Artificial Intelligence

Causality in Bayesian Networks

Andres Mendez-Vazquez

February 11, 2020

Outline

Outline

Cinvestav

Causality

What do we naturally?

A way of structuring a situation for reasoning under uncertainty is to construct a graph representing causal relations between events.

Example of events with possible outputs

- Fuel? {Yes, No}
- Clean Spark Plugs? {full, 1/2, empty}
- Start? {Yes, No}

Causality

What do we naturally?

A way of structuring a situation for reasoning under uncertainty is to construct a graph representing causal relations between events.

Example of events with possible outputs

- Fuel? {Yes, No}
- Clean Spark Plugs? {full, $1/2$, empty}
- Start? {Yes, No}

Causality

We know

We know that the state of **Fuel?** and the state of **Clean Spark Plugs?** have a causal impact on the state of **Start?**.

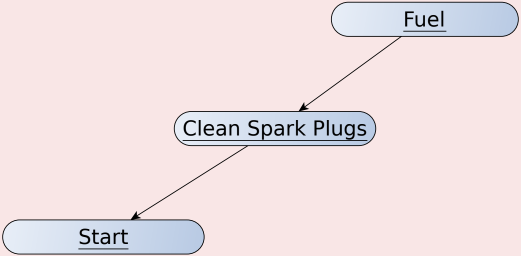
Thus we have something like

Causality

We know

We know that the state of **Fuel?** and the state of **Clean Spark Plugs?** have a causal impact on the state of **Start?**.

Thus we have something like



Outline

Cinvestav

Causal Structure - Judea Perl (1988)

Definition

A causal structure of a set of variables V is a directed acyclic graph (DAG) in which each node corresponds to a distinct element of V, and each edge represents direct functional relationship among the corresponding variables.

Observation

Causal Structure $\approxeq$ A precise specification of how each variable is influenced by its parents in the DAG.

Causal Structure - Judea Perl (1988)

Definition

A causal structure of a set of variables V is a directed acyclic graph (DAG) in which each node corresponds to a distinct element of V, and each edge represents direct functional relationship among the corresponding variables.

Observation

Causal Structure $\approxeq$ A precise specification of how each variable is influenced by its parents in the DAG.

Causal Model

Definition

A causal model is a pair $M = \langle D, \Theta_D \rangle$ consisting of a causal structure D and a set of parameters Θ_D compatible with D.

Causal Model

Definition

A causal model is a pair $M = \langle D, \Theta_D \rangle$ consisting of a causal structure D and a set of parameters Θ_D compatible with D.

Thus

The parameters Θ_D assign a distribution

$$x_i = f_i\left(\mathsf{pa}_i, u_i\right)$$
$$u_i \sim p\left(u_i\right)$$

- pa_i are the parents of x_i in D.
- u_i is independent of any other u.

Causal Model

Definition

A causal model is a pair $M = \langle D, \Theta_D \rangle$ consisting of a causal structure D and a set of parameters Θ_D compatible with D.

Thus

The parameters Θ_D assign a distribution

$$x_i = f_i\left(\mathsf{pa}_i, u_i\right)$$
$$u_i \sim p\left(u_i\right)$$

Where

- x_i is a variable in the model D.
- pa_i are the parents of x_i in D.
- u_i is independent of any other u.

Causal Model

Definition

A causal model is a pair $M = \langle D, \Theta_D \rangle$ consisting of a causal structure D and a set of parameters Θ_D compatible with D.

Thus

The parameters Θ_D assign a distribution

$$x_i = f_i\left(\mathsf{pa}_i, u_i\right)$$
$$u_i \sim p\left(u_i\right)$$

Where

- x_i is a variable in the model D.
- pa_i are the parents of x_i in D.
- u_i is independent of any other u.

Causal Model

Definition

A causal model is a pair $M = \langle D, \Theta_D \rangle$ consisting of a causal structure D and a set of parameters Θ_D compatible with D.

Thus

The parameters Θ_D assign a distribution

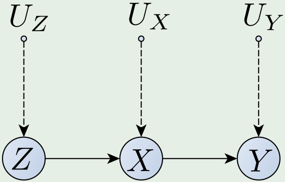
$$x_i = f_i\left(\mathsf{pa}_i, u_i\right)$$
$$u_i \sim p\left(u_i\right)$$

Where

- x_i is a variable in the model D.
- pa_i are the parents of x_i in D.
- u_i is independent of any other u.

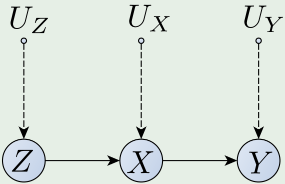
Example

From the point of view of Statistics



Example

From the point of view of Statistics



Formulation

$$z = f_Z\left(u_Z\right)$$

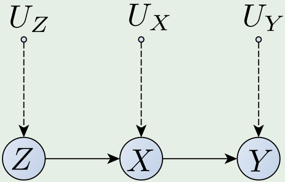
$$x = f_X\left(z, u_X\right)$$

$$y = f_Y\left(x, u_Y\right)$$

Cinvestav

Example

From the point of view of Statistics

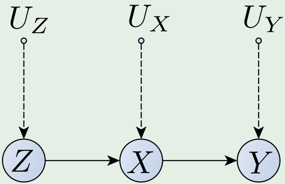


Formulation

$$z = f_Z(u_Z)$$
$$x = f_X(z, u_X)$$
$$y = f_Y(x, u_Y)$$

Example

From the point of view of Statistics



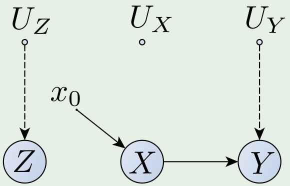
Formulation

$$z = f_Z(u_Z)$$
$$x = f_X(z, u_X)$$
$$y = f_Y(x, u_Y)$$

Cinvestav

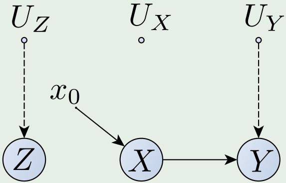
Now add an observation x_0

From the point of view of Statistics



Now add an observation x_0

From the point of view of Statistics



Formulation after blocking information

$$z = f_Z(u_Z)$$

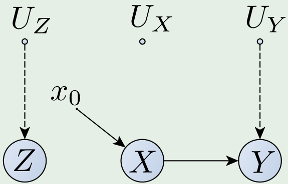
$$x = x_0$$

$$y = f_Y(x, u_Y)$$

Cinvestav

Now add an observation x_0

From the point of view of Statistics



Formulation after blocking information

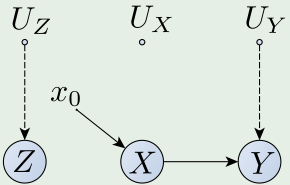
$$z = f_Z(u_Z)$$

$$x = x_0$$

$$y = f_Y(x, u_Y)$$

Now add an observation x_0

From the point of view of Statistics



Formulation after blocking information

$$z = f_Z(u_Z)$$

$$x = x_0$$

$$y = f_Y(x, u_Y)$$

Cinvestav

Outline

Cinvestav

Causal Networks

Definition

A **causal network** consists of a set of variables and a set of directed links (also called edges) between variables.

Causal Networks

Definition

A **causal network** consists of a set of variables and a set of directed links (also called edges) between variables.

Thus

In order to analyze a causal network is necessary to analyze:

- Causal Chains
- Common Causes
- Common Effects

Cinvestav

Causal Networks

Definition

A **causal network** consists of a set of variables and a set of directed links (also called edges) between variables.

Thus

In order to analyze a causal network is necessary to analyze:

- Causal Chains
- Common Causes
- Common Effects

Causal Networks

Definition

A **causal network** consists of a set of variables and a set of directed links (also called edges) between variables.

Thus

In order to analyze a causal network is necessary to analyze:

- Causal Chains
- Common Causes
- Common Effects

Causal Networks

Definition

A **causal network** consists of a set of variables and a set of directed links (also called edges) between variables.

Thus

In order to analyze a causal network is necessary to analyze:

- Causal Chains
- Common Causes
- Common Effects

Causal Networks

Definition

A **causal network** consists of a set of variables and a set of directed links (also called edges) between variables.

Thus

In order to analyze a causal network is necessary to analyze:

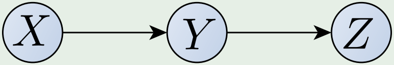
- Causal Chains
- Common Causes
- Common Effects

Outline

Cinvestav

Causal Chains

This configuration is a "causal chain"



X : Low Pressure

Y : Rain

Z : Traffic

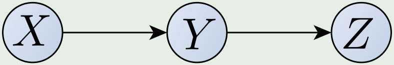
What about the Joint Distribution?

We have by the Chain Rule

$$P(X, Y, Z) = P(X)\, P(Y|X)\, P(Z|Y, X) \quad (1)$$

Causal Chains

This configuration is a "causal chain"



$$X : \text{Low Pressure}$$
$$Y : \text{Rain}$$
$$Z : \text{Traffic}$$

What about the Joint Distribution?

We have by the Chain Rule

$$P(X, Y, Z) = P(X) P(Y|X) P(Z|Y, X) \quad (1)$$

Propagation of Information

Given no information about Y

Information can propagate from X to Z.

Thus

The natural question is What does happen if $Y = y$ for some value y?

Cinvestav

Propagation of Information

Given no information about Y

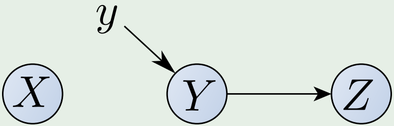
Information can propagate from X to Z.

Thus

The natural question is What does happen if $Y = y$ for some value y?

Thus, we have that

Blocking Propagation of Information



X : Low Pressure

Y : Rain

Z : Traffic

Using Our Probabilities

Then Z is independent of X given a $Y = y$

And making the assumption that once an event happens

$$P(Z|X, Y = y) = P(Z|Y = y) \quad (2)$$

Using Our Probabilities

Then Z is independent of X given a $Y = y$

And making the assumption that once an event happens

$$P(Z|X, Y = y) = P(Z|Y = y) \quad (2)$$

YES!!!

Evidence along the chain "blocks" the influence.

Thus

Something Notable

Knowing that X has occurred does not make any difference to our beliefs about Z if we already know that Y has occurred.

Thus conditional independencies can be written

$$I_P(Z, X|Y = y) \quad (3)$$

Thus

Something Notable

Knowing that X has occurred does not make any difference to our beliefs about Z if we already know that Y has occurred.

Thus conditional independencies can be written

$$I_P(Z, X|Y = y) \tag{3}$$

Therefore

The Joint Probability is equal to

$$P(X, Y = y, Z) = P(X)\, P(Y = y|X)\, P(Z|Y = y) \tag{4}$$

Cinvestav

Thus

Is X independent of Z given $Y = y$?

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

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

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

Cinvestav

Thus

Is X independent of Z given $Y = y$?

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

Thus

Is X independent of Z given $Y = y$?

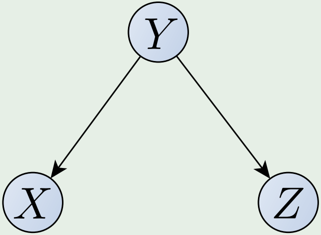
$$
\begin{aligned}
P\left(Z|X, Y=y\right) &= \frac{P\left(X, Y=y, Z\right)}{P\left(X, Y=y\right)} \\
&= \frac{P\left(X\right)P\left(Y=y|X\right)P\left(Z|Y=y\right)}{P\left(X\right)P\left(Y=y|X\right)} \\
&= P\left(Z|Y=y\right)
\end{aligned}
$$

Outline

Common Causes

Another basic configuration: two effects of the same cause

X : John Calls
Y : Alarm
Z : Mary Calls



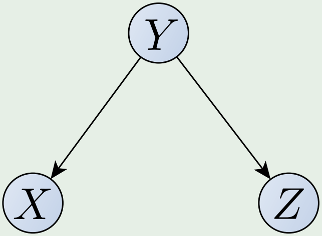
Thus

$$P(X, Y = y, Z) = P(X)\, P(Y = y|X)\, \underbrace{P(Z|X, Y = y)}_{P(Z|Y=y)} \quad (5)$$

Common Causes

Another basic configuration: two effects of the same cause

X : John Calls
Y : Alarm
Z : Mary Calls



Thus

$$P(X, Y = y, Z) = P(X)\, P(Y = y|X) \underbrace{P(Z|X, Y = y)}_{P(Z|Y=y)} \tag{5}$$

Common Causes

What happened if X is independent of Z given $Y = y$?

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

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

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

Evidence on the top of the chain "blocks" the influence between X and Z

Common Causes

What happened if X is independent of Z given $Y = y$?

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

YES!!!

Evidence on the top of the chain "blocks" the influence between X and Z.

Common Causes

What happened if X is independent of Z given $Y = y$?

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

YES!!!

Evidence on the top of the chain "blocks" the influence between X and Z.

Cinvestav

Thus

It gives rise to the same conditional independent structure as chains

$$I_P\left(Z, X | Y = y\right) \tag{6}$$

i.e.

if we already know about Y, then an additional information about X will not tell us anything new about Z.

Thus

It gives rise to the same conditional independent structure as chains

$$I_P(Z, X|Y = y) \tag{6}$$

i.e.

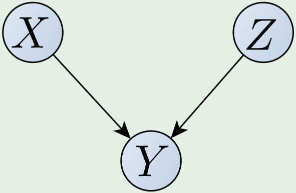
if we already know about Y, then an additional information about X will not tell us anything new about Z.

Outline

Common Effect

Last configuration: two causes of one effect (v-structures)

X : Raining
Y : Traffic
Z : Ballgame



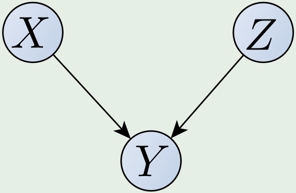
Are X and Z independent if we do not have information about Y?

Yes!!! Because the ballgame and the rain can cause traffic, but they are not correlated.

Common Effect

Last configuration: two causes of one effect (v-structures)

X : Raining
Y : Traffic
Z : Ballgame



Are X and Z independent if we do not have information about Y?

Yes!!! Because the ballgame and the rain can cause traffic, but they are not correlated.

Proof

We have the following

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

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

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

$$= P(Z)$$

Proof

We have the following

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

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

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

$$= P(Z)$$

Proof

We have the following

$$
\begin{aligned}
P\left(Z|X,Y\right) &= \frac{P\left(X,Y,Z\right)}{P\left(X,Y\right)} \\
&= \frac{P\left(X|Z,Y\right)P\left(Y|Z\right)P\left(Z\right)}{P\left(X|Y\right)P\left(Y\right)} \\
&= \frac{P\left(X\right)P\left(Y|Z\right)P\left(Z\right)}{P\left(X\right)P\left(Y|Z\right)} \\
&= P\left(Z\right)
\end{aligned}
$$

Proof

We have the following

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

Cinvestav

Common Effects

Are X and Z independent given $Y = y$?

No!!! Because seeing traffic puts the rain and the ballgame in competition as explanation!!!

$$P\left(X, Z | Y = y\right)$$

Common Effects

Are X and Z independent given $Y = y$?

No!!! Because seeing traffic puts the rain and the ballgame in competition as explanation!!!

Why?

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

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

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

Common Effects

Are X and Z independent given $Y = y$?

No!!! Because seeing traffic puts the rain and the ballgame in competition as explanation!!!

Why?

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

Cinvestav

Common Effects

Are X and Z independent given $Y = y$?

No!!! Because seeing traffic puts the rain and the ballgame in competition as explanation!!!

Why?

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

Cinvestav

The General Case

Backwards from the other cases

- Observing an effect activates influence between possible causes.

Fact

Any complex example can be analyzed using these three canonical cases

Question

In a given Bayesian Network, Are two variables independent (given evidence)?

Solution

- **Analyze Graph Deeply!!!**

The General Case

Backwards from the other cases

- Observing an effect activates influence between possible causes.

Fact

Any complex example can be analyzed using these three canonical cases

Question

In a given Bayesian Network, Are two variables independent (given evidence)?

Solution

- **Analyze Graph Deeply!!!**

The General Case

Backwards from the other cases

- Observing an effect activates influence between possible causes.

Fact

Any complex example can be analyzed using these three canonical cases

Question

In a given Bayesian Network, Are two variables independent (given evidence)?

Solution

- **Analyze Graph Deeply!!!**

The General Case

Backwards from the other cases

- Observing an effect activates influence between possible causes.

Fact

Any complex example can be analyzed using these three canonical cases

Question

In a given Bayesian Network, Are two variables independent (given evidence)?

Solution

- **Analyze Graph Deeply!!!**

Outline

Analyze the Graph

Definition 2.1

- Let $G = (V, E)$ be a DAG, where V is a set of random variables. We say that, based on the Markov condition, G entails conditional independence:
  - $I_P(A, B|C)$ for $A, B, C \subseteq V$ if $I_P(A, B|C)$ holds for every $P \in P_G$,

  where P_G is the set of all probability distributions P such that (G, P) satisfies the Markov condition.

Analyze the Graph

Definition 2.1

- Let $G = (V, E)$ be a DAG, where V is a set of random variables. We say that, based on the Markov condition, G entails conditional independence:
  - $I_P(A, B|C)$ for $A, B, C \subseteq V$ if $I_P(A, B|C)$ holds for every $P \in P_G$,

  where P_G is the set of all probability distributions P such that (G, P) satisfies the Markov condition.

Thus

We also say the Markov condition entails the conditional independence for G and that the conditional independence is in G.

Cinvestav

Analyze the Graph

Definition 2.1

- Let $G = (V, E)$ be a DAG, where V is a set of random variables. We say that, based on the Markov condition, G entails conditional independence:
  - $I_P(A, B|C)$ for $A, B, C \subseteq V$ if $I_P(A, B|C)$ holds for every $P \in P_G$,

  where P_G is the set of all probability distributions P such that (G, P) satisfies the Markov condition.

Thus

We also say the Markov condition entails the conditional independence for G and that the conditional independence is in G.

Cinvestav

Analyze the Graph

Question

In a given Bayesian Network, Are two variables independent (Given evidence)?

Solution

Analyze Graph Deeply!!!

Analyze the Graph

Question

In a given Bayesian Network, Are two variables independent (Given evidence)?

Solution

Analyze Graph Deeply!!!

Outline

Cinvestav

Examples of Entailed Conditional independence

Example

G: Graduate Program Quality.
F: First Job Quality.
B: Number of Publications.
C: Number of Citations.

F is given some evidence!!!

Examples of Entailed Conditional independence

Example

G: Graduate Program Quality.

F: First Job Quality.

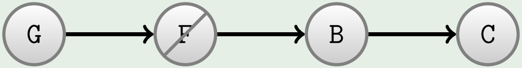
B: Number of Publications.

C: Number of Citations.

F is given some evidence!!!

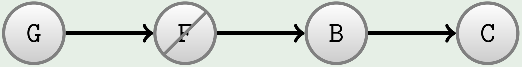
Using Markov Condition

If the graph satisfies the Markov Condition



Using Markov Condition

If the graph satisfies the Markov Condition

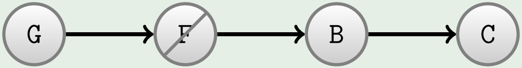


Thus

$$
\begin{aligned}
P(C|G, F=f) &= \sum_b P(C|B=b, G, F=f)\, P(B=b|G, F=f) \\
&= \sum_b P(C|B=b, F=f)\, P(B=b|F=f) \\
&= P(C|F=f)
\end{aligned}
$$

Using Markov Condition

If the graph satisfies the Markov Condition

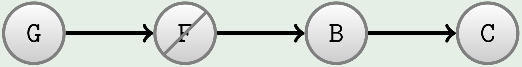


Thus

$$
\begin{aligned}
P(C|G, F = f) &= \sum_{b} P(C|B = b, G, F = f)\, P(B = b|G, F = f) \\
&= \sum_{b} P(C|B = b, F = f)\, P(B = b|F = f) \\
&= P(C|F = f)
\end{aligned}
$$

Using Markov Condition

If the graph satisfies the Markov Condition



Thus

$$\begin{aligned} P(C|G, F = f) &= \sum_{b} P(C|B = b, G, F = f)\, P(B = b|G, F = f) \\ &= \sum_{b} P(C|B = b, F = f)\, P(B = b|F = f) \\ &= P(C|F = f) \end{aligned}$$

Finally

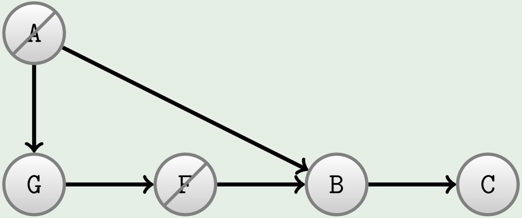
We have

$$I_p\left(C, G|F\right) \tag{7}$$

D-Separation ≈ Conditional independence

F and G are given as evidence

Example C and G are d-separated by A, F in the DAG in



Outline

Cinvestav

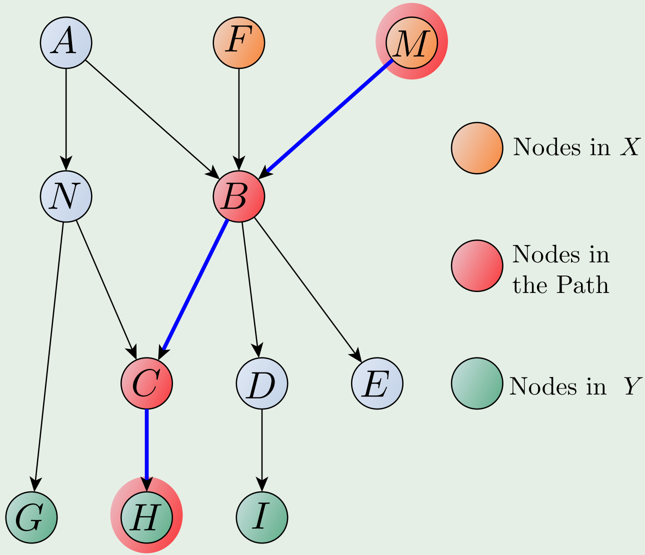
Basic Definitions

Definition (Undirected Paths)

A path between two sets of nodes X and Y is any sequence of nodes between a member of X and a member of Y such that every adjacent pair of nodes is connected by an edge (regardless of direction) and no node appears in the sequence twice.

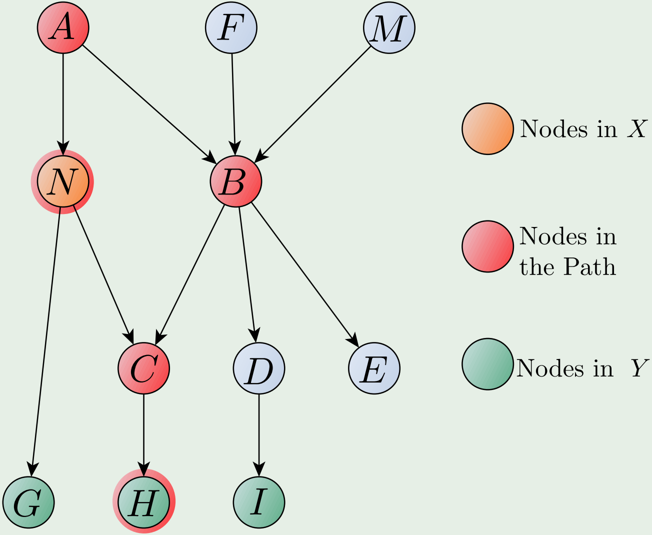
Example

An example



Example

Another one



Thus

Given a path in $G = (V, E)$

There are the edges connecting $[X_1, X_2, ..., X_k]$.

Therefore

Given the directed edge $X \to Y$, we say the tail of the edge is at X and the head of the edge is Y.

Cinvestav

Thus

Given a path in $G = (V, E)$

There are the edges connecting $[X_1, X_2, ..., X_k]$.

Therefore

Given the directed edge $X \rightarrow Y$, we say the tail of the edge is at X and the head of the edge is Y.

Thus

Given a path in $G = (V, E)$

There are the edges connecting $[X_1, X_2, ..., X_k]$.

Therefore

Given the directed edge $X \rightarrow Y$, we say the tail of the edge is at X and the head of the edge is Y.

Basic Classifications of Meetings

Head-to-Tail

A path $X \to Y \to Z$ is a **head-to-tail meeting**, the edges meet head-to-tail at Y, and Y is a head-to-tail node on the path.

Tail-to-Tail

A path $X \leftarrow Y \to Z$ is a **tail-to-tail meeting**, the edges meet tail-to-tail at Z, and Z is a tail-to-tail node on the path.

Head-to-Head

A path $X \to Y \leftarrow Z$ is a **head-to-head meeting**, the edges meet head-to-head at Y, and Y is a head-to-head node on the path.

Basic Classifications of Meetings

Head-to-Tail

A path $X \to Y \to Z$ is a **head-to-tail meeting**, the edges meet head-to-tail at Y, and Y is a head-to-tail node on the path.

Tail-to-Tail

A path $X \leftarrow Y \to Z$ is a **tail-to-tail meeting**, the edges meet tail-to-tail at Z, and Z is a tail-to-tail node on the path.

Head-to-Head

A path $X \to Y \leftarrow Z$ is a **head-to-head meeting**, the edges meet head-to-head at Y, and Y is a head-to-head node on the path.

Cinvestav

Basic Classifications of Meetings

Head-to-Tail

A path $X \to Y \to Z$ is a **head-to-tail meeting**, the edges meet head-to-tail at Y, and Y is a head-to-tail node on the path.

Tail-to-Tail

A path $X \leftarrow Y \to Z$ is a **tail-to-tail meeting**, the edges meet tail-to-tail at Z, and Z is a tail-to-tail node on the path.

Head-to-Head

A path $X \to Y \leftarrow Z$ is a **head-to-head meeting**, the edges meet head-to-head at Y, and Y is a head-to-head node on the path.

Examples

Head-to-Tail

X : Low Pressure

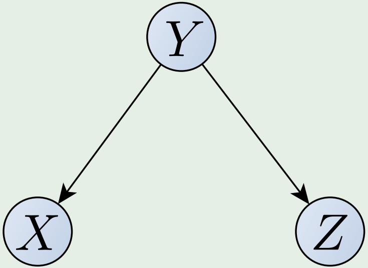
Y : Rain

Z : Traffic

Examples

Tail-to-Tail

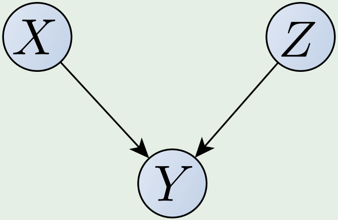
X : John Calls
Y : Alarm
Z : Mary Calls



Examples

Head-to-Head

X : Raining
Y : Traffic
Z : Ballgame



Basic Classifications of Meetings

Finally

- A path (undirected) $X - Z - Y$, such that X and Y are not adjacent, is an uncoupled meeting.

Outline

Cinvestav

Blocking Information ≈ Conditional Independence

Definition 2.2

Definition 2.2 Let$G = (V, E)$ be a DAG, $A \subseteq V$, X and Y be distinct nodes in $V - A$, and ρ be a path between X and Y .

Then ρ is **blocked** by A if one of the following holds:

1. There is a node $Z \in A$ on the path ρ, and the edges incident to Z on ρ meet head-to-tail at Z.
2. There is a node $Z \in A$ on the path ρ, and the edges incident to Z on ρ, meet tail-to-tail at Z.
3. There is a node Z, such that Z and all of Z's descendent's are not in A, on the chain ρ, and the edges incident to Z on ρ meet head-to-head at Z.

Blocking Information ≈ Conditional Independence

Definition 2.2

Definition 2.2 Let$G = (V, E)$ be a DAG, $A \subseteq V$, X and Y be distinct nodes in $V - A$, and ρ be a path between X and Y .
Then ρ is **blocked** by A if one of the following holds:

1. There is a node $Z \in A$ on the path ρ, and the edges incident to Z on ρ meet head-to-tail at Z.
2. There is a node $Z \in A$ on the path ρ, and the edges incident to Z on ρ, meet tail-to-tail at Z.
3. There is a node Z, such that Z and all of Z's descendent's are not in A, on the chain ρ, and the edges incident to Z on ρ meet head-to-head at Z.

Cinvestav

Blocking Information ≈ Conditional Independence

Definition 2.2

Definition 2.2 Let$G = (V, E)$ be a DAG, $A \subseteq V$, X and Y be distinct nodes in $V - A$, and ρ be a path between X and Y .
Then ρ is **blocked** by A if one of the following holds:

1. There is a node $Z \in A$ on the path ρ, and the edges incident to Z on ρ meet head-to-tail at Z.
2. There is a node $Z \in A$ on the path ρ, and the edges incident to Z on ρ, meet tail-to-tail at Z.
3. There is a node Z, such that Z and all of Z's descendent's are not in A, on the chain ρ, and the edges incident to Z on ρ meet head-to-head at Z.

Blocking Information $\approx$ Conditional Independence

Definition 2.2

Definition 2.2 Let$G = (V, E)$ be a DAG, $A \subseteq V$, X and Y be distinct nodes in $V - A$, and ρ be a path between X and Y .
Then ρ is **blocked** by A if one of the following holds:

1. There is a node $Z \in A$ on the path ρ, and the edges incident to Z on ρ meet head-to-tail at Z.
2. There is a node $Z \in A$ on the path ρ, and the edges incident to Z on ρ, meet tail-to-tail at Z.
3. There is a node Z, such that Z and all of Z's descendent's are not in A, on the chain ρ, and the edges incident to Z on ρ meet head-to-head at Z.

Blocking Information ≈ Conditional Independence

Definition 2.2

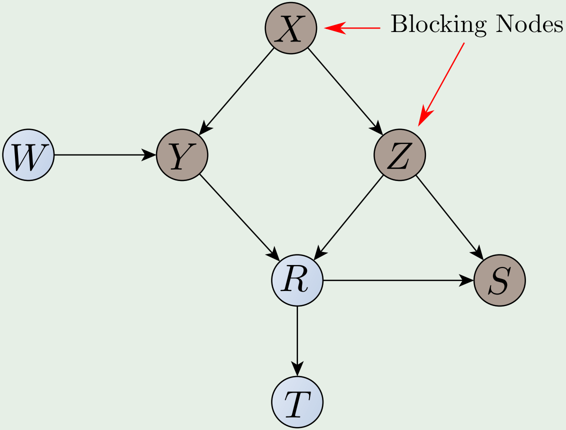
Definition 2.2 Let$G = (V, E)$ be a DAG, $A \subseteq V$, X and Y be distinct nodes in $V - A$, and ρ be a path between X and Y .
Then ρ is **blocked** by A if one of the following holds:

1. There is a node $Z \in A$ on the path ρ, and the edges incident to Z on ρ meet head-to-tail at Z.
2. There is a node $Z \in A$ on the path ρ, and the edges incident to Z on ρ, meet tail-to-tail at Z.
3. There is a node Z, such that Z and all of Z's descendent's are not in A, on the chain ρ, and the edges incident to Z on ρ meet head-to-head at Z.

Example

We have that the path $[Y, X, Z, S]$ is blocked by $\{X\}$ and $\{Z\}$

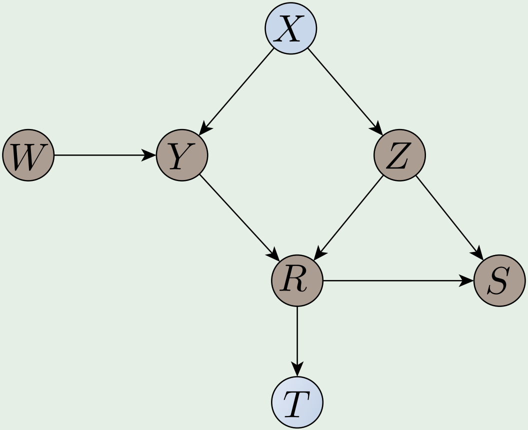
- Because the edges on the chain incident to X meet tail-to-tail at X.



Example

We have that the path $[W, Y, R, Z, S]$ is blocked by $\emptyset$

- Because $R \notin \emptyset$ and $T \notin \emptyset$ and the edges on the chain incident to R meet head-to-head at R .



Outline

Cinvestav

Definition of D-Separation

Definition 2.3

Let $G = (V, E)$ be a DAG, $A \subseteq V$, and X and Y be distinct nodes in $V - A$. We say X and Y are **D-Separated** by A in G if every path between X and Y is **blocked** by A.

Definition 2.4

Let $G = (V, E)$ be a DAG, and A, B, and C be mutually disjoint subsets of V. We say A and B are d-separated by C in G if for every $X \in A$ and $Y \in B$, X and Y are **D-Separated** by C.

We write

$$I_G(A, B|C) \text{ or } A \perp B|C \tag{8}$$

If $C = \emptyset$, we write only $I_G(A, B)$ or $A \perp B$.

Definition of D-Separation

Definition 2.3

Let $G = (V, E)$ be a DAG, $A \subseteq V$, and X and Y be distinct nodes in $V - A$. We say X and Y are **D-Separated** by A in G if every path between X and Y is **blocked** by A.

Definition 2.4

Let $G = (V, E)$ be a DAG, and A, B, and C be mutually disjoint subsets of V. We say A and B are d-separated by C in G if for every $X \in A$ and $Y \in B$, X and Y are **D-Separated** by C.

We write

$$I_G(A, B|C) \text{ or } A \perp B|C \quad (8)$$

If $C = \emptyset$, we write only $I_G(A, B)$ or $A \perp B$.

Definition of D-Separation

Definition 2.3

Let $G = (V, E)$ be a DAG, $A \subseteq V$, and X and Y be distinct nodes in $V - A$. We say X and Y are **D-Separated** by A in G if every path between X and Y is **blocked** by A.

Definition 2.4

Let $G = (V, E)$ be a DAG, and A, B, and C be mutually disjoint subsets of V. We say A and B are d-separated by C in G if for every $X \in A$ and $Y \in B$, X and Y are **D-Separated** by C.

We write

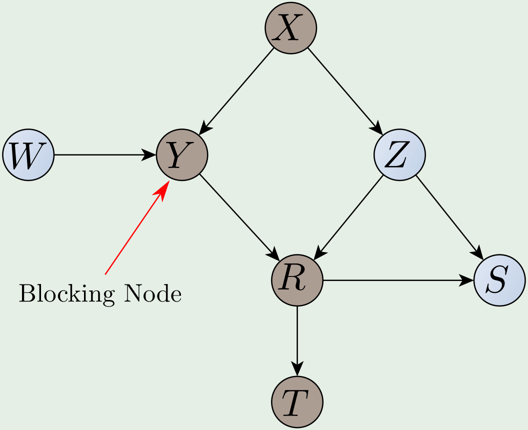
$$I_G(A, B|C) \text{ or } A \perp\!\!\!\perp B|C \quad (8)$$

If $C = \emptyset$, we write only $I_G(A, B)$ or $A \perp\!\!\!\perp B$.

Example

X and T are D-Separated by $\{Y, Z\}$

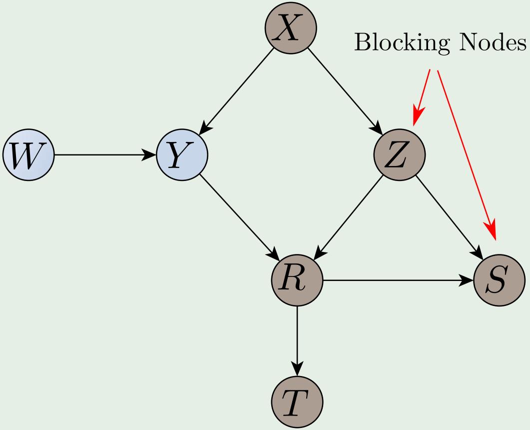
- Because the chain $[X, Y, R, T]$ is blocked at Y.



Example

X and T are D-Separated by $\{Y, Z\}$ - It is the set that block all paths

- Because the chain $[X, Z, S, R, T]$ is blocked at Z, S.



D-Separation ⇒ Independence

D-Separation Theorem

Let P be a probability distribution of the variables in V and $G = (V, E)$ be a DAG. Then (G, P) satisfies the Markov condition if and only if

- for every three mutually disjoint subsets $A, B, C \subseteq V$, whenever A and B are D-Separated by C, A and B are conditionally independent in P given C.

D-Separation $\Rightarrow$ Independence

D-Separation Theorem

Let P be a probability distribution of the variables in V and $G = (V, E)$ be a DAG. Then (G, P) satisfies the Markov condition if and only if

- for every three mutually disjoint subsets $A, B, C \subseteq V$, whenever A and B are D-Separated by C, A and B are conditionally independent in P given C.

That is, (G, P) satisfies the Markov condition if and only if

$$I_G(A, B|C) \Rightarrow I_P(A, B|C) \qquad (9)$$

Cinvestav

D-Separation $\Rightarrow$ Independence

D-Separation Theorem

Let P be a probability distribution of the variables in V and $G = (V, E)$ be a DAG. Then (G, P) satisfies the Markov condition if and only if

- for every three mutually disjoint subsets $A, B, C \subseteq V$, whenever A and B are D-Separated by C, A and B are conditionally independent in P given C.

That is, (G, P) satisfies the Markov condition if and only if

$$I_G(A, B|C) \Rightarrow I_P(A, B|C) \quad (9)$$

Cinvestav

Proof

The proof that, if (G, P) satisfies the Markov condition

Then, each D-Separation implies the corresponding conditional independence is quite lengthy and can be found in [Verma and Pearl, 1990] and in [Neapolitan, 1990].

Then, we will only prove the other direction

Suppose each D-Separation implies a conditional independence.

Thus, the following implication holds

$$I_G(A, B|C) \Rightarrow I_P(A, B|C) \quad (10)$$

Cinvestav

Proof

The proof that, if (G, P) satisfies the Markov condition

Then, each D-Separation implies the corresponding conditional independence is quite lengthy and can be found in [Verma and Pearl, 1990] and in [Neapolitan, 1990].

Then, we will only prove the other direction

Suppose each D-Separation implies a conditional independence.

Thus, the following implication holds

$$I_G(A, B|C) \Rightarrow I_P(A, B|C) \quad (10)$$

Cinvestav

Proof

The proof that, if (G, P) satisfies the Markov condition

Then, each D-Separation implies the corresponding conditional independence is quite lengthy and can be found in [Verma and Pearl, 1990] and in [Neapolitan, 1990].

Then, we will only prove the other direction

Suppose each D-Separation implies a conditional independence.

Thus, the following implication holds

$$I_G(A, B|C) \Rightarrow I_P(A, B|C) \quad (10)$$

Proof

Something Notable

It is not hard to see that a node's parents D-Separate the node from all its non-descendent's that are not its parents.

This is

If we denote the sets of parents and non-descendent's of X by PA_X and ND_X respectively, we have

$$I_G\left(\{X\}, \text{ND}_X - \text{PA}_X | \text{PA}_X\right) \quad (11)$$

Thus

$$I_P\left(\{X\}, \text{ND}_X - \text{PA}_X | \text{PA}_X\right) \quad (12)$$

Proof

Something Notable

It is not hard to see that a node's parents D-Separate the node from all its non-descendent's that are not its parents.

This is

If we denote the sets of parents and non-descendent's of X by PA_X and ND_X respectively, we have

$$I_G\left(\{X\}, \mathsf{ND}_X - \mathsf{PA}_X | \mathsf{PA}_X\right) \tag{11}$$

Thus

$$I_P\left(\{X\}, \mathsf{ND}_X - \mathsf{PA}_X | \mathsf{PA}_X\right) \tag{12}$$

Proof

Something Notable

It is not hard to see that a node's parents D-Separate the node from all its non-descendent's that are not its parents.

This is

If we denote the sets of parents and non-descendent's of X by PA_X and ND_X respectively, we have

$$I_G\left(\{X\}, \mathsf{ND}_X - \mathsf{PA}_X | \mathsf{PA}_X\right) \quad (11)$$

Thus

$$I_P\left(\{X\}, \mathsf{ND}_X - \mathsf{PA}_X | \mathsf{PA}_X\right) \quad (12)$$

Proof

This states the same than

$$I_P\left(\{X\},\mathsf{ND}_X|\mathsf{PA}_X\right) \tag{13}$$

Meaning

The Markov condition is satisfied.

Proof

This states the same than

$$I_P\left(\{X\}, \mathsf{ND}_X | \mathsf{PA}_X\right) \tag{13}$$

Meaning

The Markov condition is satisfied.

Cinvestav

Every Entailed Conditional Independence is Identified by D-Separation

Lemma 2.2

Any conditional independence entailed by a DAG, based on the Markov condition, is equivalent to a conditional independence among disjoint sets of random variables.

Theorem 2.1

Based on the Markov condition, a DAG G entails all and only those conditional independences that are identified by D-Separations in G.

Every Entailed Conditional Independence is Identified by D-Separation

Lemma 2.2

Any conditional independence entailed by a DAG, based on the Markov condition, is equivalent to a conditional independence among disjoint sets of random variables.

Theorem 2.1

Based on the Markov condition, a DAG G entails all and only those conditional independences that are identified by D-Separations in G.

Outline

Now

We would like to find the D-Separations

Since d-separations entail conditional independencies.

We want an efficient algorithm

For determining whether two sets are D-Separated by another set.

For This, we need to build an algorithm

One that can find all D-Separated nodes from one set of nodes by another.

Cinvestav

Now

We would like to find the D-Separations

Since d-separations entail conditional independencies.

We want an efficient algorithm

For determining whether two sets are D-Separated by another set.

For This, we need to build an algorithm

One that can find all D-Separated nodes from one set of nodes by another.

Now

We would like to find the D-Separations

Since d-separations entail conditional independencies.

We want an efficient algorithm

For determining whether two sets are D-Separated by another set.

For This, we need to build an algorithm

One that can find all D-Separated nodes from one set of nodes by another.

How?

To accomplish this

We will first find every node X such that there is at least one active path given A between X and a node in D.

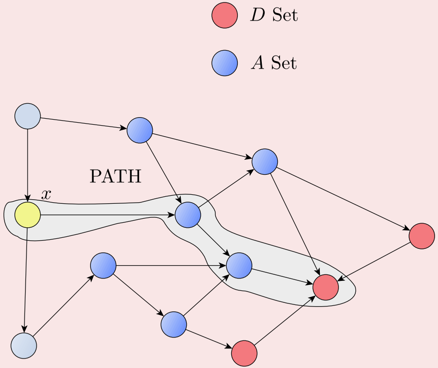
Something like

How?

To accomplish this

We will first find every node X such that there is at least one active path given A between X and a node in D.

Something like



We solve the following problem

Suppose we have a directed graph

We say that certain edges cannot appear consecutively in our paths of interest.

We solve the following problem

Suppose we have a directed graph

We say that certain edges cannot appear consecutively in our paths of interest.

Thus, we identify certain pair of edges $(u \to v, v \to w)$

As legal and the rest as illegal!!

We solve the following problem

Suppose we have a directed graph

We say that certain edges cannot appear consecutively in our paths of interest.

Thus, we identify certain pair of edges $(u \to v, v \to w)$

As legal and the rest as illegal!!

Legal?

- We call a **path legal** if it does not contain any illegal ordered pairs of edges.
- We say Y is **reachable** from x if there is a legal path from x to y.

We solve the following problem

Suppose we have a directed graph

We say that certain edges cannot appear consecutively in our paths of interest.

Thus, we identify certain pair of edges $(u \to v, v \to w)$

As legal and the rest as illegal!!

Legal?

- We call a **path legal** if it does not contain any illegal ordered pairs of edges.
- We say Y is **reachable** from x if there is a legal path from x to y .

Cinvestav

Thus

We can find the set R of all nodes reachable from x as follows

Any node V such that the edge $x \to v$ exists is reachable.

Then

We label such edge with 1.

Next for each such v

We check all unlabeled edges $v \to w$ and see if $(x \to v, v \to w)$ is a legal pair.

Cinvestav

Thus

We can find the set R of all nodes reachable from x as follows

Any node V such that the edge $x \rightarrow v$ exists is reachable.

Then

We label such edge with 1.

Next for each such v

We check all unlabeled edges $v \rightarrow w$ and see if $(x \rightarrow v, v \rightarrow w)$ is a legal pair.

Thus

We can find the set R of all nodes reachable from x as follows

Any node V such that the edge $x \to v$ exists is reachable.

Then

We label such edge with 1.

Next for each such v

We check all unlabeled edges $v \to w$ and see if $(x \to v, v \to w)$ is a legal pair.

Then

We label each such edge with a 2

And keep going!!!

Similar to a Breadth-First Graph

Here, we are visiting links rather than nodes.

Then

We label each such edge with a 2

And keep going!!!

Similar to a Breadth-First Graph

Here, we are visiting links rather than nodes.

What do we want?

Identifying the set of nodes D that are

The one that are D-Separated from B by A in a DAG $G = (V, E)$.

What do we want?

Identifying the set of nodes D that are

The one that are D-Separated from B by A in a DAG $G = (V, E)$.

For this

We need to find the set R such that

- $y \in R \Longleftrightarrow$ Either
  - $y \in B$.
  - There is at least one active chain given A between y and a node in B.

What do we want?

Identifying the set of nodes D that are

The one that are D-Separated from B by A in a DAG $G = (V, E)$.

For this

We need to find the set R such that

- $y \in R \iff$ Either
  - $y \in B$.
  - There is at least one active chain given A between y and a node in B.

What do we want?

Identifying the set of nodes D that are

The one that are D-Separated from B by A in a DAG $G = (V, E)$.

For this

We need to find the set R such that

- $y \in R \iff$ Either
  - $y \in B$.
  - There is at least one active chain given A between y and a node in B.

What do we want?

Identifying the set of nodes D that are

The one that are D-Separated from B by A in a DAG $G = (V, E)$.

For this

We need to find the set R such that

- $y \in R \iff$ Either
  - $y \in B$.
  - There is at least one active chain given A between y and a node in B.

Cinvestav

Then

If there is an active path ρ between node X and some other node

Then every 3-node sub-path $u - v - w$ $(u \neq w)$ of ρ has the following property.

Either

$u - v - w$ is not head-to-head at v and v is not in A.

Or

$u - v - w$ is a head-to-head at v and v is a or has a descendant in A.

Then

If there is an active path ρ between node X and some other node

Then every 3-node sub-path $u - v - w$ $(u \neq w)$ of ρ has the following property.

Either

$u - v - w$ is not head-to-head at v and v is not in A.

Or

$u - v - w$ is a head-to-head at v and v is a or has a descendant in A.

Then

If there is an active path ρ between node X and some other node

Then every 3-node sub-path $u - v - w$ $(u \neq w)$ of ρ has the following property.

Either

$u - v - w$ is not head-to-head at v and v is not in A.

Or

$u - v - w$ is a head-to-head at v and v is a or has a descendant in A.

The Final Legal Rule

The algorithm find-reachable-nodes uses the RULE

Find if $(u \to v, v \to w)$ is legal in G'.

The Final Legal Rule

The algorithm find-reachable-nodes uses the RULE

Find if $(u \to v, v \to w)$ is legal in G'.

The pair $(u \to v, v \to w)$ is legal if and only if

$$u \neq w \tag{14}$$

The Final Legal Rule

The algorithm find-reachable-nodes uses the RULE

Find if $(u \to v, v \to w)$ is legal in G'.

The pair $(u \to v, v \to w)$ is legal if and only if

$$u \neq w \tag{14}$$

And one of the following holds

1. $(u - v - w)$ is not head-to-head in G and $in_A[v]$ is false.
2. $(u - v - w)$ is head-to-head in G and $descendent[v]$ is true.

The Final Legal Rule

The algorithm find-reachable-nodes uses the RULE

Find if $(u \to v, v \to w)$ is legal in G'.

The pair $(u \to v, v \to w)$ is legal if and only if

$$u \neq w \tag{14}$$

And one of the following holds

1. $(u - v - w)$ is not head-to-head in G and $in_A[v]$ is false.
2. $(u - v - w)$ is head-to-head in G and $descendent[v]$ is true.

Cinvestav

Outline

Cinvestav

Example

Reachable nodes from X when $A = \emptyset$, thus

$in_A[v]$ is false and $descendent[v]$ is false for all $v \in V$

Therefore

Only the rule 1 is applicable.

Example

Reachable nodes from X when $A = \emptyset$, thus

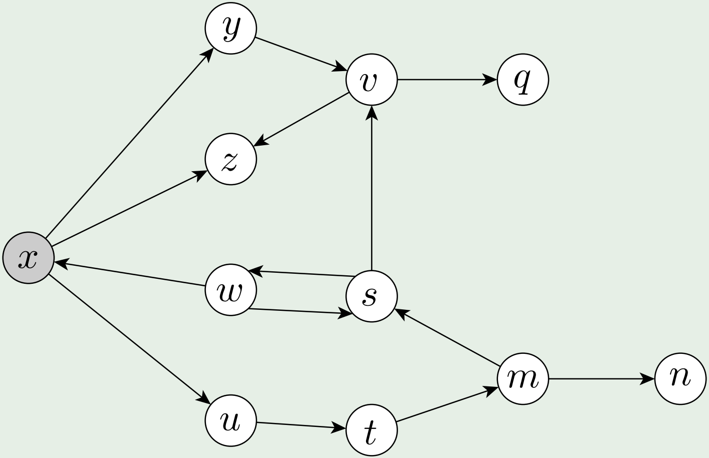
$in_A[v]$ is false and $descendent[v]$ is false for all $v \in V$

Therefore

Only the rule 1 is applicable.

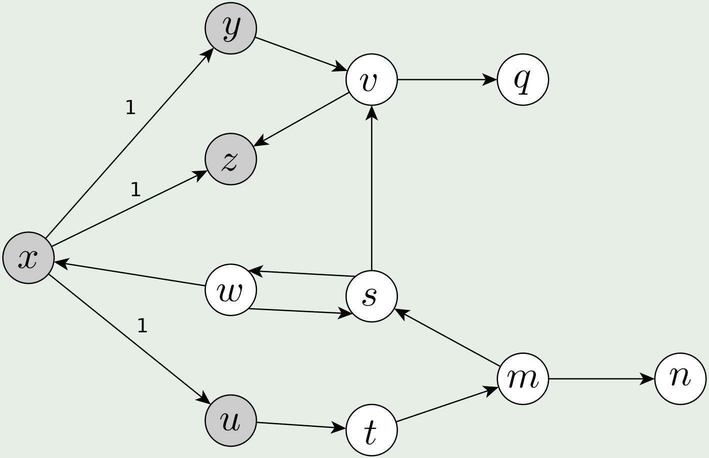
Example

Labeling edges to 1 and shaded nodes are in R



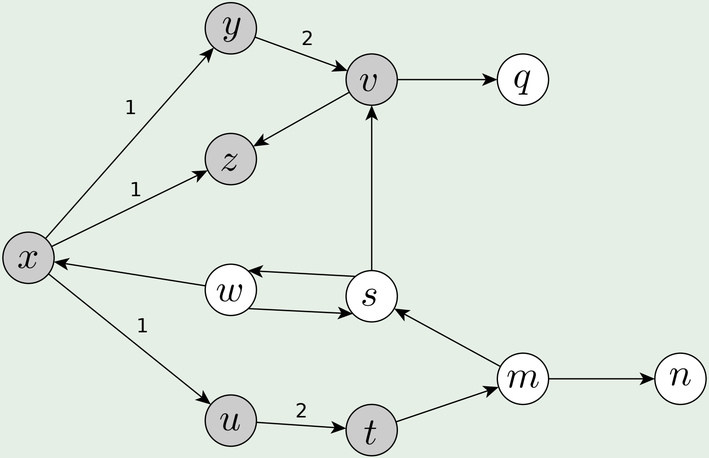
Example

Labeling edges to 1 and shaded nodes are in R



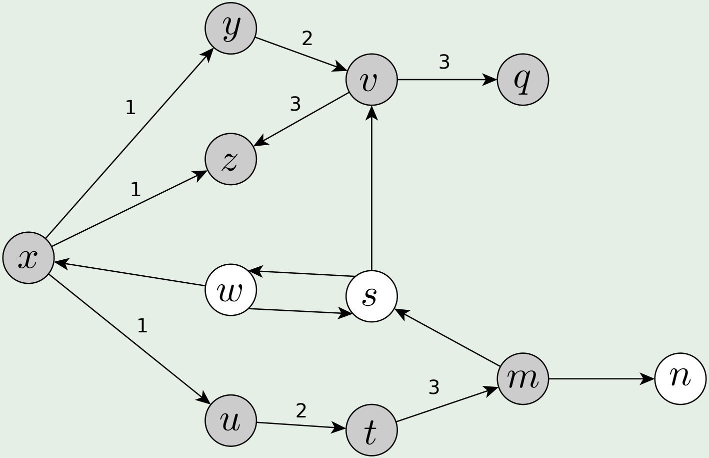
Example

Labeling edges to 2 and shaded nodes are in R



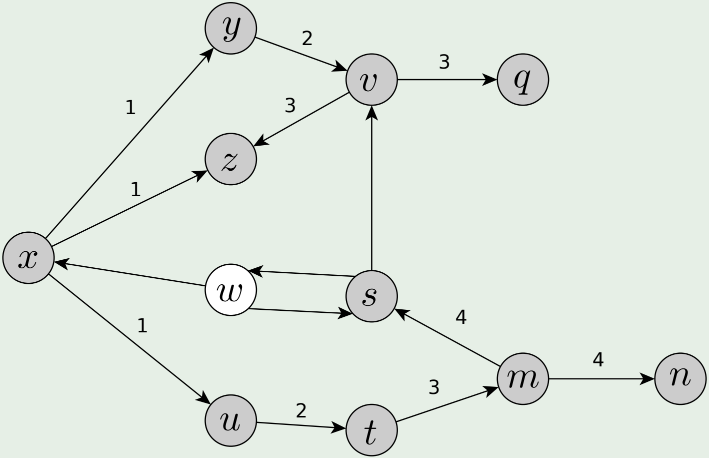
Example

Labeling edges to 3 and shaded nodes are in R



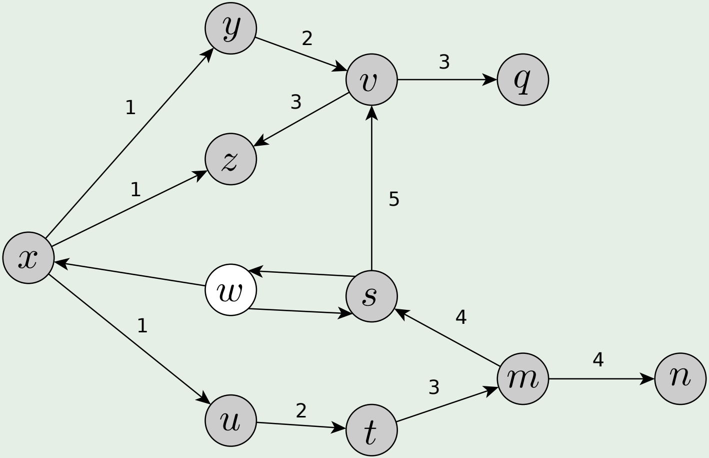
Example

Labeling edges to 4 and shaded nodes are in R



Example

Labeling edges to 5 and shaded nodes are in R



Outline

Cinvestav

Algorithm to Finding Reachability

find-reachable-nodes(G, set of nodes B, set of nodes $\&R$)

Input: G = (V,E), subset B $\subset$ V and a RULE to find if two consecutive edges are legal
Output: R $\subset$ V of all nodes reachable from B

1. for each $x \in$ B
2. add x to R
3. for (each v such that $x \rightarrow v \in$ E)
5. add v to R
6. label $x \rightarrow v$ with 1
7. $i = 1$
8. $found$ =true

Algorithm to Finding Reachability

find-reachable-nodes(G, set of nodes B, set of nodes $\&R$)

Input: $G=$ (V,E), subset B $\subset$ V and a RULE to find if two consecutive edges are legal
Output: R $\subset$ V of all nodes reachable from B

```
for each x ∈ B
    add x to R
        for (each v such that x → v ∈ E)
            add v to R
            label x → v with 1
i = 1
found = true
```

Algorithm to Finding Reachability

find-reachable-nodes$(G,$ set of nodes $B,$ set of nodes $\&R)$

Input: $G=$ (V,E), subset B $\subset$ V and a RULE to find if two consecutive edges are legal
Output: R $\subset$ V of all nodes reachable from B

1. for each $x \in$ B
2. add x to R
3. for (each v such that $x \rightarrow v \in$ E)
5. add v to R
6. label $x \rightarrow v$ with 1
7. $i = 1$
8. $found$ =true

Cinvestav

Algorithm to Finding Reachability

find-reachable-nodes(G, set of nodes B, set of nodes $\&R$)

Input: $G =$ (V,E), subset B $\subset$ V and a RULE to find if two consecutive edges are legal
Output: R $\subset$ V of all nodes reachable from B

1. for each $x \in$ B
2. add x to R
3. for (each v such that $x \to v \in$ E)
5. add v to R
6. label $x \to v$ with 1
7. $i = 1$
8. $found$ =true

Cinvestav

Continuation

The following steps

9. while $(found)$
10. $\quad found =$false
11. $\quad$ for (each v such that $u \rightarrow v$ labeled i)
12. $\quad\quad$ for (each *unlabeled* edge $v \rightarrow w$
13. $\quad\quad\quad$ such $(u \rightarrow v, v \rightarrow w)$ is legal)
14. $\quad\quad\quad\quad$ add w to R
15. $\quad\quad\quad\quad$ label $v \rightarrow w$ with $i + 1$
16. $\quad\quad\quad\quad$ $found =$true
17. $\quad$ $i = i + 1$

Continuation

The following steps

9. while $(found)$
10. $found$ =false
11.   for (each v such that $u \to v$ labeled i)
12.   for (each $unlabeled$ edge $v \to w$
13.   such $(u \to v, v \to w)$ is legal)
14.   add w to R
15.   label $v \to w$ with $i + 1$
16. $found$ =true
17. $i = i + 1$

Continuation

The following steps

```
while (found)
    found = false
    for (each v such that u → v labeled i)
        for (each unlabeled edge v → w
            such (u → v, v → w) is legal)
                add w to R
                label v → w with i + 1
                found = true
    i = i + 1
```

Outline

Cinvestav

Complexity of find-reachable-nodes

We have that

Let n be the number of nodes and m be the number of edges.

Something Notable

In the worst case, each of the nodes can be reached from n entry points.

Thus

Each time a node is reached, an edge emanating from it may need to be re-examined.

Complexity of find-reachable-nodes

We have that

Let n be the number of nodes and m be the number of edges.

Something Notable

In the worst case, each of the nodes can be reached from n entry points.

Thus

Each time a node is reached, an edge emanating from it may need to be re-examined.

Complexity of find-reachable-nodes

We have that

Let n be the number of nodes and m be the number of edges.

Something Notable

In the worst case, each of the nodes can be reached from n entry points.

Thus

Each time a node is reached, an edge emanating from it may need to be re-examined.

Complexity of find-reachable-nodes

Then

Then, in the worst case each edge may be examined n times

Thus, the complexity

$$W(m,n) = \Theta(mn) \tag{15}$$

Complexity of find-reachable-nodes

Then

Then, in the worst case each edge may be examined n times

Thus, the complexity

$$W(m,n) = \Theta(mn) \tag{15}$$

Outline

Cinvestav

Algorithm for D-separation

Find-D-Separations($DAG\ G =$ (V,E) , set of nodes A,B, set of nodes D)

Input: $G =$ (V,E)and two disjoint subsets A,B $\subset$ V

Output: D $\subset$ V containing all nodes D-Separated from every node in B by A. That is I_G (B,D|A) holds and no superset D has this property.

1. for each $v \in$ V
2. if ($v \in$ A)
3. $in_A[v] = true$
4. else
5. $in_A[v] = false$
6. if (v is or has a descendent in A)
7. $descendent[v] = true$
8. else
9. $descendent[v] = false$
10. E$'$ = E $\cup$ $\{u \to v | v \to u \in$ E$\}$
11. G' = (V,E$'$)
12. Run the algorithm:

    find-reachable-nodes(G', B,R)

    $\triangleright$ Note B $\subseteq$ R
13. return D=V $-$ (A $\cup$ R)

Algorithm for D-separation

Find-D-Separations($DAG\ G=$ (V,E) , set of nodes A,B, set of nodes D)

Input: $G=$ (V,E)and two disjoint subsets A,B $\subset$ V

Output: D $\subset$ V containing all nodes D-Separated from every node in B by A. That is I_G (B,D|A) holds and no superset D has this property.

1. for each $v \in$ V
2. if ($v \in$ A)
3. $in_A[v] = true$
4. else
5. $in_A[v] = false$
6. if (v is or has a descendent in A)
7. $descendent[v] = true$
8. else
9. $descendent[v] = false$
10. E$'$ = E $\cup \{u \to v | v \to u \in$ E$\}$
11. G' = (V,E$'$)
12. Run the algorithm:

    find-reachable-nodes(G', B,R)

    ▷ Note B $\subseteq$ R
13. return D=V − (A $\cup$ R)

Algorithm for D-separation

Find-D-Separations($DAG\ G =$ (V,E) , set of nodes A,B, set of nodes D)

Input: $G =$ (V,E)and two disjoint subsets A,B $\subset$ V

Output: D $\subset$ V containing all nodes D-Separated from every node in B by A. That is I_G (B,D|A) holds and no superset D has this property.

1. for each $v \in$ V
2. if ($v \in$ A)
3. $in_A[v] = true$
4. else
5. $in_A[v] = false$
6. if (v is or has a descendent in A)
7. $descendent[v] = true$
8. else
9. $descendent[v] = false$
10. E′ = E ∪ $\{u \to v | v \to u \in$ E$\}$
11. G' = (V,E′)
12. Run the algorithm:

    find-reachable-nodes(G', B,R)

    ▷ Note B $\subseteq$ R
13. return D=V − (A ∪ R)

Algorithm for D-separation

Find-D-Separations($DAG\ G=$ (V,E), set of nodes A,B, set of nodes D)

Input: $G=$ (V,E) and two disjoint subsets A,B $\subset$ V

Output: D $\subset$ V containing all nodes D-Separated from every node in B by A. That is I_G(B,D|A) holds and no superset D has this property.

for each $v \in$ V
    if ($v \in$ A)
        $in_A[v] = true$
    else
        $in_A[v] = false$
    if (v is or has a descendent in A)
        $descendent[v] = true$
    else
        $descendent[v] = false$
E' = E $\cup \{u \to v | v \to u \in$ E$\}$
G' = (V,E')
Run the algorithm:
    find-reachable-nodes(G', B,R)
    ▷ Note B $\subseteq$ R
return D=V $-$ (A $\cup$ R)

Algorithm for D-separation

Find-D-Separations($DAG\ G=$ (V,E) , set of nodes A,B, set of nodes D)

Input: $G=$ (V,E)and two disjoint subsets A,B $\subset$ V

Output: D $\subset$ V containing all nodes D-Separated from every node in B by A. That is I_G (B,D|A) holds and no superset D has this property.

1. for each $v \in$ V
2. if $(v \in$ A)
3. $in_A[v] = true$
4. else
5. $in_A[v] = false$
6. if (v is or has a descendent in A)
7. $descendent[v] = true$
8. else
9. $descendent[v] = false$
10. E$' =$ E $\cup \{u \to v | v \to u \in$ E$\}$
11. $G' =$ (V,E$'$)
12. Run the algorithm:

    find-reachable-nodes(G', B,R)

    ▷ Note B $\subseteq$ R
13. return D=V $-$ (A $\cup$ R)

Algorithm for D-separation

Find-D-Separations($DAG\ G =$ (V,E) , set of nodes A,B, set of nodes D)

Input: $G =$ (V,E)and two disjoint subsets A,B $\subset$ V

Output: D $\subset$ V containing all nodes D-Separated from every node in B by A. That is I_G (B,D|A) holds and no superset D has this property.

1. for each $v \in$ V
2. if ($v \in$ A)
3. $in_A[v] = true$
4. else
5. $in_A[v] = false$
6. if (v is or has a descendent in A)
7. $descendent[v] = true$
8. else
9. $descendent[v] = false$
10. E$' =$ E $\cup \{u \to v | v \to u \in$ E$\}$
11. $G' =$ (V,E')
12. Run the algorithm:
    find-reachable-nodes(G', B,R)
    ▷ Note B $\subseteq$ R
13. return D=V − (A $\cup$ R)

Algorithm for D-separation

Find-D-Separations($DAG\ G =$ (V,E) , set of nodes A,B, set of nodes D)

Input: $G =$ (V,E)and two disjoint subsets A,B $\subset$ V

Output: D $\subset$ V containing all nodes D-Separated from every node in B by A. That is I_G (B,D|A) holds and no superset D has this property.

1. for each $v \in$ V
2. if ($v \in$ A)
3. $in_A[v] = true$
4. else
5. $in_A[v] = false$
6. if (v is or has a descendent in A)
7. $descendent[v] = true$
8. else
9. $descendent[v] = false$
10. E$'$ = E $\cup \{u \to v | v \to u \in$ E$\}$
11. $G' =$ (V,E')
12. Run the algorithm:

    find-reachable-nodes(G', B,R)

    ▷ Note B $\subseteq$ R
13. return D=V − (A $\cup$ R)

Algorithm for D-separation

Find-D-Separations($DAG\ G =$ (V,E) , set of nodes A,B, set of nodes D)

Input: $G =$ (V,E)and two disjoint subsets A,B $\subset$ V

Output: D $\subset$ V containing all nodes D-Separated from every node in B by A. That is I_G (B,D|A) holds and no superset D has this property.

1. for each $v \in$ V
2.   if ($v \in$ A)
3. $in_A[v] = true$
4.   else
5. $in_A[v] = false$
6.   if (v is or has a descendent in A)
7. $descendent[v] = true$
8.   else
9. $descendent[v] = false$
10. E$'$ = E $\cup \{u \to v | v \to u \in$ E$\}$
11. $G' =$ (V,E')
12. Run the algorithm:

    find-reachable-nodes(G', B,R)

    ▷ Note B $\subseteq$ R
13. return D=V $-$ (A $\cup$ R)

Observation about $descendent\,[v]$

We can implement the construction of $descendent\,[v]$ as follow

Initially set $descendent\,[v] =$ true for all nodes in A.

Then

Then follow the incoming edges in A to their parents, their parents' parents, and so on.

Thus

We set $descendent\,[v] =$ true for each node found along the way.

Observation about $descendent\,[v]$

We can implement the construction of $descendent\,[v]$ as follow

Initially set $descendent\,[v] =$ true for all nodes in A.

Then

Then follow the incoming edges in A to their parents, their parents' parents, and so on.

Thus

We set $descendent\,[v] =$ true for each node found along the way.

Observation about $descendent\,[v]$

We can implement the construction of $descendent\,[v]$ as follow

Initially set $descendent\,[v] =$ true for all nodes in A.

Then

Then follow the incoming edges in A to their parents, their parents' parents, and so on.

Thus

We set $descendent\,[v] =$ true for each node found along the way.

Observation about E'

The RULE about legal and E'

The E' is necessary because using only the RULE on E will no get us all the active paths.

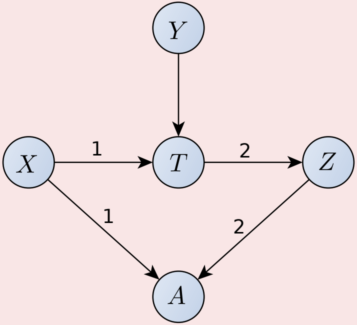
For example

Observation about E'

The RULE about legal and E'

The E' is necessary because using only the RULE on E will no get us all the active paths.

For example



Thus

Something Notable

Given A is the only node in A and $X \to T$ is the only edge in B, the edges in that DAG are numbered according to the method.

Then

The active chain $X \to A \leftarrow Z \leftarrow T \leftarrow Y$ is missed because the edge $T \to Z$ is already numbered by the time the chain $A \leftarrow Z \leftarrow T$ is investigated.

But

If we use the set of edges E'.

Thus

Something Notable

Given A is the only node in A and $X \rightarrow T$ is the only edge in B, the edges in that DAG are numbered according to the method.

Then

The active chain $X \rightarrow A \leftarrow Z \leftarrow T \leftarrow Y$ is missed because the edge $T \rightarrow Z$ is already numbered by the time the chain $A \leftarrow Z \leftarrow T$ is investigated.

But

If we use the set of edges E'.

Cinvestav

Thus

Something Notable

Given A is the only node in A and $X \to T$ is the only edge in B, the edges in that DAG are numbered according to the method.

Then

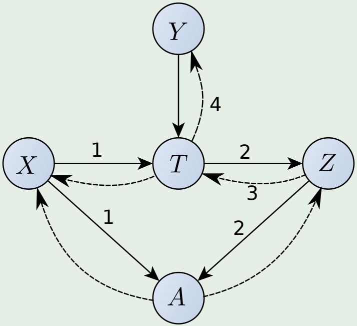
The active chain $X \to A \leftarrow Z \leftarrow T \leftarrow Y$ is missed because the edge $T \to Z$ is already numbered by the time the chain $A \leftarrow Z \leftarrow T$ is investigated.

But

If we use the set of edges E'.

Thus

Once, we add the extra edges, we get



Outline

Complexity

Please take a look at page 81 at

"Learning Bayesian Networks" by Richard E. Neapolitan.

For the analysis of the algorithm for m edges and n nodes

$$\Theta(m) \text{ with } m \geq n. \tag{16}$$

Complexity

Please take a look at page 81 at

"Learning Bayesian Networks" by Richard E. Neapolitan.

For the analysis of the algorithm for m edges and n nodes

$$\Theta(m) \text{ with } m \geq n. \tag{16}$$

The D-Separation Algorithm works

Theorem 2.2

The set D contains all and only nodes D-Separated from every node in B by A.

- That is, we have I_G (B,D|A) and no superset of D has this property.

Proof

- The set R determined by the algorithm contains
  - All nodes in B.
  - All nodes reachable from B via a legal path in G'.

The D-Separation Algorithm works

Theorem 2.2

The set D contains all and only nodes D-Separated from every node in B by A.

- That is, we have I_G (B,D|A) and no superset of D has this property.

Proof

- The set R determined by the algorithm contains
  - All nodes in B.
  - All nodes reachable from B via a legal path in G'.

Proof

For any two nodes $x \in$ B and $y \notin$ A $\cup$ B

The path $x - ... - y$ is active in G if and only if the path $x \rightarrow ... \rightarrow y$ is legal in G'.

Thus

Thus R contains the nodes in B plus all and only those nodes that have active paths between them and a node in B.

By the definition of D-Separation

A node is D-Separated from every node in B in A if the node is not in A $\cup$ B and there is not active path between the node and a node in B.

Proof

For any two nodes $x \in$ B and $y \notin$ A $\cup$ B

The path $x - ... - y$ is active in G if and only if the path $x \rightarrow ... \rightarrow y$ is legal in G'.

Thus

Thus R contains the nodes in B plus all and only those nodes that have active paths between them and a node in B.

By the definition of D-Separation

A node is D-Separated from every node in B in A if the node is not in A $\cup$ B and there is not active path between the node and a node in B.

Proof

For any two nodes $x \in$ B and $y \notin$ A $\cup$ B

The path $x - ... - y$ is active in G if and only if the path $x \rightarrow ... \rightarrow y$ is legal in G'.

Thus

Thus R contains the nodes in B plus all and only those nodes that have active paths between them and a node in B.

By the definition of D-Separation

A node is D-Separated from every node in B in A if the node is not in A $\cup$ B and there is not active path between the node and a node in B.

Proof

Thus

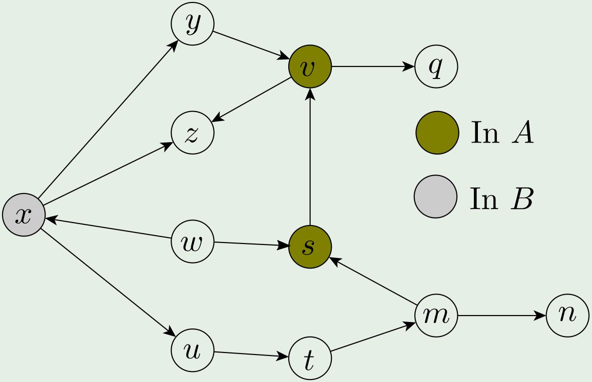
D=V-$(A \cup R)$ is the set of all nodes D-Separated from every node in B by A.

Outline

Cinvestav

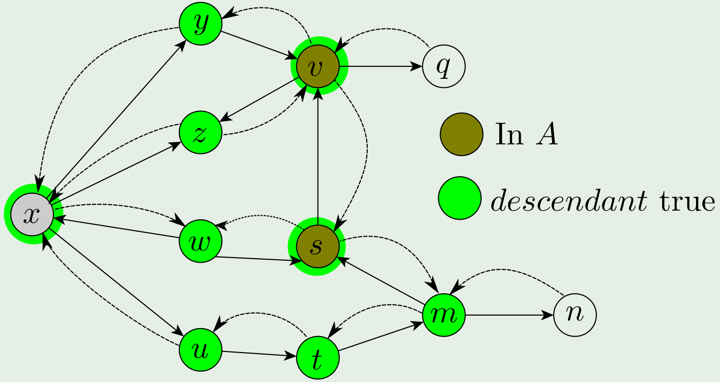
Example

$B = \{x\}$ and $A = \{s, v\}$- Original Graph!!!



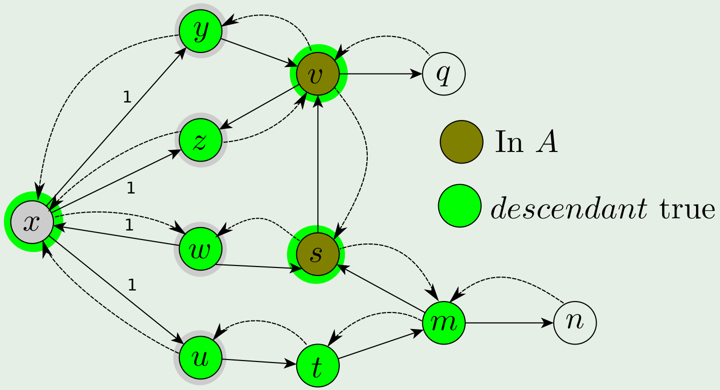
Example

$B = \{x\}$ and $A = \{s, v\}$- Moralized and with the tracking of descendants



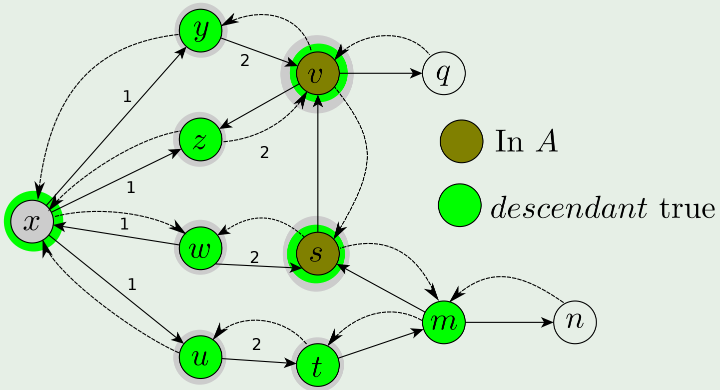
Example

$B = \{x\}$ and $A = \{s, v\}$ and the first part of the reachability algorithm



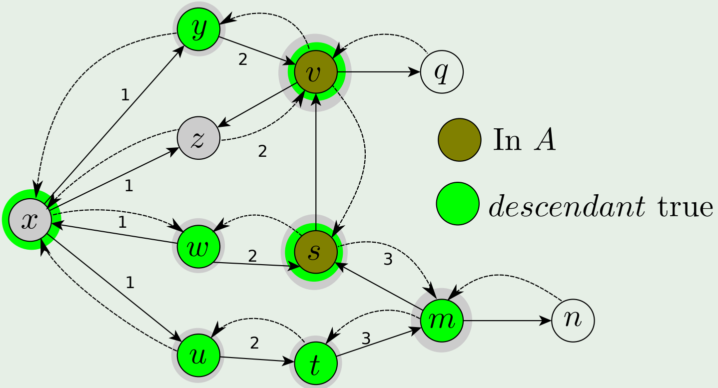
Example

Remember that Legality is in G



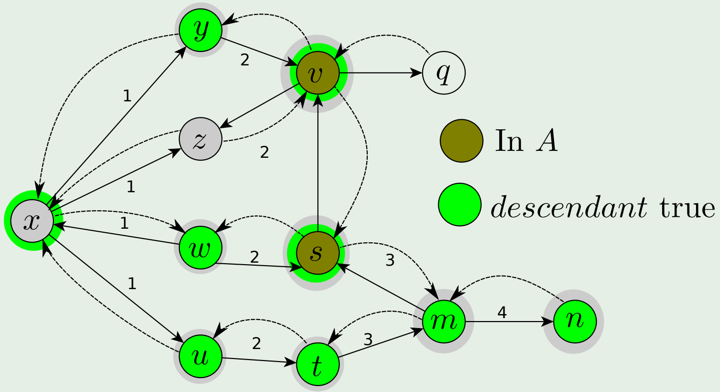
Example

Now, we have that



Example

$D - \{A \cup R\} = \{q\}$



Outline

Cinvestav

Application

Something Notable

In general, the inference problem in Bayesian networks is to determine $P(B|A)$, where A and B are two sets of variables.

Thus

We can use the D-Separation for that.

Application

Something Notable

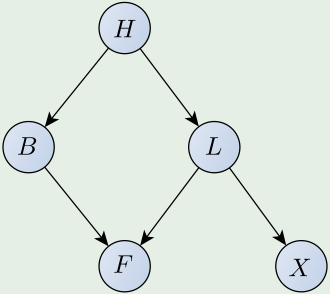
In general, the inference problem in Bayesian networks is to determine $P(B|A)$, where A and B are two sets of variables.

Thus

We can use the D-Separation for that.

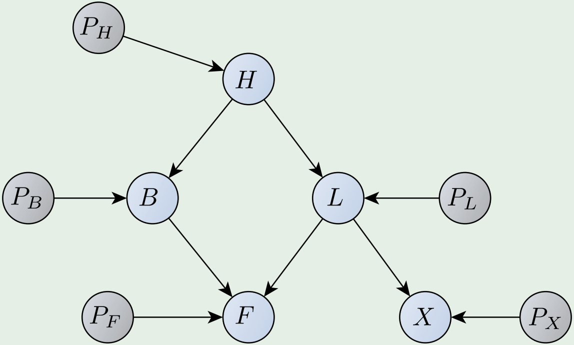
Example

Given the following the DAG G



Example

We generate G'



Where

The extra nodes represent

The probabilities in the interval $[0,1]$ and representing $P(X=x)$

Creating a set of P be the set of auxiliary parent nodes

Thus, if we want to determine $P(\text{B}|\text{A})$ in G, we can use the algorithm for D-Separation to find D.

Such that

$$I_{G'}(\text{B,D}|\text{A}) \quad (17)$$

And no superset of D has this property, then take D∩P.

Where

The extra nodes represent

The probabilities in the interval $[0, 1]$ and representing $P(X = x)$

Creating a set of P be the set of auxiliary parent nodes

Thus, if we want to determine $P(\mathrm{B}|\mathrm{A})$ in G, we can use the algorithm for D-Separation to find D.

Such that

$$I_{G'}(\mathrm{B},\mathrm{D}|\mathrm{A}) \tag{17}$$

And no superset of D has this property, then take D∩P.

Where

The extra nodes represent

The probabilities in the interval $[0,1]$ and representing $P(X=x)$

Creating a set of P be the set of auxiliary parent nodes

Thus, if we want to determine $P(\text{B}|\text{A})$ in G, we can use the algorithm for D-Separation to find D.

Such that

$$I_{G'}(\text{B,D}|\text{A}) \tag{17}$$

And no superset of D has this property, then take D∩P.

For example

We want $P(f)$

To determine it we need that all and only the values of P_H , P_B , P_L , and P_F

Because

$$I_{G'}(\{F\}, \{P_X\} | \emptyset) \quad (18)$$

Thus

P_X is the only auxiliary parent variable D-Separated from $\{F\}$ by the empty set.

For example

We want $P(f)$

To determine it we need that all and only the values of P_H , P_B , P_L , and P_F

Because

$$I_{G'}\left(\{F\},\{P_X\}\,|\emptyset\right) \tag{18}$$

Thus

P_X is the only auxiliary parent variable D-Separated from $\{F\}$ by the empty set.

For example

We want $P(f)$

To determine it we need that all and only the values of P_H , P_B , P_L , and P_F

Because

$$I_{G'}(\{F\}, \{P_X\} | \emptyset) \quad (18)$$

Thus

P_X is the only auxiliary parent variable D-Separated from $\{F\}$ by the empty set.

For example

We want $P(f|b)$

To determine it we need that all and only the values of P_H , P_L , and P_F when separation set is $\{B\}$

Then

$$I_{G'}\left(\{F\},\{P_X,P_B\}|\{B\}\right) \quad (19)$$

Thus

P_X is the only auxiliary parent variable D-Separated from $\{F\}$ by the empty set.

For example

We want $P\left(f|b\right)$

To determine it we need that all and only the values of P_H , P_L , and P_F when separation set is $\{B\}$

Then

$$I_{G'}\left(\{F\}, \{P_X, P_B\} \,|\, \{B\}\right) \tag{19}$$

Thus

P_X is the only auxiliary parent variable D-Separated from $\{F\}$ by the empty set.

For example

We want $P(f|b)$

To determine it we need that all and only the values of P_H , P_L , and P_F when separation set is $\{B\}$

Then

$$I_{G'}(\{F\}, \{P_X, P_B\} | \{B\}) \quad (19)$$

Thus

P_X is the only auxiliary parent variable D-Separated from $\{F\}$ by the empty set.

Outline

Cinvestav

Remarks

Bayes Networks can reflect the true causal patterns

- Often simpler (nodes have fewer parents).
- Often easier to think about.
- Often easier to elicit from experts.

Remarks

Bayes Networks can reflect the true causal patterns

- Often simpler (nodes have fewer parents).
- Often easier to think about.
- Often easier to elicit from experts.

Something Notable

- Sometimes no causal net exists over the domain.
- For example, consider the variables Traffic and Drips.
  - Arrows reflect correlation not causation.

Remarks

Bayes Networks can reflect the true causal patterns

- Often simpler (nodes have fewer parents).
- Often easier to think about.
- Often easier to elicit from experts.

Something Notable

- Sometimes no causal net exists over the domain.
- For example, consider the variables Traffic and Drips.
  - Arrows reflect correlation not causation.

Remarks

Bayes Networks can reflect the true causal patterns

- Often simpler (nodes have fewer parents).
- Often easier to think about.
- Often easier to elicit from experts.

Something Notable

- Sometimes no causal net exists over the domain.
- For example, consider the variables Traffic and Drips.
    - Arrows reflect correlation not causation.

Remarks

Bayes Networks can reflect the true causal patterns

- Often simpler (nodes have fewer parents).
- Often easier to think about.
- Often easier to elicit from experts.

Something Notable

- Sometimes no causal net exists over the domain.
- For example, consider the variables Traffic and Drips.
  - Arrows reflect correlation not causation.

Remarks

Bayes Networks can reflect the true causal patterns

- Often simpler (nodes have fewer parents).
- Often easier to think about.
- Often easier to elicit from experts.

Something Notable

- Sometimes no causal net exists over the domain.
- For example, consider the variables Traffic and Drips.
  - Arrows reflect correlation not causation.

Remarks

What do the arrows really mean?

- Topologies may happen to encode causal structure.
- Topologies are only guaranteed to encode conditional independence!

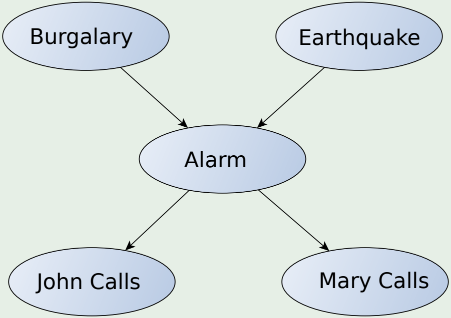
Remarks

What do the arrows really mean?

- Topologies may happen to encode causal structure.
- Topologies are only guaranteed to encode conditional independence!

Example

Example

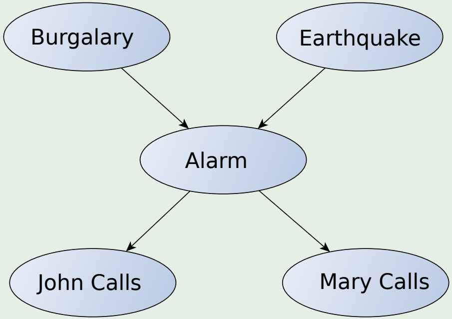


Add more stuff

- Given that some information is not being encoded into the network:
  - We have to add more edges to the graph.

Example

Example



Add more stuff

- Given that some information is not being encoded into the network:
  - We have to add more edges to the graph.

Example

Thus

- Adding edges allows to make different conditional independence assumptions.

New Network

Example

Thus

- Adding edges allows to make different conditional independence assumptions.

New Network