Deriving Testable Implications from DAGs: The d-Separation Property

The d-separation property (in which the 'd' stands for "directed") is a graphical criterion that allows one to derive testable implications from any given DAG. Here, we give a brief explanation of this property; for a more detailed yet accessible account, please see chapter 2.4 in Pearl et al.s recent textbook (1).

First, some standard notation: variables within the DAGs presented here are denoted with capital letters (e.g. X, M and Y), while sets of variables are denoted in bold (e.g. $\mathbf{Z}$ = {X,Y}). Second, an explanation of some key terms: in DAG parlance, a path between any two variables (e.g. X and Y) with at least one intervening variable (e.g. where the variable M is situated between X and Y) is described as "blocked", when one (or more) of these intervening variables is a so-called "collider" (i.e. where the arrows "collide" at one or more of those variables, as in: X→M←Y). Such paths can be "opened" by conditioning on the collider itself (i.e. on M, in this instance), or any variable that is a "descendant" of the collider in the DAG. Conditioning on an intervening variable that is not a collider will cause these otherwise "open" paths to become "blocked" (which will apply, in this example, to the three other potential paths between X and Y with M in between: X→M→Y; X←M←Y; and X←M→Y; all of these becoming "blocked" by conditioning on M).

In larger, more complex DAGs than the example based on X and Y with M in between, paths between two variables with more than one intervening variable can be blocked by blocking any of the intervening sub-paths. For example, a path X→M→W→Y can be blocked by conditioning on: $\mathbf{Z}$={M}, because this set blocks the sub-path X→M→W; $\mathbf{Z}$={W}, because this set blocks the sub-path M→W→Y; or $\mathbf{Z}$={M,W}, because this set blocks both sub-paths. Likewise, in those DAGs that contain more than one "collider" such as X→V←W→U→T←Y (where both V and T are colliders), the path between X and Y is "opened" by conditioning on the set $\mathbf{Z}$={V,T}. In this way, conditioning on some (sets of) variables can block paths (such as $\mathbf{Z}$={M}, {U} and {M,U} in the first example), or it can can open other paths (such as $\mathbf{Z}$={V,T} in the collider path example) – indeed, a set can sometimes block some paths and open others at the same time.

The d-separation property states that a set of variables, $\mathbf{Z}$, "d-separates" two variables X and Y when *all possible paths* from X to Y are blocked by the set $\mathbf{Z}$. It is this level of complexity that can make identifying "d-separation" statements challenging in larger and more complex DAGs; though in each case, testing the "d-separation" property ultimately boils down to examining all paths within a DAG that contain (at least) three variables. Formal tests of DAG-dataset consistency are then feasible because the statistical implication of a statement such as "X and Y are d-separated by $\mathbf{Z}$={M}" (which would be true of the "full mediation model") is that X and Y must be conditionally independent given $\mathbf{Z}$ (i.e. conditioned on $\mathbf{Z}$). If $\mathbf{Z}$ is the empty set (i.e. all paths between X and Y contain a collider), then the implication is simply "X and Y are statistically independent".

The R package 'dagitty' automates the application of the d-separation property to find testable implications, but it also provides a function that allows detailed investigation of individual paths. To illustrate this, we recapitulate some of the examples above in R. The code below defines the "full mediation model":

```
library( dagitty )
g <- dagitty( 'dag{ X -> M -> Y }' )
```

Now we can use the 'paths' function to determine which paths from X to Y exist (for instance), and whether these are open:

```
paths( g, "X", "Y" )
```

```
## $paths
```

```
## [1] "X -> M -> Y"
##
## $open
## [1] TRUE
```

The function returns a list with two components: 'paths', which contains the paths, and 'open', which contains a Boolean value for each paths, indicating whether that specific path is open.

To determine whether any open paths are closed by conditioning on the set **Z**={M}, we can supply that set as a fourth argument to the 'paths' function:

```
paths( g, "X", "Y", list("M") )
```

```
## $paths
## [1] "X -> M -> Y"
##
## $open
## [1] FALSE
```

The code below verifies that in our example with two colliders U and V, it is necessary to condition on *both* of these to open the path, whereas conditioning on only U or only V is not sufficient to open the path:

```
g <- dagitty( 'dag{ X -> V <- W -> U <- T <- Y }' )
paths( g, "X", "Y" )
```

```
## $paths
## [1] "X -> V <- W -> U <- T <- Y"
##
## $open
## [1] FALSE
```

```
paths( g, "X", "Y", list("V") )
```

```
## $paths
## [1] "X -> V <- W -> U <- T <- Y"
##
## $open
## [1] FALSE
```

```
paths( g, "X", "Y", list("U") )
```

```
## $paths
## [1] "X -> V <- W -> U <- T <- Y"
##
## $open
## [1] FALSE
```

```
paths( g, "X", "Y", list("U","V") )
```

```
## $paths
## [1] "X -> V <- W -> U <- T <- Y"
##
## $open
## [1] TRUE
```

We conclude this brief explanation by showing that the same function is applicable to more complex DAGs, in which several paths between two variables exist. For instance, the code below examines all paths between "team motivation" (TM) and "performance of warm-up exercises"

(WUE) from the DAG in Figure 1A in the main paper, and shows that all of these paths are indeed closed by the set $\mathbf{Z}$={*C*} ("coach"):

```
g <- dagitty( 'dag{
  C -> {FL TM}
  TM -> {PI IGP}
  CS -> {IGP PI}
  FL -> {PGP NMF}
  NMF -> {IGP I}
  PGP -> WUE -> IGP -> I
}' )
paths( g, "TM", "WUE", list("C")  )
```

```
## $paths
## [1] "TM -> IGP -> I <- NMF <- FL -> PGP -> WUE"
## [2] "TM -> IGP <- NMF <- FL -> PGP -> WUE"
## [3] "TM -> IGP <- WUE"
## [4] "TM -> PI <- CS -> IGP -> I <- NMF <- FL -> PGP -> WUE"
## [5] "TM -> PI <- CS -> IGP <- NMF <- FL -> PGP -> WUE"
## [6] "TM -> PI <- CS -> IGP <- WUE"
## [7] "TM <- C -> FL -> NMF -> I <- IGP <- WUE"
## [8] "TM <- C -> FL -> NMF -> IGP <- WUE"
## [9] "TM <- C -> FL -> PGP -> WUE"
##
## $open
## [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
```

This reveals that there are a total of 9 paths from TM to WUE in the DAG, and that all of these are closed by the set $\mathbf{Z}$={*C*}. Hence, the DAG implies that TM and WUE *must* be conditionally independent given C.

References

[1] Judea Pearl, Nicholas P. Jewell, and Madelyn Glymour. *Causal Inference in Statistics: A Primer*. Wiley, 2016.