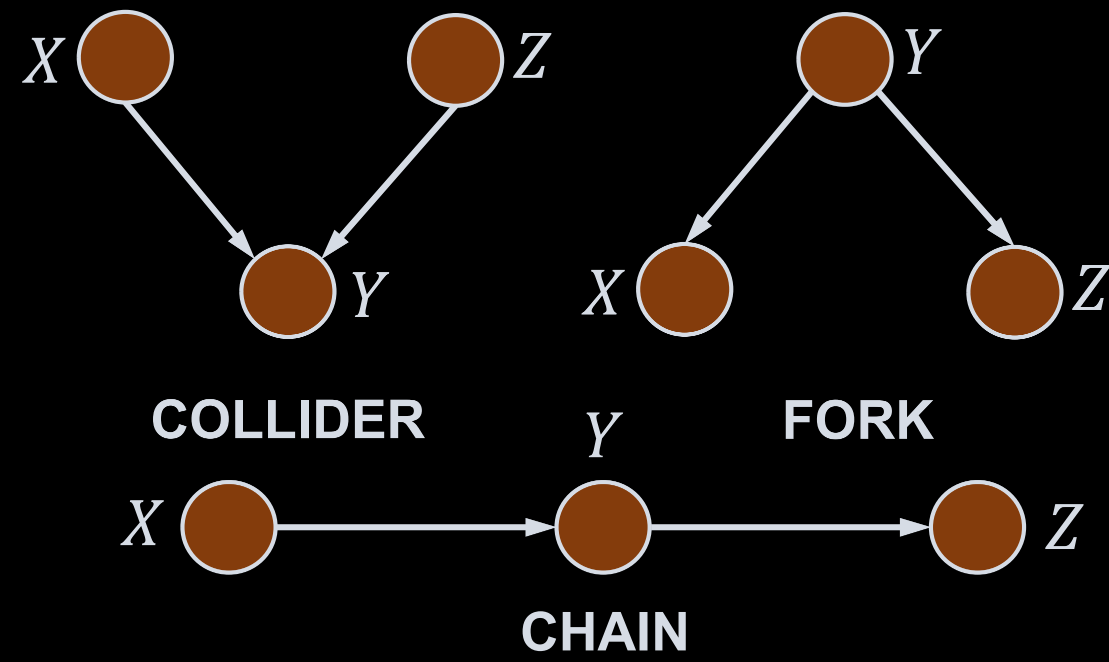
We can think of X and Z simply as unrelated events that can happen, and which both contribute to some common effect (Y).

To show that $X \perp\!\!\!\perp Z$ we apply the Bayesian network factorization and then marginalize out Y.

BAYESIAN NETWORK FACTORIZATION

Given a probability distribution P and a DAG $\mathcal{G}$, P factorizes according to $\mathcal{G}$ if

$$P(X_1, X_2, \dots, X_n) = \prod_{i=1}^{n} P\big(X_i | pa(X_i)\big)$$



$$\begin{aligned} P(X,Z) &= \sum_y P(X, Y=y, Z) \\ &= \sum_y P(X)P(Z)P(Y=y|X,Z) \\ &= P(X)P(Z)\sum_y P(Y=y|X,Z) \\ &= P(X)P(Z) \end{aligned}$$

A **COLLIDER** is a node with two or more parents.

In a **COLLIDER** X and Z are independent, i.e., $X \perp\!\!\!\perp Z$

How could it be that X and Z are associated?

- X isn't the descendant of Z, and Z isn't the descendant of X, like in chains.
- X and Z don't share a common cause, like in forks.

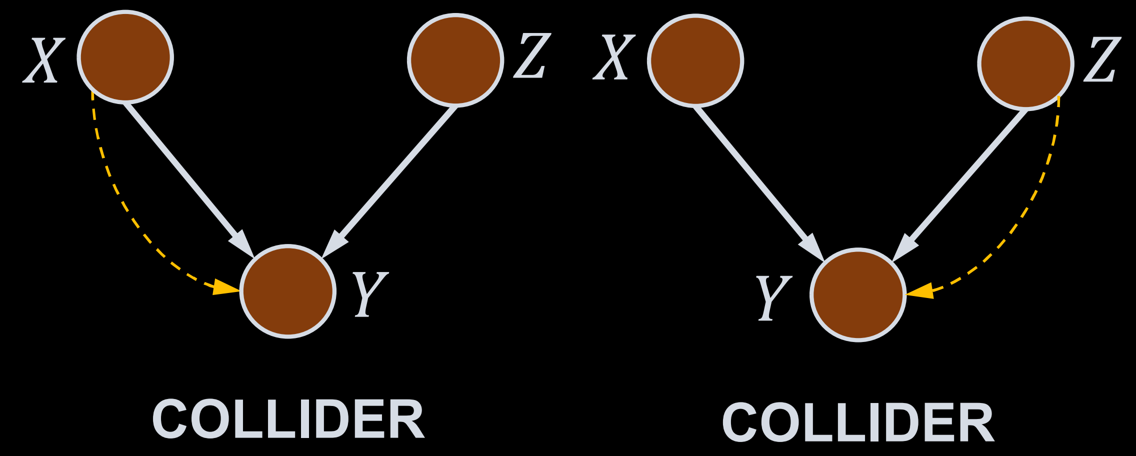
We can think of X and Z simply as unrelated events that can happen, and which both contribute to some common effect (Y).

To show that $X \perp\!\!\!\perp Z$ we apply the Bayesian network factorization and then marginalize out Y.

BAYESIAN NETWORK FACTORIZATION

Given a probability distribution P and a DAG $\mathcal{G}$, P factorizes according to $\mathcal{G}$ if

$$P(X_1, X_2, \dots, X_n) = \prod_{i=1}^{n} P\big(X_i | pa(X_i)\big)$$



The flow of information from node X to node Z is blocked by node Y.

The collider Y blocks the path from node X to node Z and blocks the path from node Z to node X.

This is another example of a **BLOCKED PATH**, but this time the path is not blocked by conditioning; the path is **UN-BLOCKED** by conditioning (on a collider).

Consider a simultaneous (independent) toss of two fair coins X and Z and a bell Y that rings whenever at least one of the coin lands on heads.

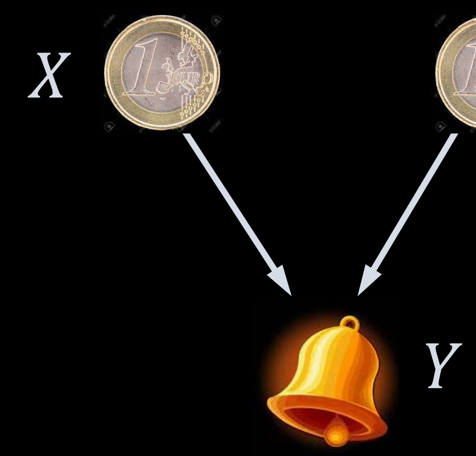
We know that the two coins X and Z are fair, thus each of them has probability equal to 0.5 to turn out *head*, when tossed.

If we do not know whether the bell Y rings or not, the two coins, X and Z, are independent, i.e., when we know that the coin X turned out to be *head*, the coin Z still has probability equal to 0.5 to turn out head, when tossed.

The same applies when we know that the coin X turned out to be *tail*.

The same applies when we swap what we know about coin X with what we know about coin Z.

What happens when we know that the bell Y rings?



$X, Z \in \{head, tail\}$

$Y \in \{silence, rings\}$

Figure 3.5

WHAT WE KNOW	WHAT WE INFER?
$X = head$	$Z =?$
$X = tail$	$Z =?$
$Z = head$	$X =?$
$Z = tail$	$X =?$
$Y = rings, X = tail$	$Z = head$
$Y = rings, Z = tail$	$X = head$
$Y = silence$	$X = Z = tail$

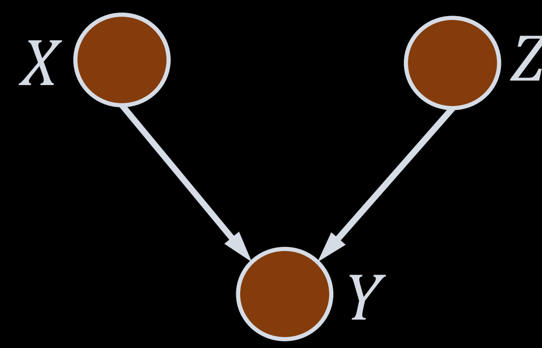
To summarize, conditioning on a collider can turn a blocked path into an **UN-BLOCKED PATH**.

The parents X and Z are not associated (no association flow) in the general population, but when we condition on their shared child Y taking on a specific value y ($Y = y$), they become associated.

This is sometimes referred to as the **SELECTION BIAS**.

- Not conditioning on the collider Y ($Y =?$) blocks association to flow along the path $X \longrightarrow Y \longleftarrow Z$.
- Conditioning on the collider Y ($Y = y$) allows association to flow along the path $X \longrightarrow Y \longleftarrow Z$.

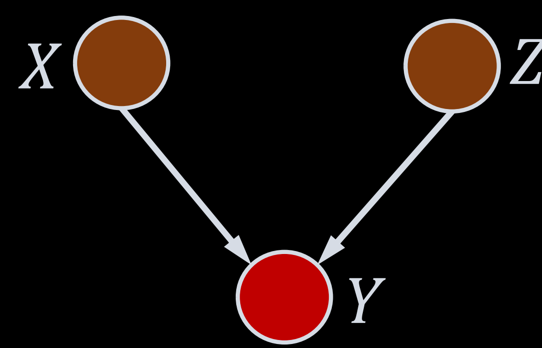
What we have discussed in these last slides was based on our intuition, but what about some quantitative example to better understand what we just stated?



$X \longrightarrow Y \longleftarrow Z$

COLLIDER
not conditioning on Y
($Y =?$)

BLOCKED PATH



$X \longrightarrow Y \longleftarrow Z$

COLLIDER
conditioning on Y
($Y = y$)

UN-BLOCKED PATH

Conditioning on **DESCENDANTS OF A COLLIDER** also induces association in between the parents of the collider.

The intuition is that if we learn something about a collider's descendant, we usually also learn something about the collider itself because there is a direct causal path from the collider to its descendants, and we know that nodes in a chain are usually associated assuming **MINIMALITY**.

MINIMALITY ASSUMPTION

> Given its parents $pa(X)$ in the DAG $\mathcal{G}$, a node X is independent of all its non-descendants (local Markov assumption).
>
> Adjacent nodes in the DAG $\mathcal{G}$ are dependent.

A descendant of a collider works as a proxy for that collider, so conditioning on one of the descendants of a collider is similar to conditioning on the collider itself.

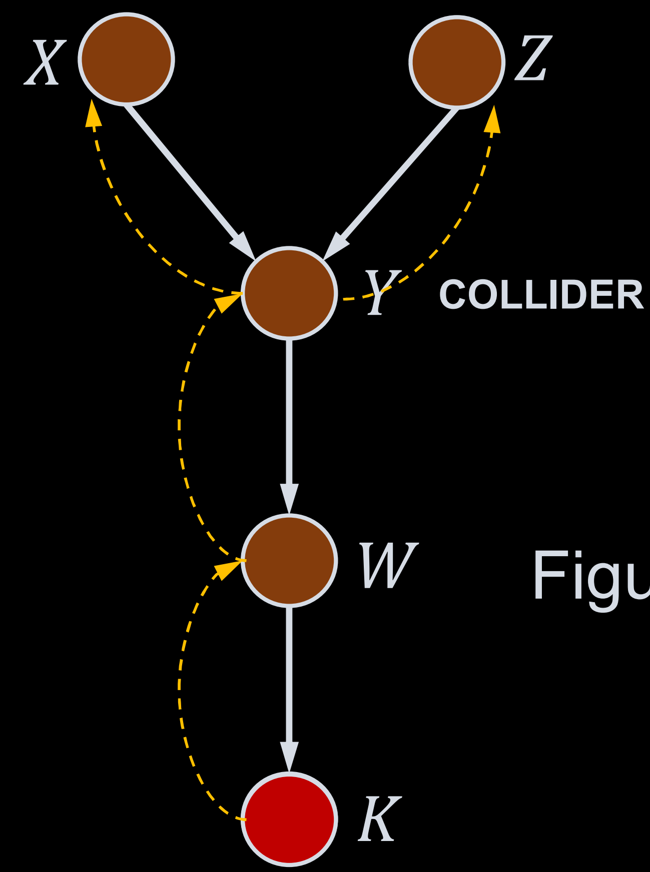
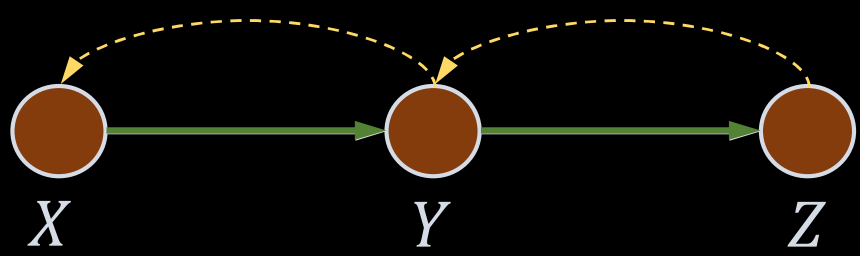


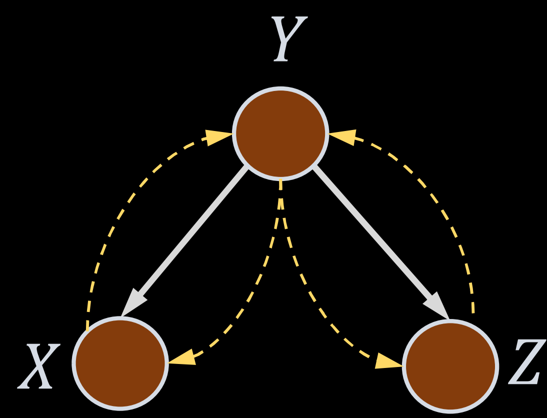
Figure 3.6

When we learn something about the collider's Y descendant K, we also learn something about the collider itself Y because there is a direct causal path $Y \longrightarrow W \longrightarrow K$ from the collider Y to its descendant K, and nodes in a chain are usually associated assuming minimality.

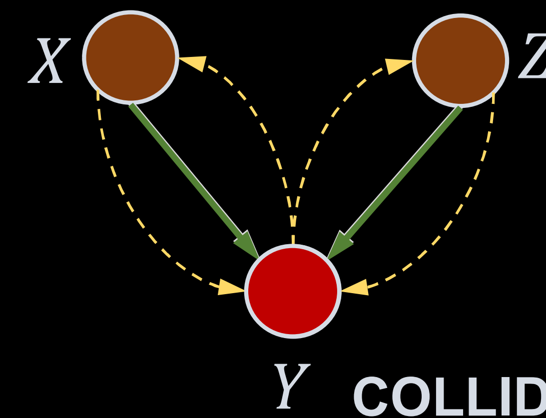
Dashed lines represent association flow. Continuous lines represent causal flow.



CHAIN



FORK



COLLIDER

CHAIN:

$X \not\!\perp\!\!\!\perp Y \quad X \not\!\perp\!\!\!\perp Z \quad Y \not\!\perp\!\!\!\perp Z$

$X \perp\!\!\!\perp Z \mid Y \quad Z \perp\!\!\!\perp X \mid Y$

FORK:

$X \not\!\perp\!\!\!\perp Y \quad X \not\!\perp\!\!\!\perp Z \quad Y \not\!\perp\!\!\!\perp Z$

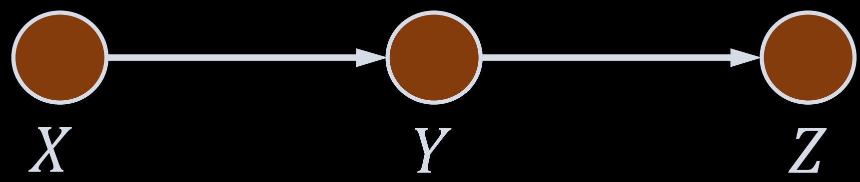
$X \perp\!\!\!\perp Z \mid Y \quad Z \perp\!\!\!\perp X \mid Y$

COLLIDER:

$X \perp\!\!\!\perp Z \quad Z \perp\!\!\!\perp X$

$X \not\!\perp\!\!\!\perp Z \mid Y \quad Z \not\!\perp\!\!\!\perp X \mid Y$

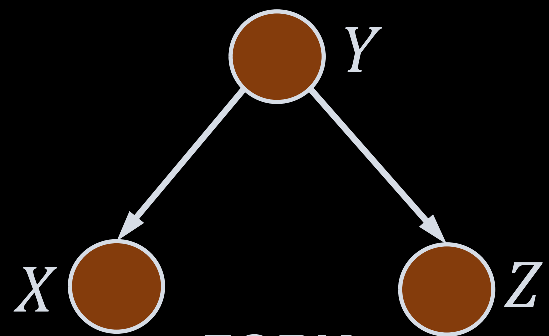
$X \perp\!\!\!\perp Z$	X is independent on Z	$X \perp\!\!\!\perp Z \mid Y$	X is independent on Z given Y
$X \not\!\perp\!\!\!\perp Z$	X is not independent on Z	$X \not\!\perp\!\!\!\perp Z \mid Y$	X is not independent on Z given Y



CHAIN

CONDITIONAL INDEPENDENCE IN CHAINS

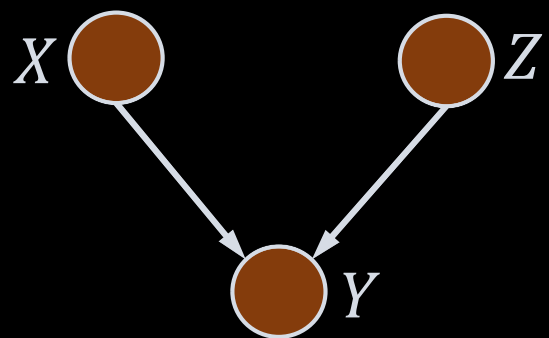
Two variables, X and Z, are conditionally independent given $\mathbf{Y}$, if there is only one directed path between X and Z, and $\mathbf{Y}$ is any set of variables that intercepts that path.



FORK

CONDITIONAL INDEPENDENCE IN FORKS

If a variable Y is a common cause of variables X and Z, and there is only one path between X and Z, then X and Z are independent conditional on Y.



COLLIDER

CONDITIONAL INDEPENDENCE IN COLLIDERS

If a variable Y is the collision node between two variables X and Z, and there is only one path between X and Z, then X and Z are unconditionally independent but are dependent conditional on Y and any descendants of Y.

We now give one of the most relevant definitions of this topic, i.e., we introduce the concept of "**D-SEPARATION**".

D-SEPARATION

A path p is blocked by a set of nodes $\mathbf{S}$ if and only if

p contains a chain of nodes $A \longrightarrow B \longrightarrow C$ or a fork $A \longleftarrow B \longrightarrow C$ such that the middle node B is in $\mathbf{S}$ (i.e., is conditioned on),

or p contains a collider $A \longrightarrow B \longleftarrow C$ such that the collision node B is not in $\mathbf{S}$, and no descendant of B is in $\mathbf{S}$.

If $\mathbf{S}$ blocks every path between two nodes X and Y, then X and Y are d-separated, conditional on $\mathbf{S}$, and thus are independent conditional on $\mathbf{S}$.

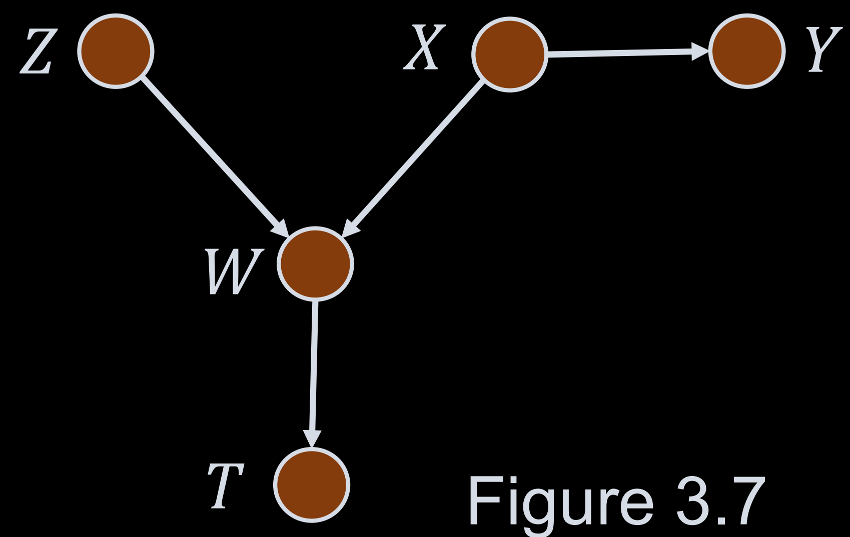


Figure 3.7

The variables might be discrete, continuous, or a mixture of the two; the relationships between them might be linear, exponential, or any of an infinite number of other relations.

No matter the model, however, **D-SEPARATION** will always provide the same set of independencies in the data the model generates.

We now give one of the most relevant definitions of this topic, i.e., we introduce the concept of "**D-SEPARATION**".

D-SEPARATION

A path p is blocked by a set of nodes $\mathbf{S}$ if and only if

p contains a chain of nodes $A \longrightarrow B \longrightarrow C$ or a fork $A \longleftarrow B \longrightarrow C$ such that the middle node B is in $\mathbf{S}$ (i.e., is conditioned on),

or p contains a collider $A \longrightarrow B \longleftarrow C$ such that the collision node B is not in $\mathbf{S}$, and no descendant of B is in $\mathbf{S}$.

If $\mathbf{S}$ blocks every path between two nodes X and Y, then X and Y are d-separated, conditional on $\mathbf{S}$, and thus are independent conditional on $\mathbf{S}$.

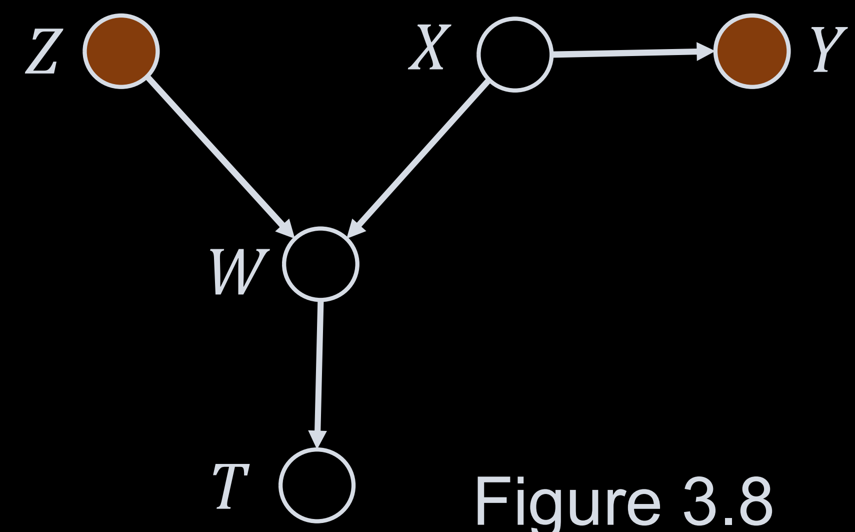


Figure 3.8

In particular, let's look at the relationship between Z and Y.

empty **CONDITIONING SET** $\mathbf{S} = \{\varnothing\}$ → Z and Y are d-separated

Because there is no un-blocked path between Z and Y.

We now give one of the most relevant definitions of this topic, i.e., we introduce the concept of “**D-SEPARATION**”.

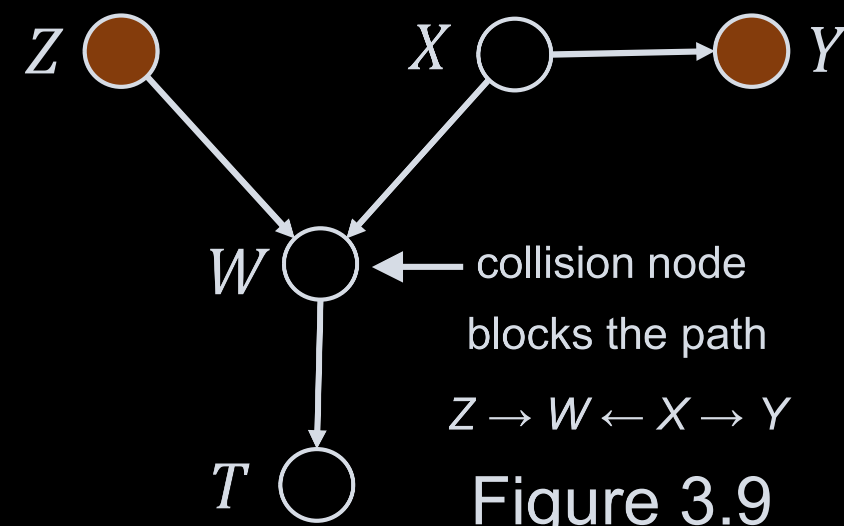
D-SEPARATION

A path p is blocked by a set of nodes $\mathbf{S}$ if and only if

p contains a chain of nodes $A \longrightarrow B \longrightarrow C$ or a fork $A \longleftarrow B \longrightarrow C$ such that the middle node B is in $\mathbf{S}$ (i.e., is conditioned on),

or p contains a collider $A \longrightarrow B \longleftarrow C$ such that the collision node B is not in $\mathbf{S}$, and no descendant of B is in $\mathbf{S}$.

If $\mathbf{S}$ blocks every path between two nodes X and Y, then X and Y are d-separated, conditional on $\mathbf{S}$, and thus are independent conditional on $\mathbf{S}$.



$Z \rightarrow W \leftarrow X \rightarrow Y$

Figure 3.9

In particular, let’s look at the relationship between Z and Y.

empty **CONDITIONING SET** $\mathbf{S} = \{\emptyset\}$ → Z and Y are d-separated

The only path connecting Z to Y, $Z \longrightarrow W \longleftarrow X \longrightarrow Y$, is blocked by the collider W.

Because there is no un-blocked path between Z and Y.

We now give one of the most relevant definitions of this topic, i.e., we introduce the concept of "**D-SEPARATION**".

D-SEPARATION

A path p is blocked by a set of nodes $\mathbf{S}$ if and only if

p contains a chain of nodes $A \longrightarrow B \longrightarrow C$ or a fork $A \longleftarrow B \longrightarrow C$ such that the middle node B is in $\mathbf{S}$ (i.e., is conditioned on),

or p contains a collider $A \longrightarrow B \longleftarrow C$ such that the collision node B is not in $\mathbf{S}$, and no descendant of B is in $\mathbf{S}$.

If $\mathbf{S}$ blocks every path between two nodes X and Y, then X and Y are d-separated, conditional on $\mathbf{S}$, and thus are independent conditional on $\mathbf{S}$.

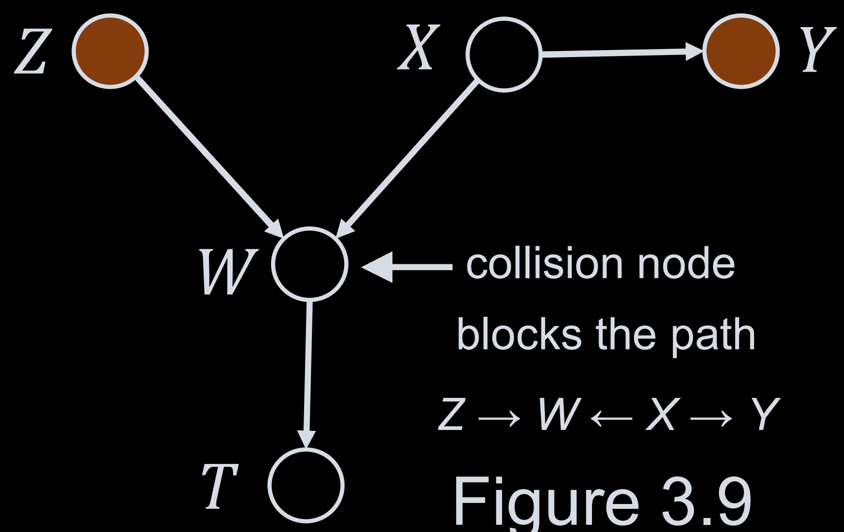


Figure 3.9

In particular, let's look at the relationship between Z and Y.

The only path connecting Z to Y, $Z \longrightarrow W \longleftarrow X \longrightarrow Y$, is blocked by the collider W.

empty **CONDITIONING SET** $\mathbf{S} = \{\varnothing\}$ → Z and Y are d-separated → Z and Y are unconditionally independent

Because there is no un-blocked path between Z and Y.

We now give one of the most relevant definitions of this topic, i.e., we introduce the concept of "**D-SEPARATION**".

D-SEPARATION

A path p is blocked by a set of nodes $\mathbf{S}$ if and only if

- p contains a chain of nodes $A \longrightarrow B \longrightarrow C$ or a fork $A \longleftarrow B \longrightarrow C$ such that the middle node B is in $\mathbf{S}$ (i.e., is conditioned on),
- or p contains a collider $A \longrightarrow B \longleftarrow C$ such that the collision node B is not in $\mathbf{S}$, and no descendant of B is in $\mathbf{S}$.

If $\mathbf{S}$ blocks every path between two nodes X and Y, then X and Y are d-separated, conditional on $\mathbf{S}$, and thus are independent conditional on $\mathbf{S}$.

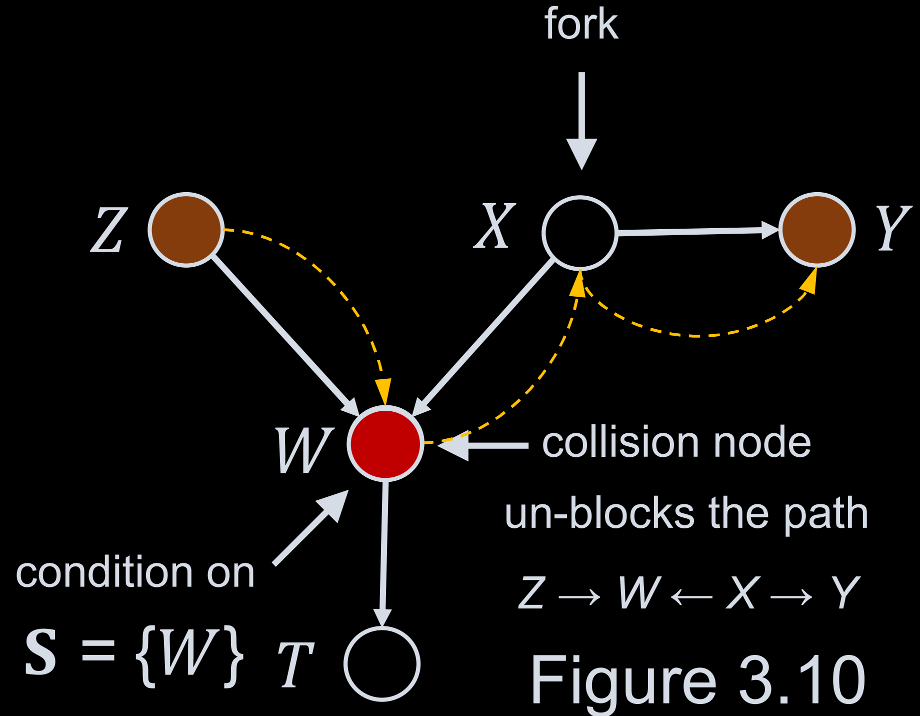


Figure 3.10

In particular, let's look at the relationship between Z and Y.

CONDITIONING SET $\mathbf{S} = \{W\}$ → Z and Y are not d-separated → Z and Y are conditionally dependent

The only path connecting Z to Y, $Z \longrightarrow W \longleftarrow X \longrightarrow Y$, is un-blocked by conditioning on the collider W.

We now give one of the most relevant definitions of this topic, i.e., we introduce the concept of "**D-SEPARATION**".

D-SEPARATION

A path p is blocked by a set of nodes $\mathbf{S}$ if and only if

- p contains a chain of nodes $A \longrightarrow B \longrightarrow C$ or a fork $A \longleftarrow B \longrightarrow C$ such that the middle node B is in $\mathbf{S}$ (i.e., is conditioned on),
- or p contains a collider $A \longrightarrow B \longleftarrow C$ such that the collision node B is not in $\mathbf{S}$, and no descendant of B is in $\mathbf{S}$.

If $\mathbf{S}$ blocks every path between two nodes X and Y, then X and Y are d-separated, conditional on $\mathbf{S}$, and thus are independent conditional on $\mathbf{S}$.

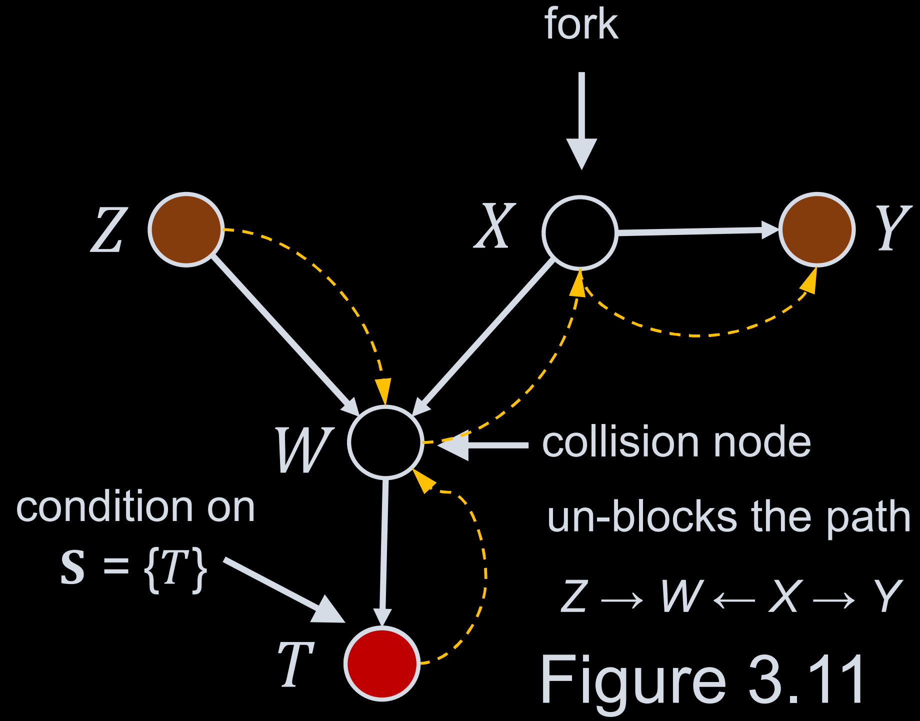


Figure 3.11

In particular, let's look at the relationship between Z and Y.

CONDITIONING SET $\mathbf{S} = \{T\}$ ⟶ Z and Y are not d-separated ⟶ Z and Y are conditionally dependent

The only path connecting Z to Y, $Z \longrightarrow W \longleftarrow X \longrightarrow Y$, is un-blocked by conditioning on the descendant T of the collider W.

We now give one of the most relevant definitions of this topic, i.e., we introduce the concept of "**D-SEPARATION**".

D-SEPARATION

A path p is blocked by a set of nodes $\mathbf{S}$ if and only if

p contains a chain of nodes $A \longrightarrow B \longrightarrow C$ or a fork $A \longleftarrow B \longrightarrow C$ such that the middle node B is in $\mathbf{S}$ (i.e., is conditioned on),

or p contains a collider $A \longrightarrow B \longleftarrow C$ such that the collision node B is not in $\mathbf{S}$, and no descendant of B is in $\mathbf{S}$.

If $\mathbf{S}$ blocks every path between two nodes X and Y, then X and Y are d-separated, conditional on $\mathbf{S}$, and thus are independent conditional on $\mathbf{S}$.

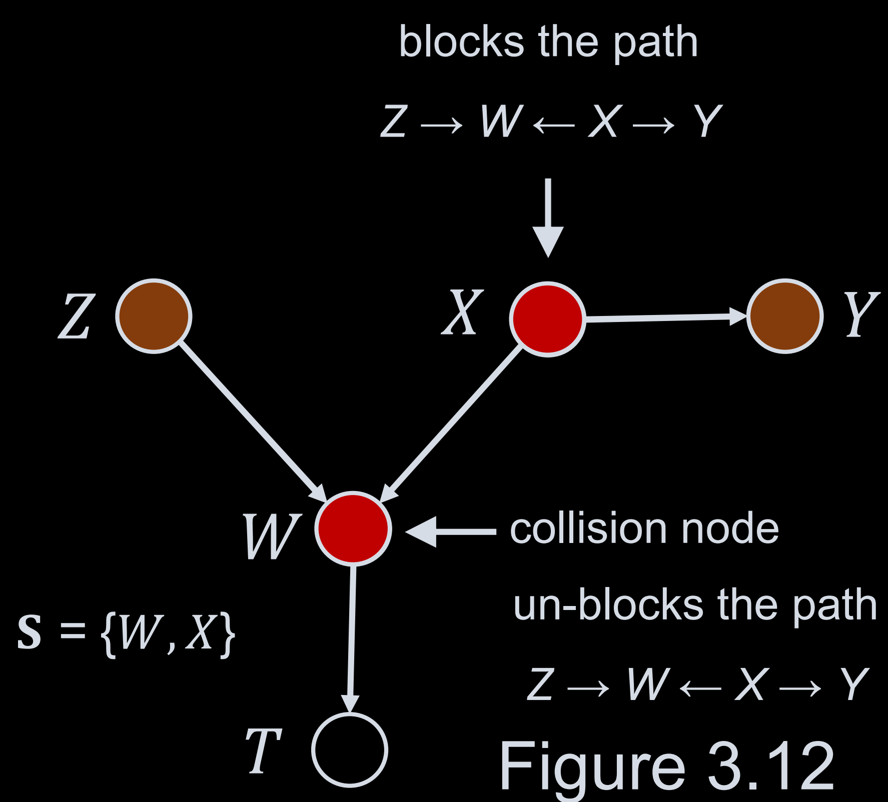


Figure 3.12

In particular, let's look at the relationship between Z and Y.

CONDITIONING SET

$\mathbf{S} = \{W, X\}$ ⟶ Z and Y are d-separated ⟶ Z and Y are conditionally independent

The only path connecting Z to Y, $Z \longrightarrow W \longleftarrow X \longrightarrow Y$, is un-blocked by conditioning on the collider W and then is blocked by conditioning on the fork X.

We now give one of the most relevant definitions of this topic, i.e., we introduce the concept of "**D-SEPARATION**".

D-SEPARATION

A path p is blocked by a set of nodes $\mathbf{S}$ if and only if

- p contains a chain of nodes $A \longrightarrow B \longrightarrow C$ or a fork $A \longleftarrow B \longrightarrow C$ such that the middle node B is in $\mathbf{S}$ (i.e., is conditioned on),
- or p contains a collider $A \longrightarrow B \longleftarrow C$ such that the collision node B is not in $\mathbf{S}$, and no descendant of B is in $\mathbf{S}$.

If $\mathbf{S}$ blocks every path between two nodes X and Y, then X and Y are d-separated, conditional on $\mathbf{S}$, and thus are independent conditional on $\mathbf{S}$.

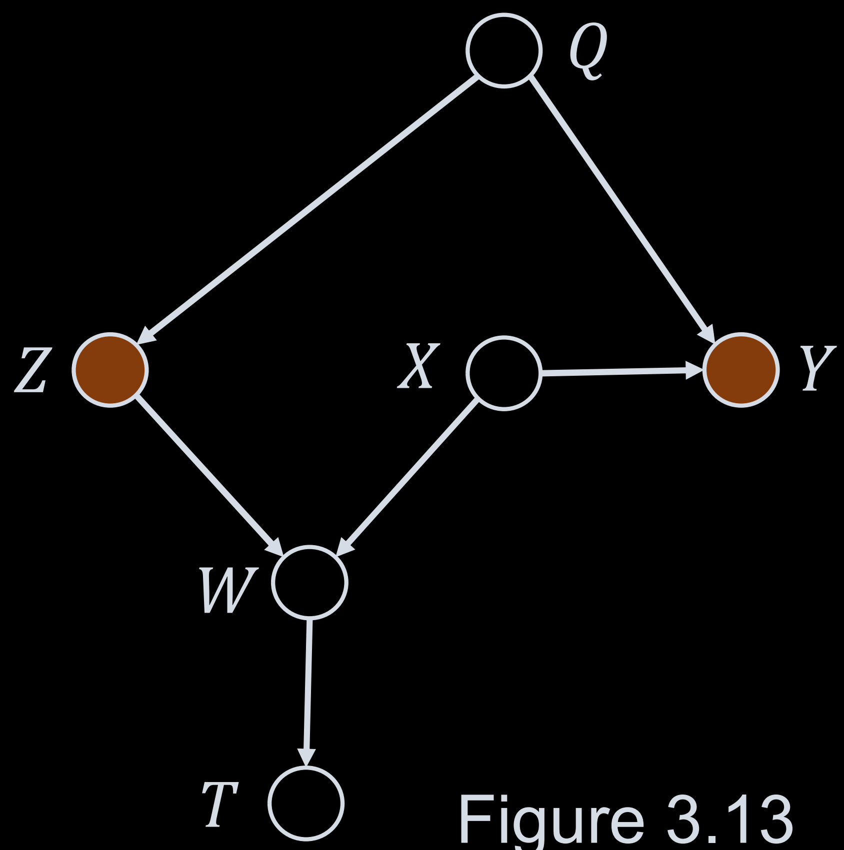


Figure 3.13

Let's look at the relationship between Z and Y, in this new graph.

empty **CONDITIONING SET** $\mathbf{S} = \{\varnothing\}$ → Z and Y are not d-separated → Z and Y are unconditionally dependent

The path connecting Z to Y, $Z \longrightarrow W \longleftarrow X \longrightarrow Y$, is blocked, while the path connecting Z to Y $Z \longleftarrow Q \longrightarrow Y$ is un-blocked.

We now give one of the most relevant definitions of this topic, i.e., we introduce the concept of "**D-SEPARATION**".

D-SEPARATION

A path p is blocked by a set of nodes $\mathbf{S}$ if and only if

p contains a chain of nodes $A \longrightarrow B \longrightarrow C$ or a fork $A \longleftarrow B \longrightarrow C$ such that the middle node B is in $\mathbf{S}$ (i.e., is conditioned on),

or p contains a collider $A \longrightarrow B \longleftarrow C$ such that the collision node B is not in $\mathbf{S}$, and no descendant of B is in $\mathbf{S}$.

If $\mathbf{S}$ blocks every path between two nodes X and Y, then X and Y are d-separated, conditional on $\mathbf{S}$, and thus are independent conditional on $\mathbf{S}$.

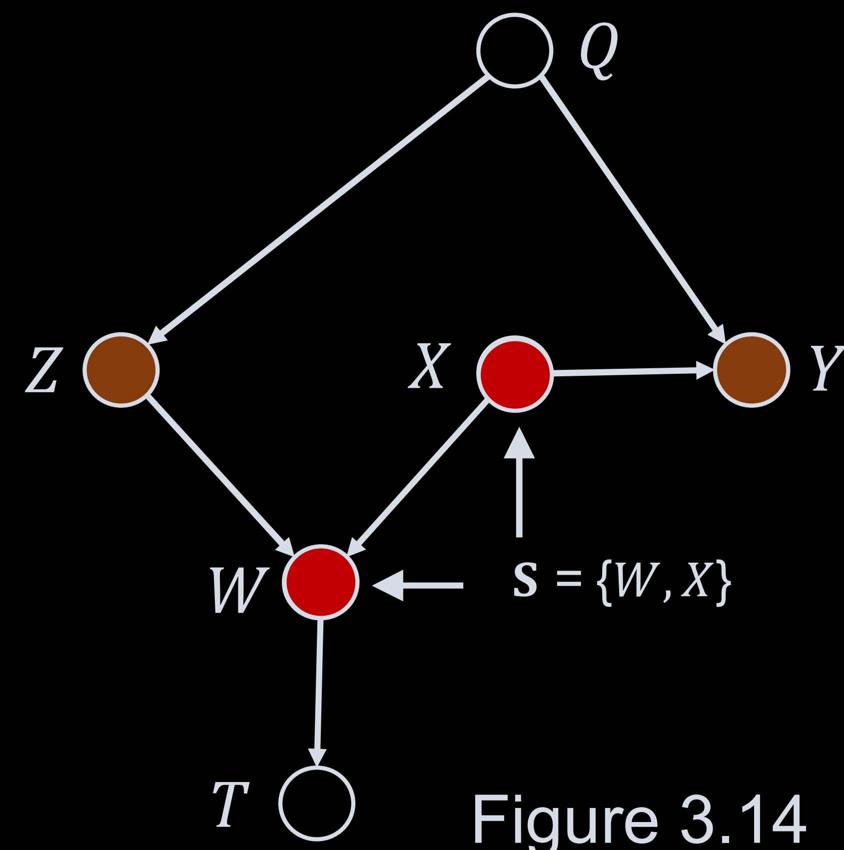


Figure 3.14

Let's look at the relationship between Z and Y, in this new graph.

CONDITIONING SET $\mathbf{S} = \{W, X\}$ ⟶ Z and Y are not d-separated ⟶ Z and Y are conditionally dependent

The path connecting Z to Y, $Z \longrightarrow W \longleftarrow X \longrightarrow Y$, is blocked, by conditioning on $\mathbf{S} = \{W, X\}$, while the path connecting Z to Y $Z \longleftarrow Q \longrightarrow Y$ is un-blocked.

We now give one of the most relevant definitions of this topic, i.e., we introduce the concept of "**D-SEPARATION**".

D-SEPARATION

A path p is blocked by a set of nodes $\mathbf{S}$ if and only if

- p contains a chain of nodes $A \longrightarrow B \longrightarrow C$ or a fork $A \longleftarrow B \longrightarrow C$ such that the middle node B is in $\mathbf{S}$ (i.e., is conditioned on),
- or p contains a collider $A \longrightarrow B \longleftarrow C$ such that the collision node B is not in $\mathbf{S}$, and no descendant of B is in $\mathbf{S}$.

If $\mathbf{S}$ blocks every path between two nodes X and Y, then X and Y are d-separated, conditional on $\mathbf{S}$, and thus are independent conditional on $\mathbf{S}$.

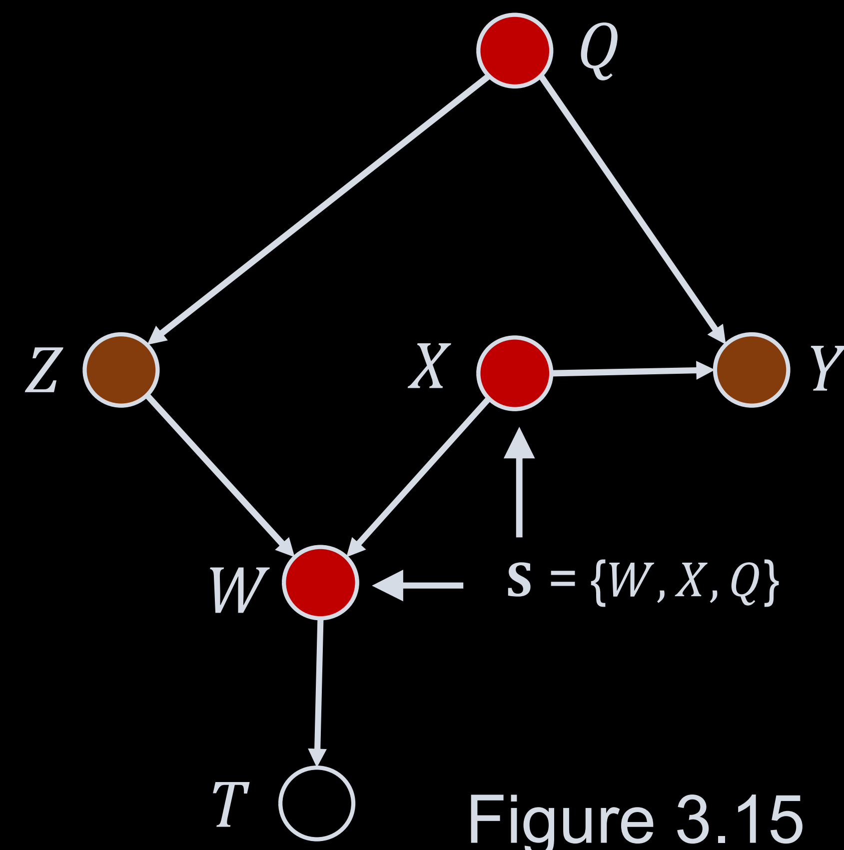


Figure 3.15

Let's look at the relationship between Z and Y, in this new graph.

CONDITIONING SET $\mathbf{S} = \{W, X, Q\}$ → Z and Y are d-separated → Z and Y are conditionally independent

The path connecting Z to Y, $Z \longrightarrow W \longleftarrow X \longrightarrow Y$, is blocked, by conditioning on $\mathbf{S} = \{W, T\}$, and the path connecting Z to Y $Z \longleftarrow Q \longrightarrow Y$ is blocked by conditioning on Q.

We now give one of the most relevant definitions of this topic, i.e., we introduce the concept of "**D-SEPARATION**".

D-SEPARATION

A path p is blocked by a set of nodes $\mathbf{S}$ if and only if

- p contains a chain of nodes $A \longrightarrow B \longrightarrow C$ or a fork $A \longleftarrow B \longrightarrow C$ such that the middle node B is in $\mathbf{S}$ (i.e., is conditioned on),
- or p contains a collider $A \longrightarrow B \longleftarrow C$ such that the collision node B is not in $\mathbf{S}$, and no descendant of B is in $\mathbf{S}$.

If $\mathbf{S}$ blocks every path between two nodes X and Y, then X and Y are d-separated, conditional on $\mathbf{S}$, and thus are independent conditional on $\mathbf{S}$.

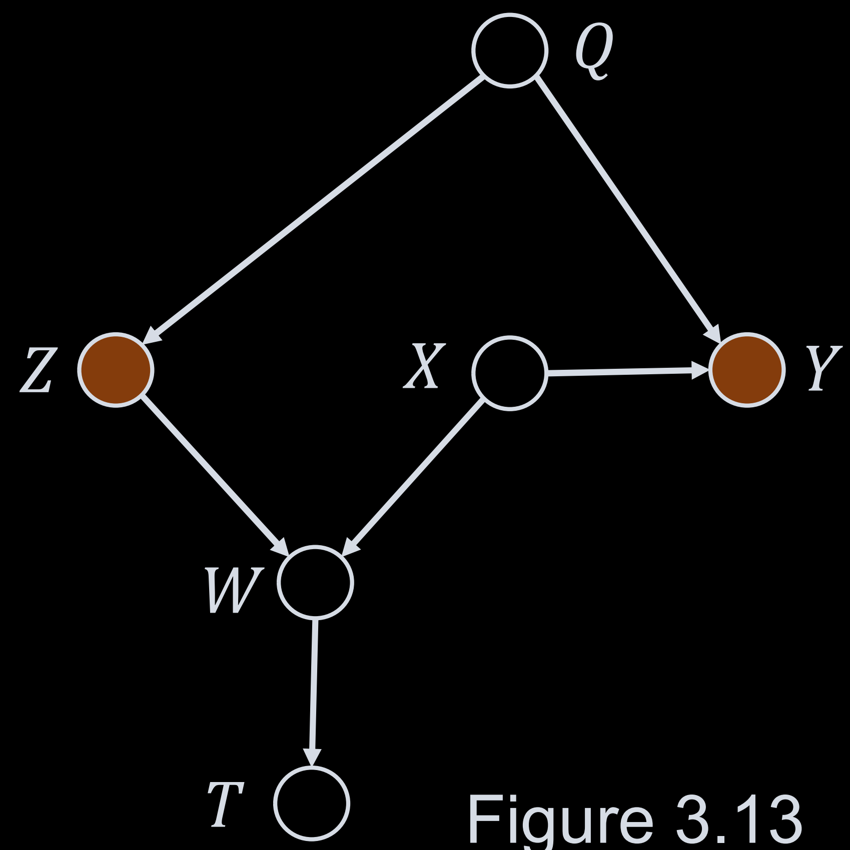


Figure 3.13

In this graph, Z and Y are **D-CONNECTED** conditional on

W, T, $\{W, T\}$, $\{W, Q\}$, $\{T, Q\}$, $\{W, T, Q\}$, $\{W, X\}$, $\{T, X\}$, $\{W, T, X\}$.

We now give one of the most relevant definitions of this topic, i.e., we introduce the concept of "**D-SEPARATION**".

D-SEPARATION

A path p is blocked by a set of nodes $\mathbf{S}$ if and only if

- p contains a chain of nodes $A \longrightarrow B \longrightarrow C$ or a fork $A \longleftarrow B \longrightarrow C$ such that the middle node B is in $\mathbf{S}$ (i.e., is conditioned on),
- or p contains a collider $A \longrightarrow B \longleftarrow C$ such that the collision node B is not in $\mathbf{S}$, and no descendant of B is in $\mathbf{S}$.

If $\mathbf{S}$ blocks every path between two nodes X and Y, then X and Y are d-separated, conditional on $\mathbf{S}$, and thus are independent conditional on $\mathbf{S}$.

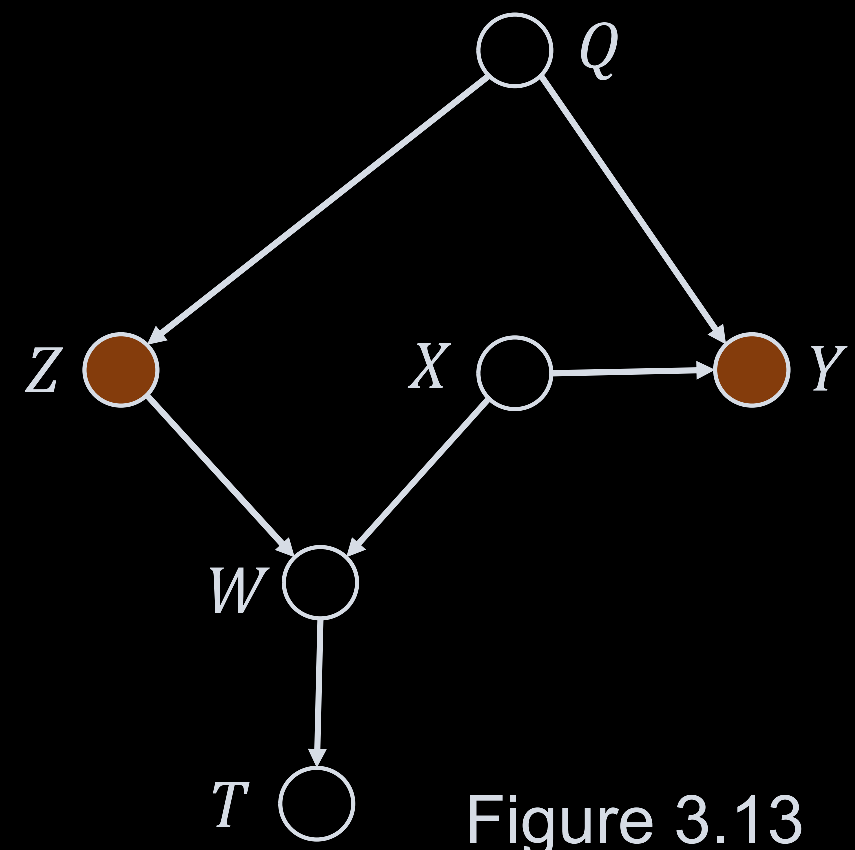


Figure 3.13

Z and Y are **D-SEPARATED** conditional on:

Q, $\{W, X, Q\}$, $\{X, Q\}$, $\{T, X, Q\}$, $\{W, T, X, Q\}$.

Q is in every **CONDITIONING SET** that d-separates Z and Y.

Q is in every conditioning set that **D-SEPARATES** Z and Y because Q is the only node in a path that unconditionally **D-CONNECTS** Z and Y, so unless it is conditioned on, Z and Y will always be **D-CONNECTED**.

D-SEPARATION is an extremely important concept, because it implies **CONDITIONAL INDEPENDENCE**.

$X \perp\!\!\!\perp_{\mathcal{G}} Y \mid \mathbf{S} \rightarrow$ X and Y are **D-SEPARATED** in the graph $\mathcal{G}$ when conditioning on the **CONDITIONING SET $\mathbf{S}$**.

$X \perp\!\!\!\perp_{P} Y \mid \mathbf{S} \rightarrow$ X and Y are **INDEPENDENT** in the distribution P when conditioning on the **CONDITIONING SET $\mathbf{S}$**.

GLOBAL MARKOV ASSUMPTION

Given that P is Markov with respect to $\mathcal{G}$ (satisfies the local Markov assumption), if X and Y are d-separated in $\mathcal{G}$ conditioned on $\mathbf{S}$, then X and Y are independent in P conditioned on $\mathbf{S}$. We can write this succinctly as follows:

$$X \perp\!\!\!\perp_{\mathcal{G}} Y \mid \mathbf{S} \Rightarrow X \perp\!\!\!\perp_{P} Y \mid \mathbf{S}$$

implies

implies

LOCAL MARKOV ASSUMPTION

Given its parents $pa(X)$ in the DAG, a node X is independent of all its non-descendants.

D-SEPARATION is an extremely important concept, because it implies **CONDITIONAL INDEPENDENCE**.

$X \perp\!\!\!\perp_{\mathcal{G}} Y \mid \mathbf{S} \rightarrow$ X and Y are **D-SEPARATED** in the graph $\mathcal{G}$ when conditioning on the **CONDITIONING SET $\boldsymbol{S}$**.

$X \perp\!\!\!\perp_{P} Y \mid \mathbf{S} \rightarrow$ X and Y are **INDEPENDENT** in the distribution P when conditioning on the **CONDITIONING SET $\boldsymbol{S}$**.

BAYESIAN NETWORK FACTORIZATION

Given a probability distribution P and a DAG $\mathcal{G}$, P factorizes according to $\mathcal{G}$ if

$$P(X_1, X_2, \dots, X_n) = \prod_{i=1}^{n} P\big(X_i | pa(X_i)\big)$$

implies

implies

LOCAL MARKOV ASSUMPTION

Given its parents $pa(X)$ in the DAG, a node X is independent of all its non-descendants.

LOCAL MARKOV ASSUMPTION

Given its parents $pa(X)$ in the DAG, a node X is independent of all its non-descendants.

GLOBAL MARKOV ASSUMPTION

Given that P is Markov with respect to $\mathcal{G}$ (satisfies the local Markov assumption), if X and Y are d-separated in $\mathcal{G}$ conditioned on $\mathbf{S}$, then X and Y are independent in P conditioned on $\mathbf{S}$. We can write this succinctly as follows:

$$X \perp\!\!\!\perp_{\mathcal{G}} Y \mid \mathbf{S} \Rightarrow X \perp\!\!\!\perp_{P} Y \mid \mathbf{S}$$

BAYESIAN NETWORK FACTORIZATION

Given a probability distribution P and a DAG $\mathcal{G}$, P factorizes according to $\mathcal{G}$ if

$$P(X_1, X_2, \dots, X_n) = \prod_{i=1}^{n} P\big(X_i | pa(X_i)\big)$$

THEY ARE ALL EQUIVALENT!!!

Therefore, we will use the slightly shortened phrase **MARKOV ASSUMPTION**, to refer to these concepts as a group, or we will simply write "P is Markov with respect to $\mathcal{G}$" to convey the same meaning.

ASSOCIATION and **CAUSATION** flow in directed graphs.

In causal graphs, causation flows along directed paths.

Not only is association not causation, but causation is a sub-category of association, thus association and causation both flow along directed paths.

flow of association along directed paths ⟹ causal association

non-causal association that makes total association not causation ⟹ confounding association

BAYESIAN NETWORKS are purely statistical models, so we can only talk about the flow of association in Bayesian networks.

Association still flows in exactly the same way in **BAYESIAN NETWORKS** as it does in **CAUSAL GRAPHS**, though.

$X \longleftarrow Z \longrightarrow Y$

confounding path

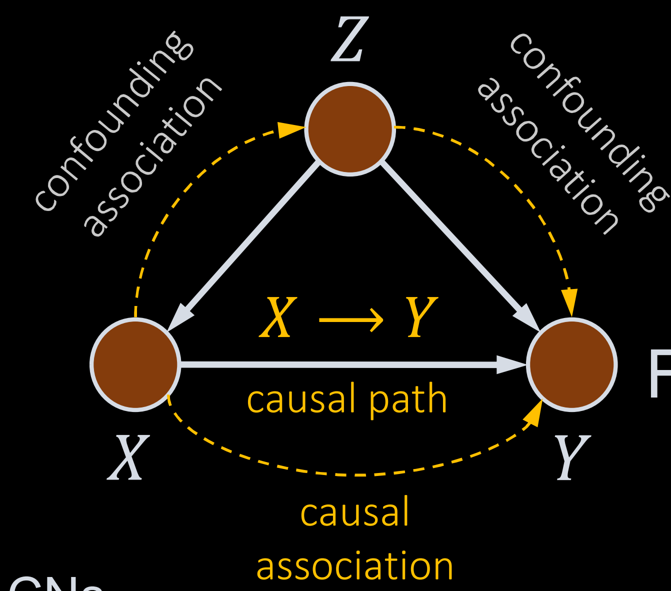


Figure 3.16

In BNs and CNs,

- association flows along chains and forks, unless a node is conditioned on,
- a collider blocks the flow of association, unless it is conditioned on.

We can tell if two nodes are not associated (no association flows between them) by whether or not they are d-separated.

CAUSAL NETWORKS are special in that we additionally assume that the **EDGES HAVE CAUSAL MEANING**

(STRICT) CAUSAL EDGES ASSUMPTION

BAYESIAN NETWORK + In a DAG, every parent is a direct cause of all its children. =

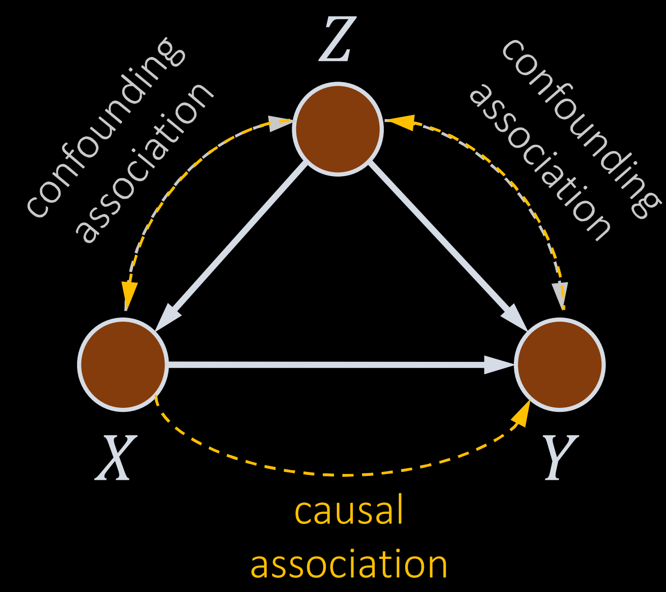
This assumption introduces causality into our models, and it makes one type of path take on a whole new meaning: **DIRECTED PATHS**.

Endows directed paths with the unique role of carrying causation along them.

This assumption is asymmetric!!!

“X is a cause of Y” $\neq$ “Y is a cause of X”.

Figure 3.17



ASSOCIATION IS SYMMETRIC

CAUSATION IS ASYMMETRIC

We have the tools to measure association, how can we isolate causation?

How can we ensure that the association we measure is causation, say, for measuring the causal effect of X on Y?

- We can do that by ensuring that there is no non-causal association flowing between X and Y.
- This is true if X and Y are **D-SEPARATED** in the **AUGMENTED GRAPH** where we remove outgoing edges from X.
- This is because all of X's causal effect on Y would flow through it's outgoing edges, so once those are removed, the only association that remains is purely non-causal association (association path).

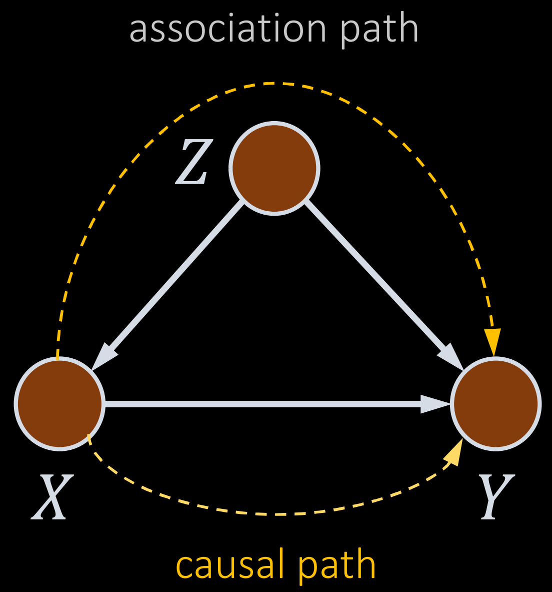


Figure 3.18

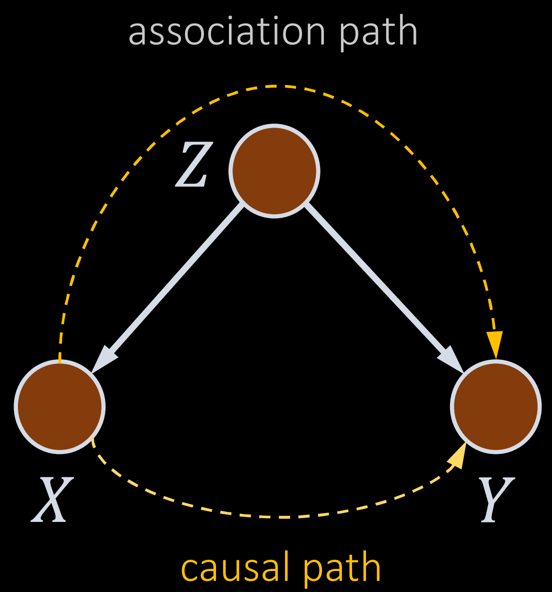


Figure 3.19

RECAP

- **BAYESIAN AND CAUSAL NETWORKS**
- **CHAIN RULE AND FACTORIZATION**
- **CHAIN, FORK AND COLLIDER**
- **D-SEPARATION**