Causation to Association III: D-separation

1000: Introduction

1100: Overview

In the previous two modules, we developed the pieces we need to move from causal graphs to predictions about association and independence. We discussed how causal connections produce unconditional association, we discussed how conditioning on common causes or mediators prevent a causal connection from producing association, and we discussed how conditioning on a common effect induces association. In this module, we put the pieces together into a unified theory. When you finish this module, you should be able to take any causal graph and write down the set of independence relations, both unconditional and conditional, entailed to hold among the variables in the graph. Put more formally, for any two variables X and Y and any set of distinct variables **Z**, you should be able to ascertain whether a causal graph containing X, Y and **Z** entails that X and Y are independent conditional on **Z**.

The unified theory connecting causal graphs to association is called **d-separation**, which stands for "dependence separation." The theory was developed by computer scientists Judea Pearl, Thomas Verma, and Dan Geiger working at UCLA in the mid 1980s. Working on the foundations of artificial intelligence, they wanted to make it possible for a robot to represent causal beliefs, incorporate uncertainty (e.g., there is a 40% chance of rain tomorrow), and learn from its interactions with the world. As it turns out, if some beliefs (e.g., those about frequencies) are independent of others, then a robot can store its joint set of beliefs much more efficiently. Since causal graphs entail that the frequencies for variables are independent of the frequencies of others, a graph's structure can help a robot store information very efficiently. For example, consider a causal graph with 10 binary variables in which the whole graph is one long causal chain:

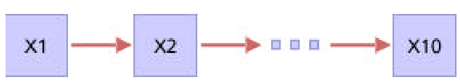

FIGURE 1100-1

Without using the independence relations implied by this graph, a robot would need 1,023 separate probabilities to represent its uncertainty (the frequencies it expects) over the joint set of possible values X1 - X10 might take on. By using the independencies entailed by this graph, however, it would only need 19 numbers to store the same information! And the efficiency gain gets proportionally larger as the number of variables goes up.

When a robot learns something by interacting in the world, it might need to update its causal beliefs, i.e., change its causal graphs. Upon doing so, it would also need to update the independence facts entailed by the new causal graphs. In this way it could learn about the world and still be able to store beliefs efficiently. It was thus essential for the robot to be capable of correctly computing the independence statements entailed by any graph it might build. Put another way, it was essential to have the general theory of d-separation.

By itself, d-separation applies to causal graphs. For any two variables **X** and **Y**, and any set of variables **Z**, you will learn how to determine if **X** and **Y** are d-separated by **Z** in a causal graph involving **X**, **Y**, and **Z** just by examining the graph itself. There are no probabilities or frequencies involved.

We should note that the names we often use in this module: **X**, **Y** and **Z** do not only apply to variables in a graph with those names. Any variable in the graph can be considered as the **X** variable in this module. Likewise for **Y** and the set we give the name **Z**.

A gentle warning. This module is a little less intuitive than the two preceding it. Although all the pieces of d-separation can be given an intuitive motivation -- that is what the previous two modules contribute -- learning the theory is more than understanding the intuitions behind each piece. Learning it means being able to compute it quickly and reliably. And just as learning to do long multiplication is aided by learning why we carry, etc., learning to really do multiplication requires a certain amount of drill and practice. Learning d-separation is the same way. It is not hard, mind you. If you understood the previous two modules, then learning it is an hour or so.

What we can promise is that its worth it. By understanding 1) causal graphs, 2) how interventions change a causal graph, and 3) d-separation, you will have all you need to qualitatively understand the main problems of causal inference and the main strategies for dealing with these problems.

1200: The Causality Lab

D-separation is programmed into the Causality Lab. You can construct a causal graph in the Causality Lab, and then ask it to compute which pairs of variables are predicted to be associated, which pairs independent, and which pairs are conditionally associated or conditionally independent.

Before you proceed with this module, or at any time during the module, we invite you to use this feature of the Lab. See if you can figure out the theory of d-separation before we present it. Exactly what characteristics must a causal graph have in order for it to predict that X and Y are associated? What characteristics must it have for it to entail that X and Y are independent? Experiment until you have a theory. Then read the module and see if you got it right.

In particular, try using the following graph:

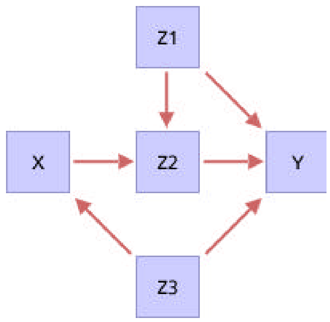

FIGURE 1200-1

< A link to a Java applet in the interactive version of this module. >

2000: Undirected Paths

The theory of d-separation begins by considering all **undirected paths** between a pair of variables X and Y. It then identifies which paths are **active** relative to **Z**, that is, produce association conditional on the set **Z**, and which are **inactive**. If every undirected path between X and Y is inactive relative to **Z**, then X and Y are d-separated by **Z**. If any are active relative to **Z**, then X and Y are d-connected by **Z**, i.e., X and Y are not d-separated by **Z**. So first you need to know how to reliably identify all the undirected paths that connect a pair of variables.

An undirected path from X to Y is just a sequence of contiguous edges from X to Y in which the arrows can be in any direction whatsoever. The sequence cannot have any loops, that is, no variable can occur twice on an undirected path.

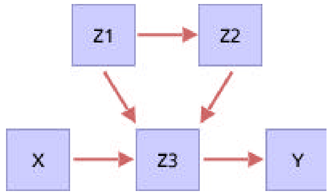

FIGURE 2000-1

For example, in the graph above, we might list the following sequences of edges as candidate undirected paths connecting **X** and **Y**:

1. **X → Z3 → Y**
2. **X → Z3 ← Z1 → Z2 → Z3 → Y**
3. **X → Z3 ← Z1 → Z2 → Z3 ← Z1 → Z2 → Z3 → Y**
4. **X → Z3 ← Z2 ← Z1 → Z3 → Y**

Candidates 2, 3, and 4 are not undirected paths, however, because **Z3** occurs twice on each.

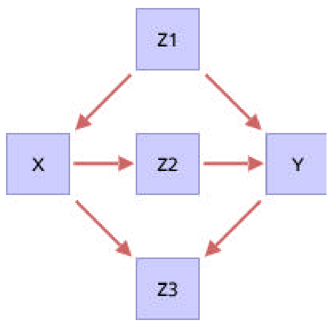

FIGURE 2000-2

If all of the arrows in an undirected path point in the same direction, then the path is an **directed path**. Undirected paths include directed paths as a subset, but not vice versa. That is, all directed paths are undirected paths, but not vice versa. For example, there are three undirected paths between **X** and **Y** in Fig. 2000-2, but only one of them is a directed path:

1. **X ← Z1 → Y**
2. **X → Z2 → Y**
3. **X → Z3 ← Y**

< A link to exercises in the interactive version of this module. >

In the graph above, one of the undirected paths between **X** and **Y** went through only **Z1**, one went through only **Z2**, and a third went through only **Z3**. In more densely connected graphs, several paths can go through the same variable.

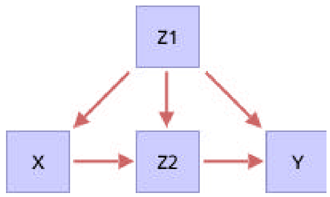

FIGURE 2000-3

For example, in this graph the undirected paths between **X** and **Y** are:

+ **X** ← **Z1** → **Y**
+ **X** ← **Z1** → **Z2** → **Y**
+ **X** → **Z2** ← **Z1** → **Y**
+ **X** → **Z2** → **Y**

Z1 occurs on three of the four paths, as does **Z2**. Notice also that the same variables can occur on two different paths. For example, paths 2 and 3 both contain **X**, **Z1**, **Z2**, and **Y**.

Consider the more complicated causal graph below.

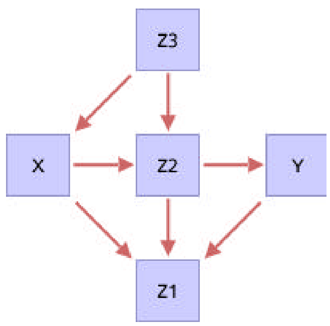

FIGURE 2000-4

Take out a piece of paper, and list all the undirected paths from **X** to **Y**.

< A link to exercises in the interactive version of this module. >

Here are the six undirected paths between **X** and **Y** in the graph above.

+ X ← Z3 → Z2 → Y
+ X → Z2 → Y
+ X → Z2 → Z1 ← Y
+ X → Z1 ← Y
+ X → Z1 ← Z2 → Y
+ X ← Z3 → Z2 → Z1 ← Y

3000: Active and Inactive Paths

How do we compute whether an undirected path between X and Y is active or inactive relative to **Z**? The answer is extremely simple, but puts off the real work: A path is active relative to **Z** just in case **every** variable on the path is active relative to **Z**.

Some analogies may help. When two variables X and Y are associated, one provides information about the other. That doesn't mean that knowing X's value tells you enough about Y to know its value exactly, or vice versa. It means that your beliefs about the frequencies for Y's values should change after you learn something about X's value, and vice versa. Undirected paths can be thought of as potential transmitters of information. If a path between X and Y is active, then information can get from X to Y or vice versa via that path. Every variable on the path has to be active for the path as a whole to be active. Like an electrical circuit with lots of switches, or a set of pipes with lots of valves, if any node on the path is inactive, the whole path is blocked.

The ideas of d-connection and d-separation are opposites. X and Y are d-connected by **Z** just in case X and Y are not d-separated by **Z**, and vice versa. We summarize the theory in terms of d-connection.

1. X and Y are **d-connected** by **Z** just in case **any path** between Y and Y is **active** relative to **Z**.
2. A **path is active** just in case **all** the variables on the path are active.

As we proceed, we will expand this summary until we reach the complete theory of d-separation, which is only 5 clauses. We now turn to determining when a variable is active or inactive on a path, **relative** to the conditioning set **Z**.

4000: Active and Inactive Variables on a Path

4100: The Unconditional Case

The unconditional case is easy, so we begin there. We presented the intuitions behind the unconditional case in the module on Causation to Association I: Unconditional Association, and again in section 2000 of the module on Causation to Association II: Conditional Association. If, at any time, you want to return to these modules for review, please do.

In general, we want to determine whether X is d-separated from Y by a set **Z** in a graph. In the unconditional case, the conditioning set **Z** is empty (**Z** = ∅), and we say: X is d-separated from Y by the empty set, or we say: X is unconditionally d-separated from Y.

When is a variable active on a path between X and Y relative to the empty set? The answer is extremely simple:

If **Z** = ∅, then non-colliders are active and colliders are inactive.

An undirected path is active relative to the empty set just in case it has no colliders. What kind of path has no colliders? A causal connection. As you surely recall from the scintillating module: Causation to Association I, only causal connections produce unconditional association.

< A link to exercises in the interactive version of this module. >

To summarize what we know so far:

1. X and Y are **d-connected** by **Z** just in case **any path** between Y and Y is **active** relative to **Z**.
2. A **path is active** just in case **all** the variables on the path are active.
3. If **Z** = ∅, then a variable is **active** on a path if it occurs as a **non-collider**, and is **inactive** if it occurs as a **collider**.

We now consider how, when the conditioning set **Z** is not empty, a variable's status can be changed by **Z**. As we consider each case, we will add a clause to this summary, until we have the complete theory.

4200: The Conditional Case

4210: Conditioning on Non-colliders

We presented the intuitions behind the conditioning on non-colliders in sections 3000, 4000, and 5000 of the module on Causation to Association II: Conditional Association.

Non-colliders are active relative to the empty set. Conditioning on a non-collider deactivates it. A non-collider is active as long as it is not in the conditioning set **Z**. If it is in **Z**, then its status reverses from active to inactive. Put another way, a non-collider carries information unless we condition on it, in which case we block it. Consider some examples.

Example 1: Chicken Pox

In the Chicken Pox example, we hypothesize that exposure is a direct cause of infection, which is a direct cause of symptoms.

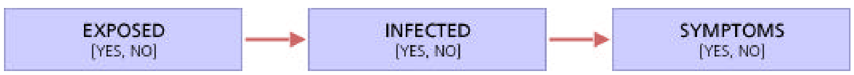

FIGURE 4210-1

EXPOSED and **SYMPTOMS** are d-connected relative to the empty set, because there is a path between them on which **INFECTED** is a non-collider and thus active. When **INFECTED** is in the conditioning set, then its status switches to inactive. **EXPOSED** and **SYMPTOMS** are therefore d-separated by **INFECTED**.

Example 2: Smoking

In the classic **SMOKES**, **YELLOW FINGERS** and **LUNG CANCER** example:

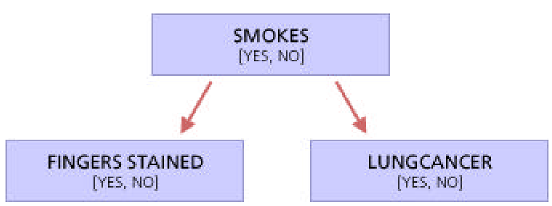

FIGURE 4210-2

YELLOW FINGERS and **LUNG CANCER** are d-connected relative to the empty set. Why? Because **SMOKES** is a non-collider on the only path connecting them -- and is thus active relative to the empty set. When **SMOKES** is in the conditioning set Z, then it is made inactive, and **YELLOW FINGERS** and **LUNG CANCER** are d-separated by **SMOKES**.

Example 3: Exercise and Body Weight

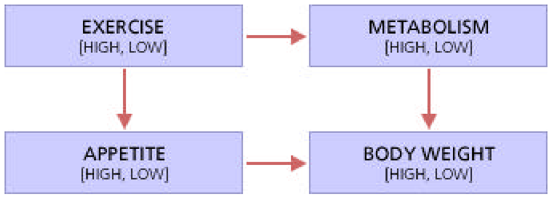

FIGURE 4210-3

EXERCISE and **BODY WEIGHT** are d-connected relative to the empty set because there are 2 active paths between them:

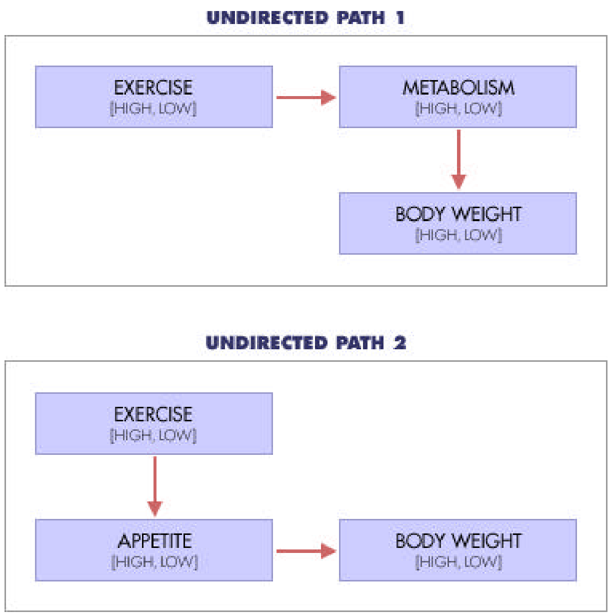

FIGURE 4210-4

METABOLISM is a non-collider and thus active on path 1, and **APPETITE** a non-collider and thus active on path 2. If the conditioning set includes just **METABOLISM**, then path 1 is inactive, but path 2 is still active. If the conditioning set includes just **APPETITE**, then the reverse is true: path 2 is inactive, but path 1 is still active. For **EXERCISE** and **BODY WEIGHT** to be d-separated by a set, the set must deactivate all the paths between them. Thus it must include both **APPETITE** and **METABOLISM**.

The Trivial Case

What happens in the trivial case of a causal connection that is a direct cause? For exampe, consider the undirected paths between **X** and **Y** in this graph:

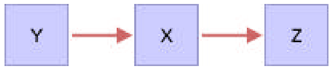

FIGURE 4210-5

The only path is the direct edge Y → X. In every undirected path, however, the endpoints are classified as non-colliders, so Y and X are non-colliders and thus active relative to the empty set. Since the only way to deactivate a non-collider is to condition on it, we can only d-separate Y and X in this case by including one of them in the conditioning set **Z**. So X and Y are not d-separated by the set {Z}, but they are d-separated by the set {X}, for example. When the conditioning set **Z** includes either X OR Y, we say that X and Y are trivially d-separated by **Z**. In general, we only consider conditioning sets that include neither X nor Y.

< A link to exercises in the interactive version of this module. >

To summarize what we know so far:

1. X and Y are **d-connected** by **Z** just in case **any path** between Y and Y is **active** relative to **Z**.
2. A **path is active** just in case **all** the variables on the path are active.
3. If **Z** = ∅, then a variable is **active** on a path if it occurs as a **non-collider**, and is **inactive** if it occurs as a **collider**.
4. If **Z** ≠ ∅, then **Non-colliders** are **inactive** if they are in **Z**, and **active** if they are not in **Z**

4220: Conditioning on Colliders

The intuitions behind conditioning on colliders is presented in section 6000 of the module on Causation to Association II: Conditional Association.

Unconditionally, non-colliders are active and colliders are inactive. In the last section we saw that conditioning on a non-collider flips its status from active to inactive. Colliders act in a similar way: conditioning on a collider flips its status from inactive to active. Consider a few examples.

Example 1: Starting the Car

TABLE 4220-1: VARIABLES FOR FOR STARTING A CAR

Variables	Values
BATTERY	[Charged, Uncharged]
CAR STARTS	[Yes, No]
GAS TANK	[Empty, Not empty]
EXHAUST PIPE	[Emitting, Not emitting]

Lets assume that the following graph depicts the causal relations among these four variables.

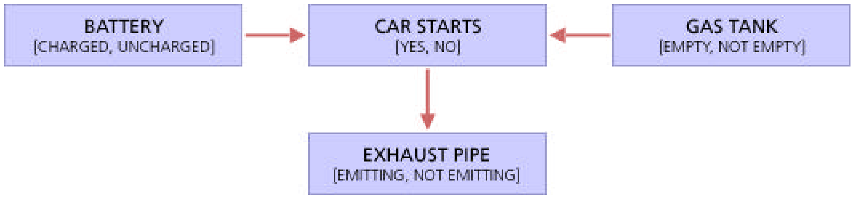

FIGURE 4220-1

Consider the only undirected path between **BATTERY** and **GAS TANK**:

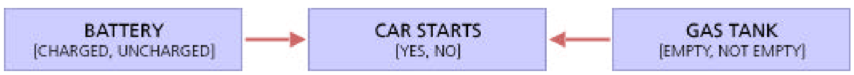

FIGURE 4220-2

Unconditionally, this path is inactive because the variable **CAR STARTS** is a collider. **BATTERY** and **GAS TANK** are d-separated given the empty set, and therefore are predicted to be unconditionally independent. As we discussed in section 7000 of Causation to Association II, however, conditioning on a direct common effect like the variable **CAR STARTS** induces dependence between its two causes, and thus **BATTERY** and **GAS TANK** are conditionally dependent on **CAR STARTS**.

Put in the language of d-separation -- **BATTERY** and **GAS TANK** are d-connected by **CAR STARTS**. The path **BATTERY** → **CAR STARTS** ← **GAS TANK** is active relative to the set {**CAR STARTS**}, because **CAR STARTS** is a collider which becomes active in virtue of being in the conditioning set.

Example 2: Heart Disease

Neglecting people above 65, it turns out that middle aged American men are the most common victims of heart disease. Women and younger people are much less likely to get it. People with Heart Disease are instructed to go on a special diet, and they often do. Thus the following variables and graph describes the causal relations between **SEX**, **AGE**, and **HEART DISEASE**.

TABLE 4220-2: VARIABLES FOR HEART DISEASE

Variables	Values
SEX	[Male, Female]
AGE	[Old, Young]
HEART DISEASE	[Yes, No]
DIET	[Special diet, Normal diet]

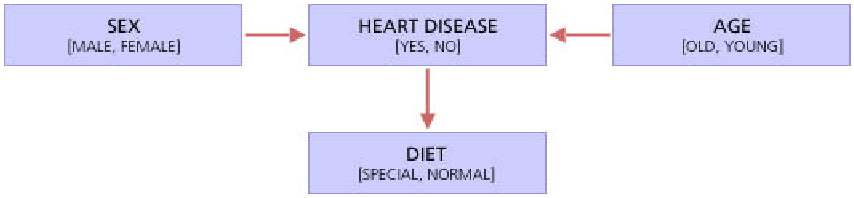

FIGURE 4220-3

SEX and **AGE** are unconditionally d-separated in this graph, thus they are predicted to be independent.

< A link to exercises in the interactive version of this module. >

Although **SEX** and **AGE** are unconditionally d-separated, they are d-connected by conditioning on **HEART DISEASE**. Thus **SEX** and **AGE** are predicted to be conditionally associated given **HEART DISEASE**. Why is this plausible?

Take a random American under 65. Suppose you know that he or she has no **HEART DISEASE**. Now suppose that I tell you this person is male. What does that extra piece of information do to your beliefs about the person's age? Knowing they are male should now make it more likely that they are young than before you knew they were male. Thus, conditional on **HEART DISEASE**, a person's sex is informative about a person's age.

We can now add clause 5:

1. X and Y are **d-connected** by **Z** just in case **any path** between Y and Y is **active** relative to **Z**.
2. A **path is active** just in case **all** the variables on the path are active.
3. If **Z** = ∅, then a variable is **active** on a path if it occurs as a **non-collider**, and is **inactive** if it occurs as a **collider**.
4. If **Z** ≠ ∅, then **Non-colliders** are **inactive** if they are in **Z**, and **active** if they are not in **Z**
5. If **Z** ≠ ∅, then **Colliders** are **active** if they are in **Z**.

This is almost the whole story. Clause 5 needs to be extended, but that is all that remains to cover.

< A link to exercises in the interactive version of this module. >

4230: Conditioning on the Effects of Colliders

So far d-separation is simple. Identify the paths between X and Y. For each path, look at the variables on the path. If a variable is a non-collider on a path, then it is active as long as it **is not** in the conditioning set, but inactive if it is. **Roughly** the opposite is true of colliders. Colliders are inactive relative to the empty set, and they can be activated by conditioning on them. They can be activated in another way, however: **by conditioning on one of their effects**.

A collider is **active** on a path if either:

+ it occurs in the conditioning set **Z**, or
+ it has an effect in the graph that is in the conditioning set **Z**.

Consider a few examples.

Example 1: Starting the Car

Again, assume that the following graph depicts the causal relations among these four variables.

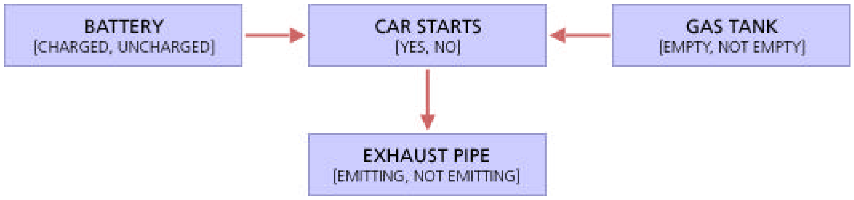

FIGURE 4230-1

Consider the only undirected path between **BATTERY** and **GAS TANK**:

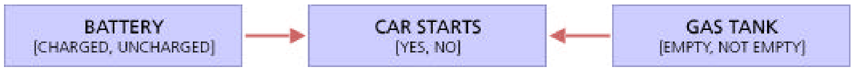

FIGURE 4230-2

Unconditionally, this path is inactive because the variable **CAR STARTS** is a collider. **BATTERY** and **GAS TANK** are conditionally associated given **CAR STARTS**, so if we include **CAR STARTS** in the conditioning set **Z**, we will activate the collider **CAR STARTS**.

BATTERY and **GAS TANK** are also conditionally associated given **EXHAUST PIPE**, however. If we know there is nothing coming out of the exhaust pipe, then telling me the battery is charged is informative about the state of the gas tank -- it must be empty.

So d-separation must tell us that relative to the set : {**EXHAUST PIPE**}, the path **BATTERY** → **CAR STARTS** ← **GAS TANK** is active. It therefore must tell us that relative to the set {**EXHAUST PIPE**}, the collider **CAR STARTS** is active. Clause 2 does this work:

A collider is active on a path if either:

+ it occurs in the conditioning set **Z**, or
+ it has an effect in the graph that is in the conditioning set **Z**.

In this case, the collider **CAR STARTS** is a cause of **EXHAUST PIPE**, which is in the conditioning set.

Example 2: Heart Disease

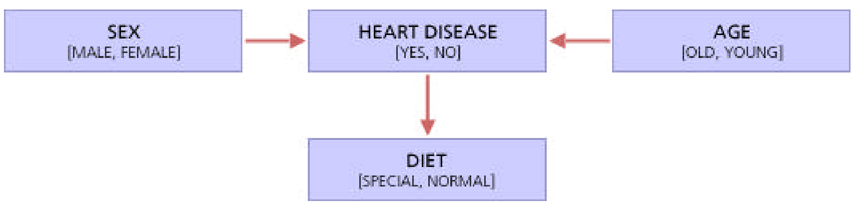

FIGURE 4230-3

The only undirected path between **SEX** and **AGE** is: **SEX** → **HEART DISEASE** ← **AGE**. Since Heart Disease is a collider, it is inactive relative to the empty set. By conditioning on **HEART DISEASE**, we activate it. By conditioning on **DIET**, however, we also activate **HEART DISEASE** on this path. Why is this plausible? For similar reasons to why its plausible that **SEX** and **AGE** are associated conditional on **HEART DISEASE**.

Take a random American under 65. Suppose you know that he or she is on no **DIET** for **HEART DISEASE**. Now suppose that I tell you this person is male. What does that extra piece of information do to your beliefs about the person's age? Knowing they are male should now make it more likely that they are young than before you knew they were male. Thus, conditional on **DIET**, a person's sex is informative about a person's age, so **SEX** and **AGE** are predicted to be conditionally associated given **DIET**.

SEX and **AGE** are unconditionally d-separated in this graph, thus they are predicted to be independent.

We can now add the rest of clause 5 to complete the theory:

1. X and Y are **d-connected** by **Z** just in case **any path** between Y and Y is **active** relative to **Z**.
2. A **path is active** just in case **all** the variables on the path are active.
3. If **Z** = ∅, then a variable is **active** on a path if it occurs as a **non-collider**, and is **inactive** if it occurs as a **collider**.
4. If **Z** ≠ ∅, then **Non-colliders** are **inactive** if they are in **Z**, and **active** if they are not in **Z**
5. If **Z** ≠ ∅, then **Colliders** are **active** if they are in **Z**, or have **an effect** in **Z**, and **inactive** otherwise.

< A link to exercises in the interactive version of this module. >

5000: The General Theory

5100: D-separation and D-connection Defined

You have now seen the entire theory of d-separation and d-connection. Remember that d-connection and d-separation are opposities. X and Y are d-separated given a set **Z** just in case they are not d-connected given **Z**, and vice versa, so we only need define one of them to effectively define both. Here again is the general definition of d-connection that we have built in this module.

1. X and Y are **d-connected** by **Z** just in case **any path** between Y and Y is **active** relative to **Z**.
2. A **path is active** just in case **all** the variables on the path are active.
3. If **Z** = ∅, then a variable is **active** on a path if it occurs as a **non-collider**, and is **inactive** if it occurs as a **collider**.
4. If **Z** ≠ ∅, then **Non-colliders** are **inactive** if they are in **Z**, and **active** if they are not in **Z**
5. If **Z** ≠ ∅, then **Colliders** are **active** if they are in **Z**, or have **an effect** in **Z**, and **inactive** otherwise.

A more compact, but equivalent definition, is as follows:

1. X and Y are **d-connected** by **Z** just in case **any path** between Y and Y is **active** relative to **Z**.
2. A **path is active** just in case **all** the variables on the path are active.
3. **Non-colliders** are **inactive** if they are in **Z**, and **active** if they are not in **Z**
4. **Colliders** are **active** if they are in **Z**, or have **an effect** in **Z**, and **inactive** otherwise.

5200: A Procedure for Computing D-separation

Although there are other, equivalent ways to define d-separation and d-connection, we have purposely defined it in a way that maps easily into a step-by-step procedure for applying the theory to any graph. In this section we give an explicit procedure.

To determine if X and Y are d-connected or d-separated by a set **Z** in a graph G:

1. List all the undirected paths between X and Y that occur in G.
2. For each path, categorize all the variables on the path as colliders or non-colliders.
3. Mark each non-collider on a path as **active** if it is **not** in **Z**, and mark it as **inactive** if it **is** in **Z**.
4. Mark each collider on a path as **active** if it is in **Z** or has an effect in **Z**, and mark it as **inactive** otherwise.
5. Mark each path as **inactive** if **any variable** on it is **inactive**, and active otherwise.
6. X and Y are d-connected by **Z** if there are any active paths. Otherwise, X and Y are d-separated by **Z**.

Lets apply this procedure to the example we suggested that you investigate in the Introduction (section 2000). Lets consider whether X and Y are d-connected or d-separated by the set **Z** = {Z2, Z3}.

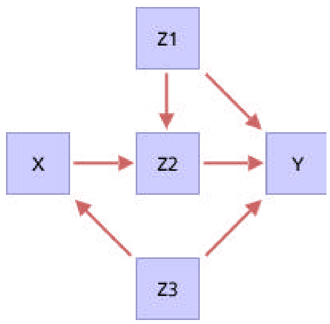

FIGURE 5200-1

Step 1

List all the undirected paths between X and Y that occur in G.

There are three undirected paths between X and Y in G:

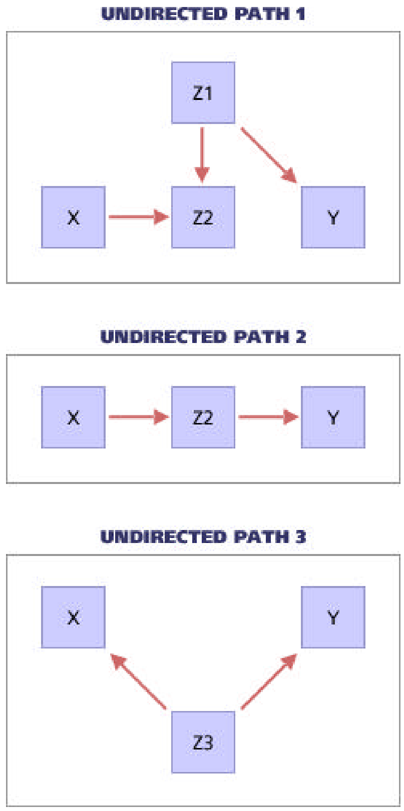

FIGURE 5200-2

Step 2

For each path, categorize all the variables on the path as colliders or non-colliders.

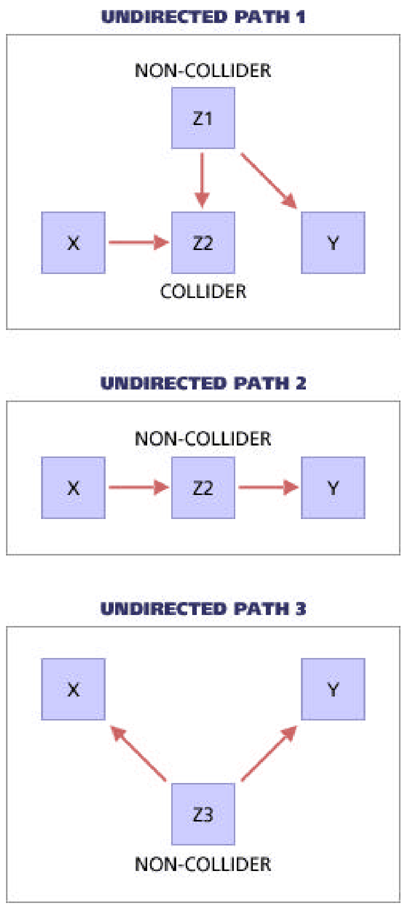

FIGURE 5200-3

Step 3

Mark each non-collider on a path as **active** if it is **not** in **Z**, and mark it as **inactive** if it **is** in **Z** {**Z2**, **Z3**}.

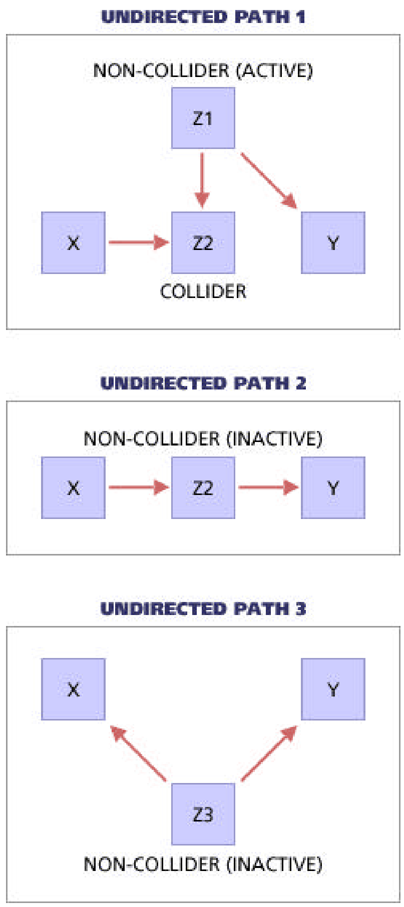

FIGURE 5200-4

In this case, **Z** = {Z2, Z3}, so the non-collider Z1 is active on path 1, Z2 is inactive on path 2, etc.

Step 4

Mark each collider on a path as **active** if it is in **Z** or has an effect in **Z** {Z2, Z3}, and mark it as **inactive** otherwise.

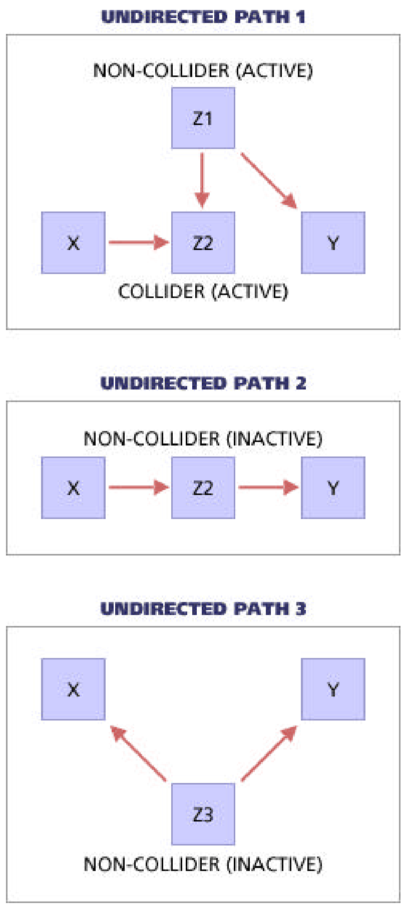

FIGURE 5200-5

Z2, which is a collider on Path 1, is active because it is in the conditioning set.

Step 5

Mark each path as **inactive** if **any variable** on it is **inactive**, and active otherwise.

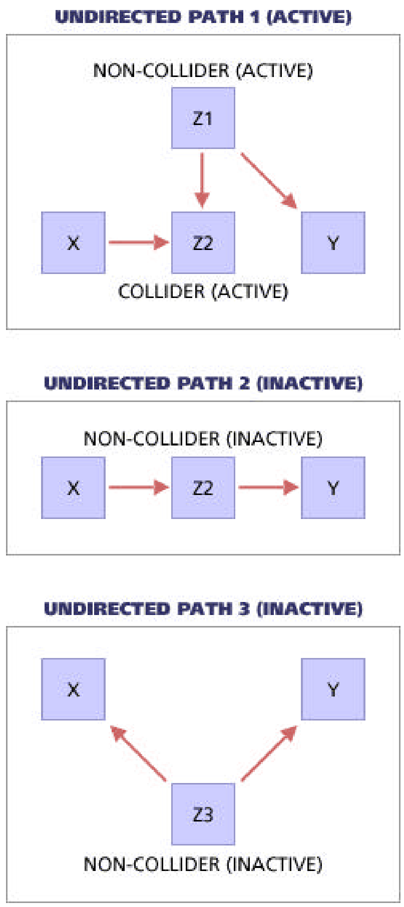

FIGURE 5200-6

All the variables on Path 1 are active, so the whole path is active. Path 2 is inactive, and path 3 is inactive.

Step 6

X and Y are d-connected by **Z** if there are any active paths. Otherwise, X and Y are d-separated by **Z**.

Path 1 is active, so X and Y are d-connected by **Z** {Z2, Z3} in this graph:

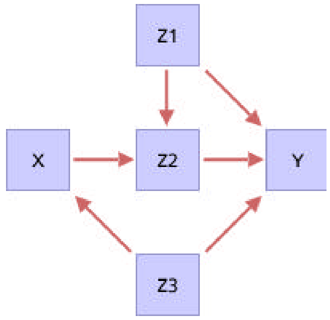

FIGURE 5200-7

Now you try. Use the same graph, but use a different conditioning set: **Z** {**Z1**, **Z2**}. Take out a piece of paper, and go through the same steps to determine if **X** and **Y** are d-connected by **Z** {**Z1**, **Z2**}:

- List all the undirected paths between **X** and **Y** that occur in G.
- For each path, categorize all the variables on the path as colliders or non-colliders.
- Mark each non-collider on a path as **active** if it is **not** in **Z**, and mark it as **inactive** if it **is** in **Z**.
- Mark each collider on a path as **active** if it is in **Z** or has an effect in **Z**, and mark it as **inactive** otherwise.
- Mark each path as **inactive** if **any variable** on it is **inactive**, and active otherwise.
- **X** and **Y** are d-connected by **Z** if there are any active paths. Otherwise, **X** and **Y** are d-separated by **Z**.

< A link to exercises in the interactive version of this module. >

6000: Summary

Causal theories can be represented qualitatively as causal graphs, which make predictions about which variables are independent and conditionally independent. For any two variables **X** and **Y** and a set of variables **Z**, a causal graph G implies that **X** _||_ **Y** |**Z** if **X** and **Y** are d-separated by **Z** in the graph G. So d-separation gives us a way to compute testable predictions about association from causal graphs. D-connection is the opposite of d-separation, and is defined as follows:

1. X and Y are **d-connected** by **Z** just in case **any path** between Y and Y is **active** relative to **Z**.
2. A **path is active** just in case **all** the variables on the path are active.
3. If **Z** = ∅, then a variable is **active** on a path if it occurs as a **non-collider**, and is **inactive** if it occurs as a **collider**.
4. If **Z** ≠ ∅, then **Non-colliders** are **inactive** if they are in **Z**, and **active** if they are not in **Z**
5. If **Z** ≠ ∅, then **Colliders** are **active** if they are in **Z**, or have **an effect** in **Z**, and **inactive** otherwise.

A more compact, but equivalent definition, is as follows.

1. X and Y are **d-connected** by **Z** just in case **any path** between Y and Y is **active** relative to **Z**.
2. A **path is active** just in case **all** the variables on the path are active.
3. **Non-colliders** are **inactive** if they are in **Z**, and **active** if they are not in **Z**
4. **Colliders** are **active** if they are in **Z**, or have **an effect** in **Z**, and **inactive** otherwise.

Procedurally, to determine if X and Y are d-connected or d-separated by a set **Z** in a graph G:

1. List all the undirected paths between X and Y that occur in G.
2. For each path, categorize all the variables on the path as colliders or non-colliders.
3. Mark each non-collider on a path as **active** if it is **not** in **Z**, and mark it as **inactive** if it **is** in **Z**.
4. Mark each collider on a path as **active** if it is in **Z** or has an effect in **Z**, and mark it as **inactive** otherwise.
5. Mark each path as **inactive** if **any variable** on it is **inactive**, and active otherwise.
6. X and Y are d-connected by **Z** if there are any active paths. Otherwise, X and Y are d-separated by **Z**.
