

Causal Search Using Graphical Causal Models

Kevin D. Hoover
Duke University

Lecture 2 **Causal Search Algorithms**

I. Recapitulation

Causal Markov Condition

*A variable in a graph is, conditional on its parents, probabilistically independent of all other variables that are neither its parents (**N**) nor its descendants – i.e., if **V** = variable of interest; **P** = parents; **D** = descendants; **N** = nonparents, nondescendants, then*

$$P(V|\mathbf{P}, \mathbf{N}) = P(V|\mathbf{P})$$

$$P(V|\mathbf{P}, \mathbf{D}) \neq P(V|\mathbf{P})$$

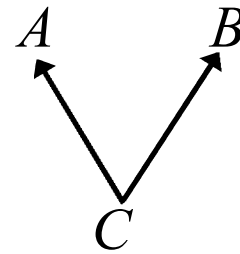
- Causal search algorithms systematically investigate patterns of conditional dependence and apply the Causal Markov Condition to reconstruct the graph of the data-generating process.

Examples

Screen

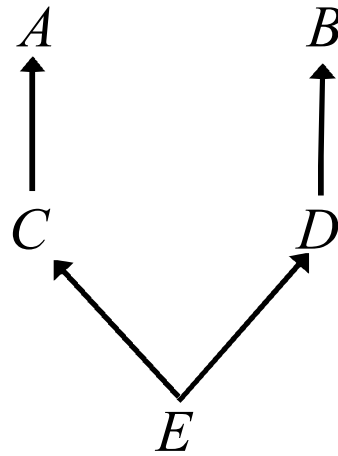


Common Cause



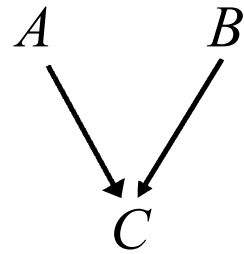
B is independent of *A* conditional on *C*

Screens with Remote Common Causes



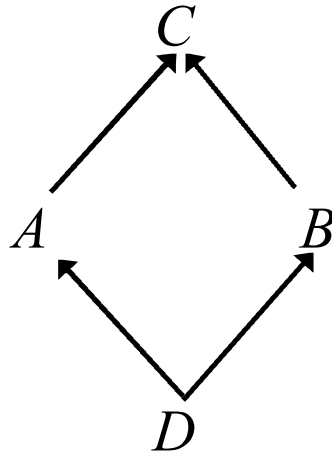
A and B are independent conditional on C and D .

Unshielded Collider



A and *B* are independent unconditionally;
but are dependent conditional on *C*

Unshielded Collider



A and B are independent conditional on D ;
but are dependent conditional on D and C .

II. The SGS and PC Algorithms

1. BACKGROUND

- Many algorithms, including
 - latent variables
 - cyclical graphs
 - large numbers of variables (address curse of dimensionality)

- SGS and PC algorithms closely related
 - SGS from **S**pirtes, **G**lymour, and **S**cheines
 - PC from **P**eter Sprites and **C**lark Glymour

- SGS and PC structured similarly.

2. STRUCTURE OF THE PC AND SGS ALGORITHMS

A. Skeleton Identification

1. *General Graph.* Start with densely connected, undirected graph – i.e., every node connected to every other node by an undirected edge. In which each variable is assumed to be connected by an undirected causal link.
2. *Edge Elimination.* Test for conditional independence starting with zero-order (i.e., unconditional) independence and continuing with first-order (conditioning on single variables), second-order (two variables), and higher orders until further conditioning is impossible. Eliminate any edges that are conditionally independent.

B. Orientation

(i) Statistical Orientation

- 3. Identifying Unshielded Colliders.* For each conditionally uncorrelated pair of variables (i.e., ones not connected by a direct edge) that are connected through a third variable, test conditional independence on that third variable. If conditionally dependent, the third variable is an unshielded collider. Orient the edges as pointing into the unshielded collider.

(ii) Logical Orientation

4. *Avoid Spurious Unshielded Colliders.* If there are any pairs A and C that are not directly connected but are connected as

$$A \rightarrow B - C,$$

orient the second edge toward C , so that the triple is

$$A \rightarrow B \rightarrow C.$$

Logic: If the edge is oriented toward B, then B is an unshielded collider on the path ABC. If that were true, it should have been discovered already in step 3.

5. *Avoid Cycles.* If there is a pair of variables, A and B connected both by an undirected edge and by a directed path, starting at A , through one or more other variables to B , orient the undirected edge as

$$A \rightarrow B.$$

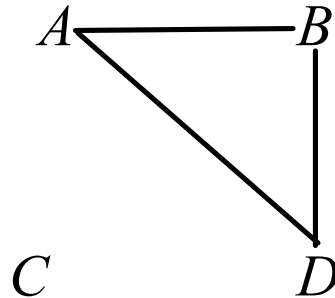
Logic: Since there is one directed path runs from A to B , directing the undirected edge from B to A would complete a cycle. If the graph is acyclical ex hypothesi, then the edge must be oriented A to B .

3. Difference between the SGS and the PC Algorithm

- In step 2 (edge elimination), the SGS algorithm tests every possible conditioning set at every order of conditioning.
- In step 2, the PC algorithm tests only conditioning sets involving variables that are connected by direct or indirect paths to the variable under test.

Example

Graph after zero-order (unconditional) independence tests and elimination.



- For the SGS algorithm, the conditioning set for the first-order conditional independence of A is $\{B, C, D\}$.
- For the PC algorithm, the conditioning set for the first-order conditional independence of A is $\{B, D\}$.

- Advantage of SGS algorithm: may be robust to nonlinearities.
- Advantage of the PC algorithm: avoids curse of dimensionality (exponential growth in number of independence relationships that must be tested).
- PC: most commonly used algorithm.

4. Tests of Conditional Independence.

- A variety of tests of statistical independence exist. Algorithm may incorporate any of them.
- Most common: conditional correlation test:

$$\rho_{x,y|z} = \frac{-\mathcal{R}_{xy}}{\sqrt{\mathcal{R}_x \mathcal{R}_y}}$$

where \mathcal{R}_{ab} = cofactor of the zero-order correlation matrix \mathbf{R} .

Fisher's z-statistic:

$$z = \frac{0.5 \log(1 + \rho_{X,Y|Z})}{(1 - \rho_{X,Y|Z})} \sim \text{approximately } N\left(0, \frac{1}{T-3}\right) \text{ under the null}$$

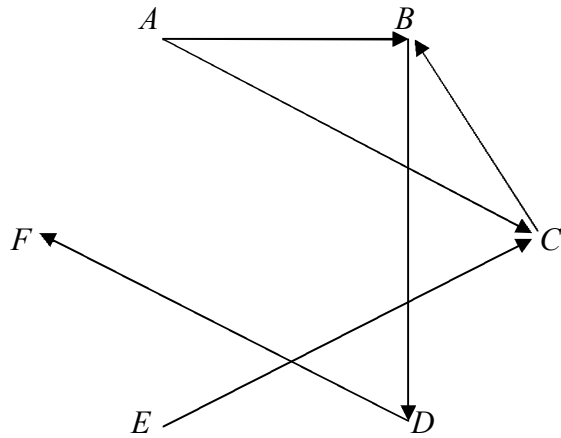
$$\rho_{X,Y|Z} = 0.$$

3. EXAMPLE OF A CAUSAL SEARCH

Example of PC Algorithm

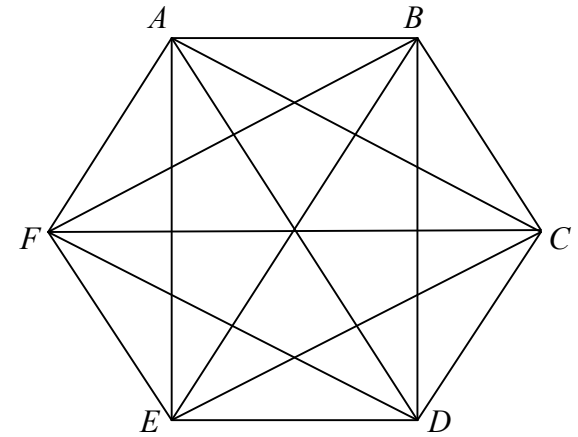
True Structure

Data-generating graph determines what the algorithm finds



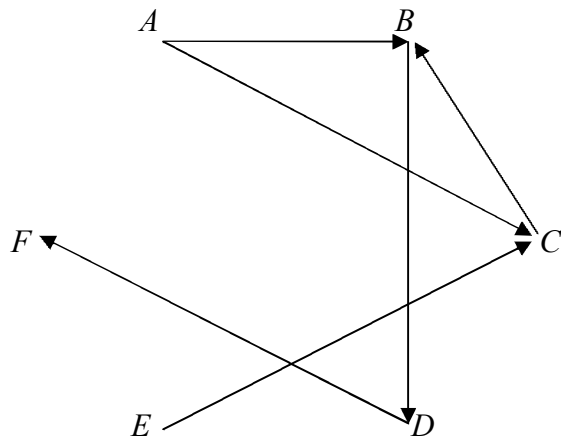
Step 1. *General Graph*

Every variable is connected by an undirected edge to every other.



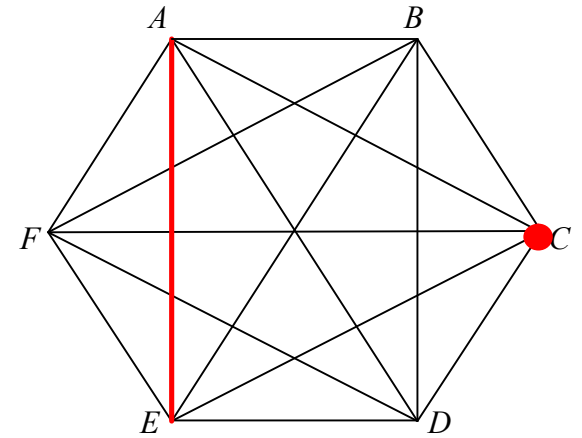
Example 3 (continued)

True Structure



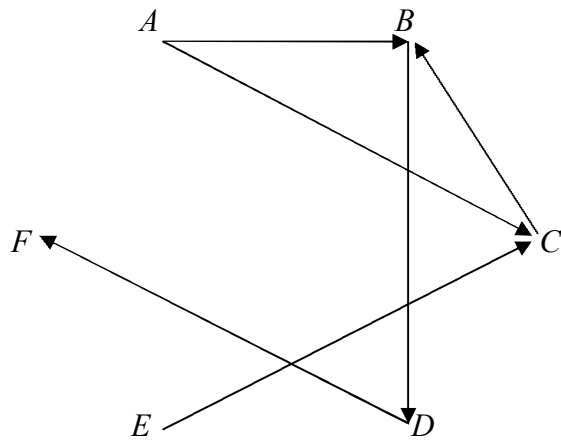
Step 2. *Edge Elimination: zero-order.*

Zero-order independence: collider C on path ACE blocks correlation between A and E.



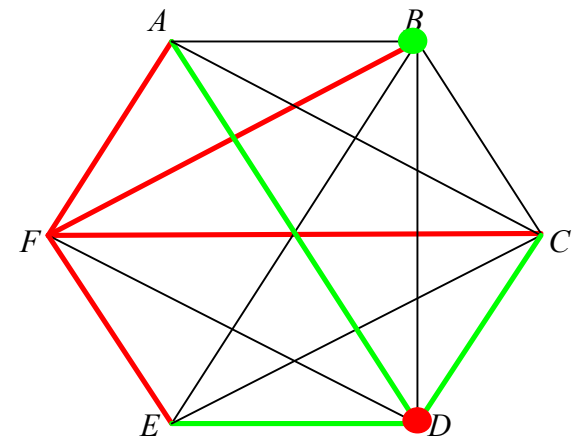
Example 3 (continued)

True Structure



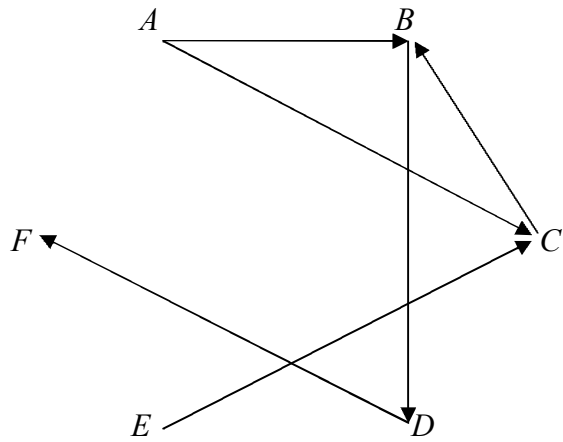
Step 2. *Edge Elimination: 1st-order.*

First order independence: *D* screens all edges to *F*, except *DE* itself. *B* screens all edges to *D* except *DF* and *DB* itself.



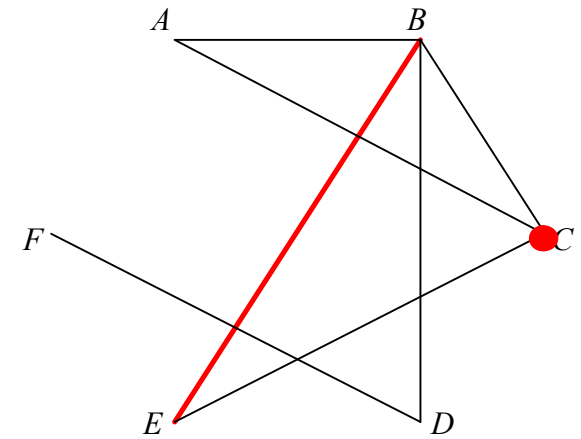
Example 3 (continued)

True Structure



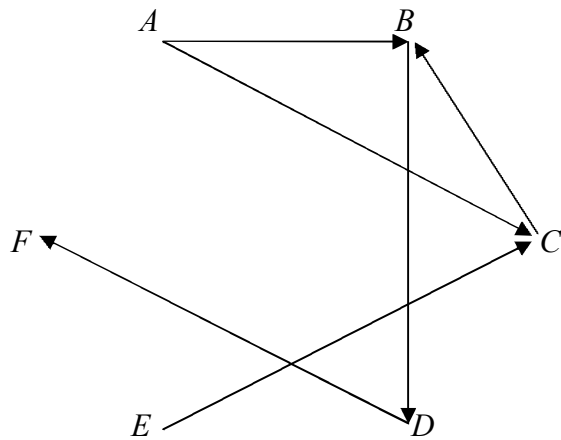
Step 2. *Edge Elimination: 2nd-order.*

Second order independence: *A* and *C* together screen *B* from *E*.
(There are no third or higher orders of conditional independence in this case.)



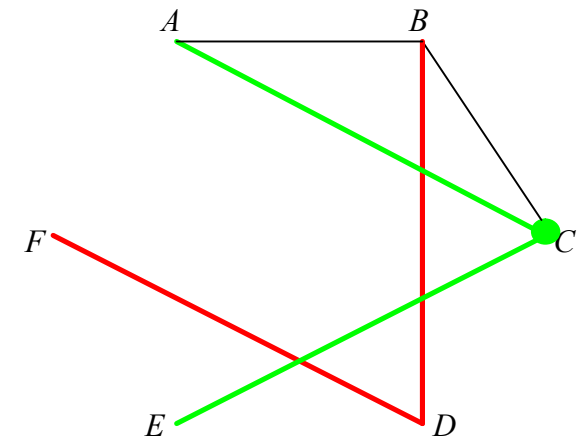
Example 3 (continued)

True Structure



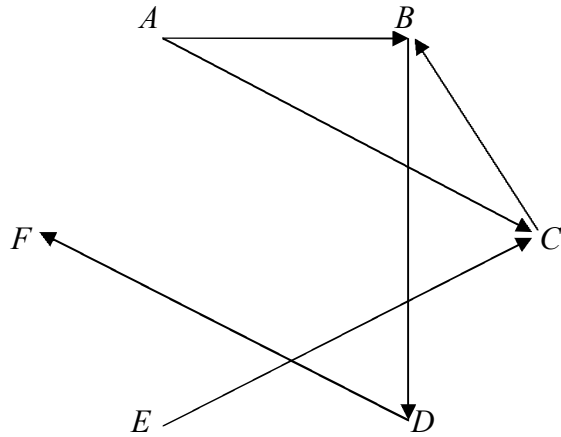
Step 3. *Identify Unshielded Colliders*

Identification of unshielded colliders: *C* on *ACE* is an unshielded collider; *BDF* candidate but not an unshielded collider. Other paths (not highlighted) are also candidates but not unshielded colliders: *BCE*, *DBA*, *CBD*.



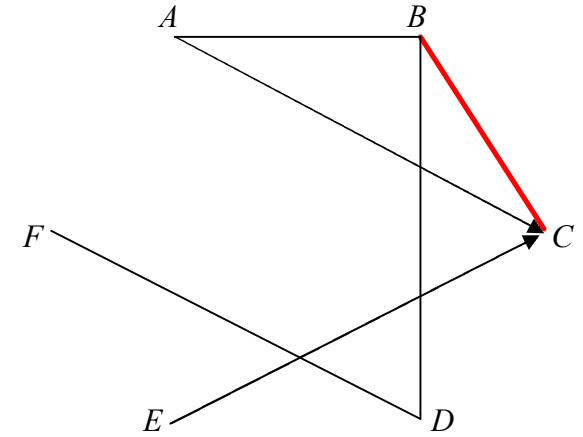
Example 3 (continued)

True Structure



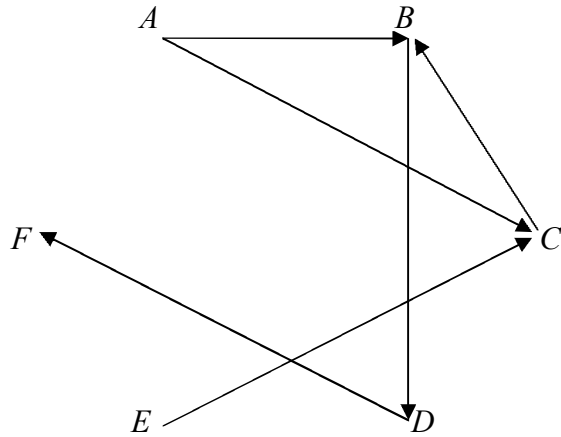
Step 4. *Avoid Unshielded Colliders*

Orientation by non-unshielded collider rule: if $B \rightarrow C$, then C would be an unshielded collider on BCE . Since it is not, orient edge as $C \rightarrow B$.



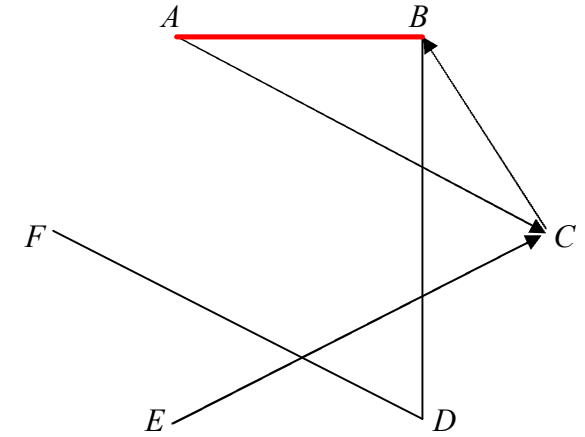
Example 3 (continued)

True Structure



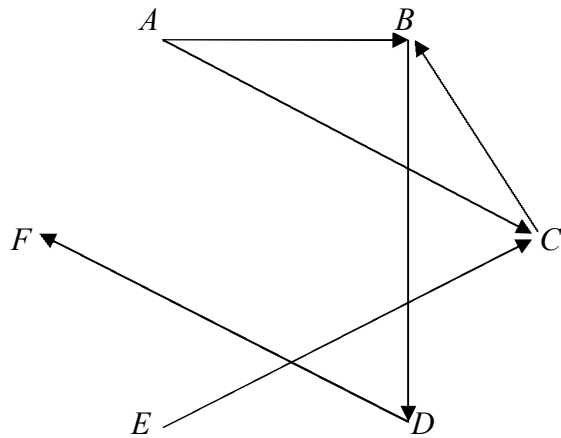
Step 5. Avoid Cycles

(vii) Orientation by non-cycle rule: if $B \rightarrow A$, then there would be a cycle $B \rightarrow A \rightarrow C \rightarrow B$. Therefore, orient edge $A \rightarrow B$



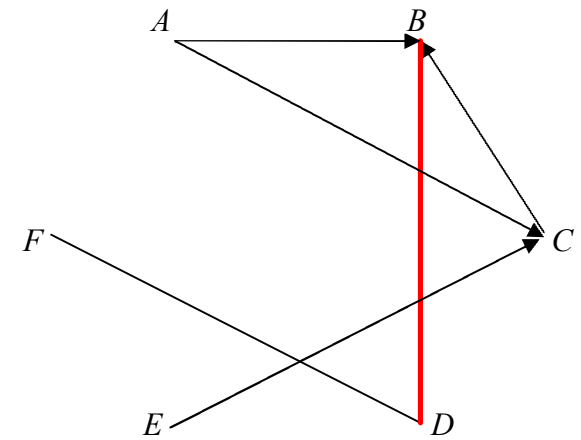
Example 3 (continued)

True Structure



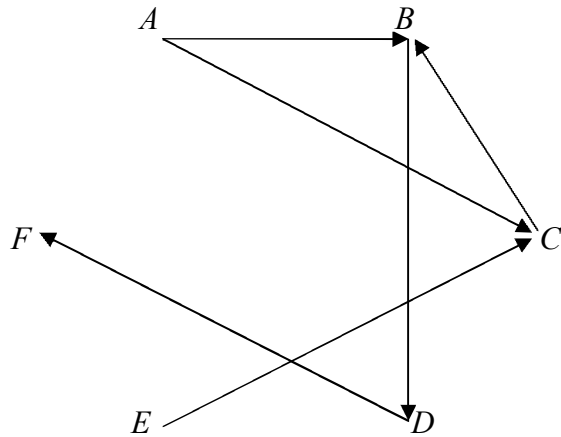
Step 4. (again) *Avoid Unshielded*

Colliders Orientation by non-unshielded collider rule: if $D \rightarrow B$, then B would be an unshielded collider on CBD . Since it is not, orient edge as $B \rightarrow D$.



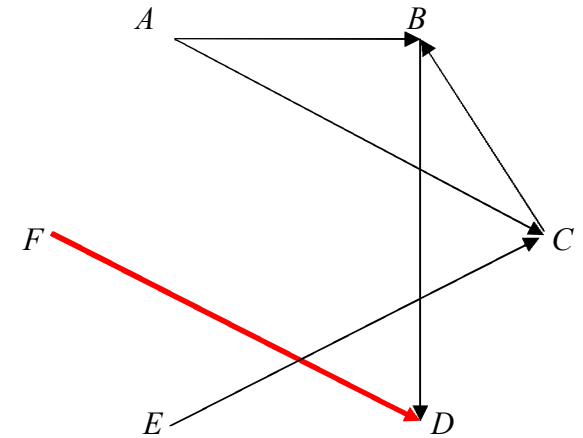
Example 3 (continued)

True Structure



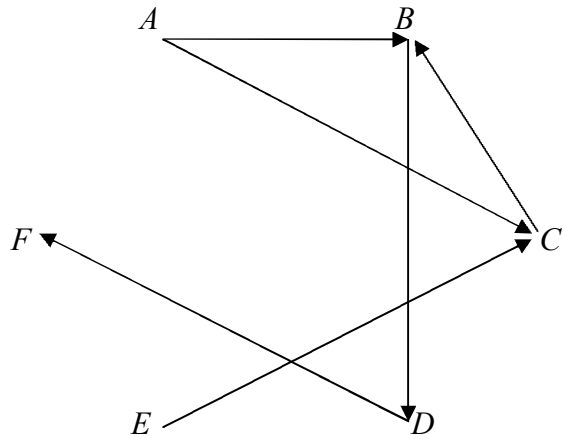
Step 4. (yet again) *Avoid Unshielded Colliders*

Orientation by non-unshielded collider rule: if $F \rightarrow D$, then D would be an unshielded collider on BBF . Since it is not, orient edge as $D \rightarrow F$.



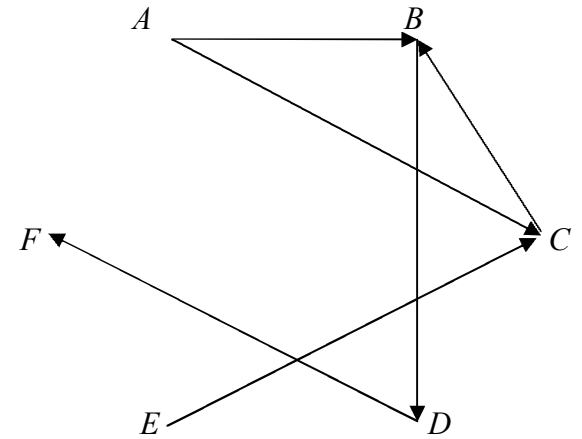
Example 3 (continued)

True Structure



Final Selected Graph

Successfully recovers true structure. May not always happen.



III. Demonstration of Tetrad IV

- *Tetrad*: family of software for causal modeling originating with Peter Spirtes, Clark Glymour, and Richard Scheines at Carnegie-Mellon University.
- Carnegie-Mellon University *TETRAD Project: Causal Models and Statistical Data* website:
<http://www.phil.cmu.edu/projects/tetrad/tetrad4.html>
- *Tetrad IV*: latest version; freely downloadable, interactive Java program.

TETRAD IV

DEMONSTRATION

IV. Application to the Structural Vector Autoregression (SVAR)

- Suppose one SVAR is the true data-generating process:

$$\mathbf{A}\mathbf{Y} = \mathbf{E},$$

where the covariance matrix $\mathbf{\Omega} = E(\mathbf{E}\mathbf{E}')$ is diagonal

- Search algorithms are not directly applicable to time-dependent data.
- Swanson and Granger (1997), however, propose to use the standard VAR to filter the time-dependence out of the data.
- Let \mathbf{A}_0 be the matrix of the zero-order terms of the matrix \mathbf{A}
 - typical element of \mathbf{A}_0 is $A_{ij}(0)$
 - \mathbf{A}_1 contains the higher order terms such that $\mathbf{A} = \mathbf{A}_1 + \mathbf{A}_0$.

$$\mathbf{A}_0 \mathbf{Y} = \mathbf{A}_1 \mathbf{Y} + \mathbf{E}$$

$$\mathbf{A}_0 \mathbf{Y}_t = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & a_{WX} & 0 \\ a_{XV} & 0 & 1 & 0 \\ 0 & 0 & a_{ZX} & 1 \end{bmatrix} \begin{bmatrix} V_t \\ W_t \\ X_t \\ Z_t \end{bmatrix}$$

$$= \begin{bmatrix} A_{VV}(L) & A_{VW}(L) & A_{VX}(L) & A_{VZ}(L) \\ A_{WV}(L) & A_{WW}(L) & A_{WX}(L) & A_{WZ}(L) \\ A_{XV}(L) & A_{XW}(L) & A_{XX}(L) & A_{XZ}(L) \\ A_{ZV}(L) & A_{ZW}(L) & A_{ZX}(L) & A_{ZZ}(L) \end{bmatrix} \begin{bmatrix} V_{t-1} \\ W_{t-1} \\ X_{t-1} \\ Z_{t-1} \end{bmatrix} + \begin{bmatrix} \varepsilon_V \\ \varepsilon_W \\ \varepsilon_X \\ \varepsilon_Z \end{bmatrix} = \mathbf{A}_1 \mathbf{Y}_{t-1} + \mathbf{E}$$

and

$$\mathbf{\Omega} = E(\mathbf{EE}') = E \begin{bmatrix} \varepsilon_V \varepsilon_V & \varepsilon_V \varepsilon_W & \varepsilon_V \varepsilon_X & \varepsilon_V \varepsilon_Z \\ \varepsilon_W \varepsilon_V & \varepsilon_W \varepsilon_W & \varepsilon_W \varepsilon_X & \varepsilon_W \varepsilon_Z \\ \varepsilon_X \varepsilon_V & \varepsilon_X \varepsilon_W & \varepsilon_X \varepsilon_X & \varepsilon_X \varepsilon_Z \\ \varepsilon_Z \varepsilon_V & \varepsilon_Z \varepsilon_W & \varepsilon_Z \varepsilon_X & \varepsilon_Z \varepsilon_Z \end{bmatrix} = \begin{bmatrix} \sigma_V^2 & 0 & 0 & 0 \\ 0 & \sigma_W^2 & 0 & 0 \\ 0 & 0 & \sigma_X^2 & 0 \\ 0 & 0 & 0 & \sigma_Z^2 \end{bmatrix}$$

The standard VAR:

$$\mathbf{Y} = \mathbf{B}\mathbf{Y} + \mathbf{U}$$

which is related to the SVAR as

$$\mathbf{Y} = \mathbf{A}_0^{-1}\mathbf{A}_1\mathbf{Y} + \mathbf{A}_0^{-1}\mathbf{E} = \mathbf{B}\mathbf{Y} + \mathbf{U}$$

- \mathbf{B} is a conformable square matrix whose terms are polynomials in the lag operator – e.g., $B_{ij}(L)$.
- \mathbf{B} differs from \mathbf{A} in that $\mathbf{B}_0 = [B_{ij}(0)] = \mathbf{0}$.
- \mathbf{U} differs from \mathbf{E} in that $\mathbf{\Sigma} = E(\mathbf{U}\mathbf{U}')$ is not diagonal.

$$\mathbf{Y}_t = \begin{bmatrix} V_t \\ W_t \\ X_t \\ Z_t \end{bmatrix} = \begin{bmatrix} B_{VV}(L) & B_{VW}(L) & B_{VX}(L) & B_{VZ}(L) \\ B_{WV}(L) & B_{WW}(L) & B_{WX}(L) & B_{WZ}(L) \\ B_{XV}(L) & B_{XW}(L) & B_{XX}(L) & B_{XZ}(L) \\ B_{ZV}(L) & B_{ZW}(L) & B_{ZX}(L) & B_{ZZ}(L) \end{bmatrix} \begin{bmatrix} V_{t-1} \\ W_{t-1} \\ X_{t-1} \\ Z_{t-1} \end{bmatrix} + \begin{bmatrix} u_V \\ u_W \\ u_X \\ u_Z \end{bmatrix} = \mathbf{B}\mathbf{Y}_{t-1} + \mathbf{U}$$

and

$$\mathbf{\Sigma} = E(\mathbf{U}\mathbf{U}') = E \begin{bmatrix} u_V u_V & u_V u_W & u_V u_X & u_V u_Z \\ u_W u_V & u_W u_W & u_W u_X & u_W u_Z \\ u_X u_V & u_X u_W & u_X u_X & u_X u_Z \\ u_Z u_V & u_Z u_W & u_Z u_X & u_Z u_Z \end{bmatrix} = \begin{bmatrix} \sigma_V^2 & \sigma_{VW} & \sigma_{VX} & \sigma_{VZ} \\ \sigma_{WV} & \sigma_W^2 & \sigma_{WX} & \sigma_{WZ} \\ \sigma_{XV} & \sigma_{XW} & \sigma_X^2 & \sigma_{XZ} \\ \sigma_{ZV} & \sigma_{ZW} & \sigma_{ZX} & \sigma_Z^2 \end{bmatrix}$$

- Since $\hat{\mathbf{U}}$ is effectively \mathbf{Y} filtered of its dynamics, we can apply causal search algorithm to $\hat{\mathbf{U}}$. The causal order of $\hat{\mathbf{U}}$ is the causal order of \mathbf{A}_0 .
- How effective is causal search?

V. The Effectiveness of Causal Search: Monte Carlo Results

1. Monte Carlo Design

(details are found in Demiralp and Hoover in the Oxford Bulletin of Economics and Statistics, 2003).

1. SVAR's with different contemporaneous causal orderings (\mathbf{A}_0) are simulated.
2. Each SVAR has the same lag structure (length and parameterization).
3. For each causal ordering, 50,000 realizations are generated
 - each 500 time-series observations long;
 - independent normal errors;
 - the non-zero coefficients of \mathbf{A}_0 (i.e., α_{ij}) drawn at random for each realization from a uniform distribution such that z-statistics range over roughly 0 to 9.

4. At each realization, the estimated $\hat{\mathbf{U}}$ is retained and treated as the filtered \mathbf{Y} .
5. Evaluation refers to a *reference graph*
 - not to the true graph, but to the graph with undirected edges placed to identify the equivalence class of the true graph;
 - coincides with true graph when the equivalence class is singular.

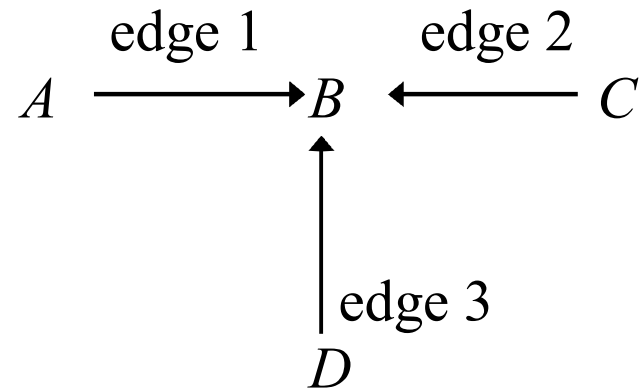
6. Every edge is evaluated on each realization. Possible outcomes:
- i. *Correct*: the edge is present and oriented the same way in both graphs or it is absent in both graphs;
 - ii. *Committed*: the edge is absent in the reference graph but present in the selected graph.
 - iii. *Omitted*: the edge is present in the reference graph but absent in the selected graph.
 - iv. *Reversed*: the edge is present in both graphs, but points in opposite directions.
 - v. *Unresolved*: the edge is oriented in the reference graph and, although present, cannot be oriented in the selected graph.
 - vi. *Overdetermined*: the edge cannot be oriented in the reference graph, but is oriented in the selected graph.

Errors fall into two groups:

- *Errors of commission*: outcome (ii) can occur only if a edge is missing in the true (and, therefore, reference) graph.
- *Edge errors*: outcomes (iii) through (vi) can occur only if a edge is present in the reference graph

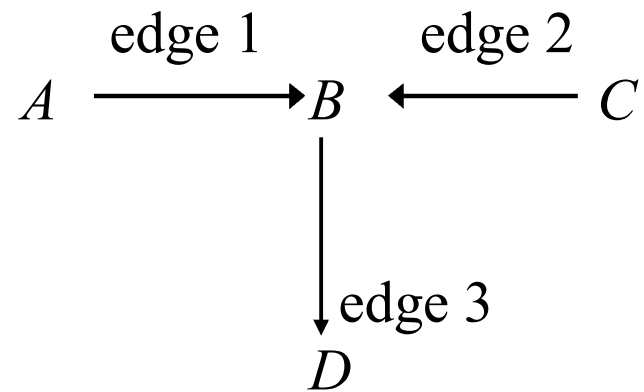
The Causal Orders

Model 1



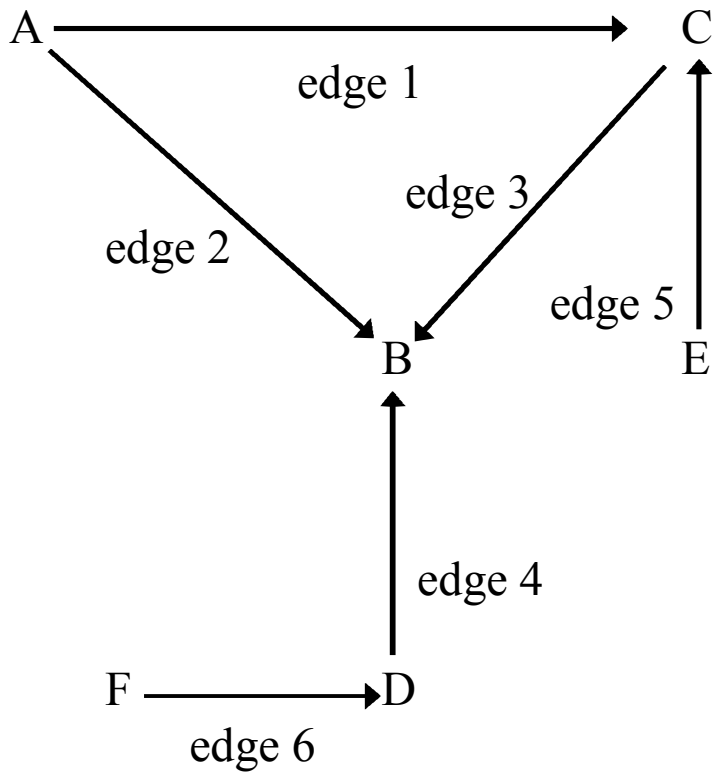
$$\mathbf{A}_0 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ \alpha_{210} & 1 & \alpha_{230} & \alpha_{240} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

Model 2

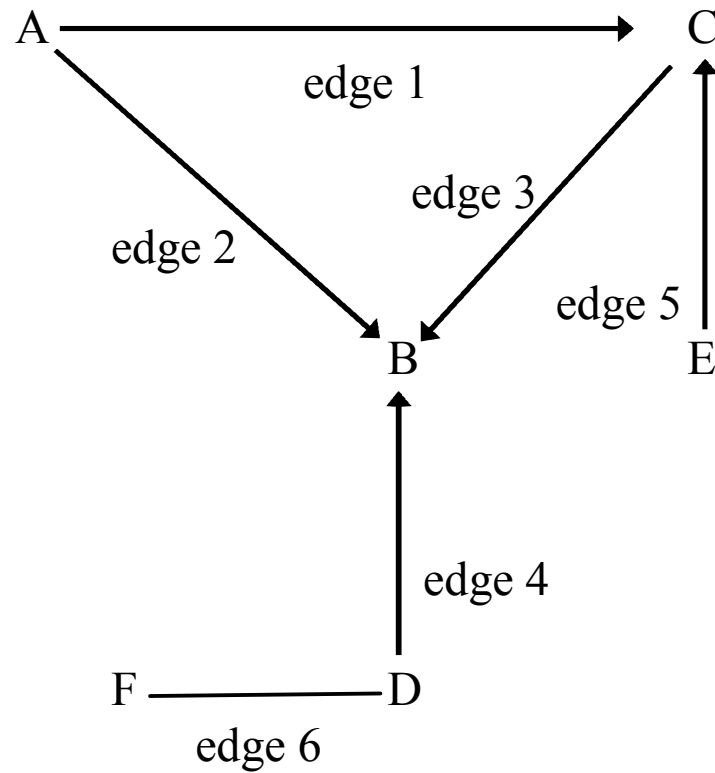


$$\mathbf{A}_0 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ \alpha_{210} & 1 & \alpha_{230} & 0 \\ 0 & 0 & 1 & 0 \\ 0 & \alpha_{420} & 0 & 1 \end{bmatrix},$$

Model 3 Data-generating Graph

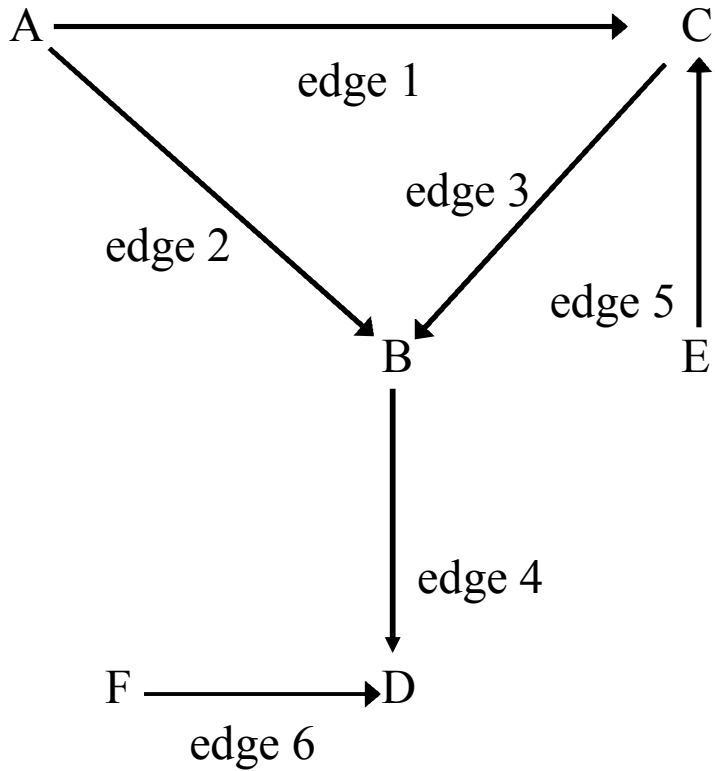


Model 3 Reference Graph

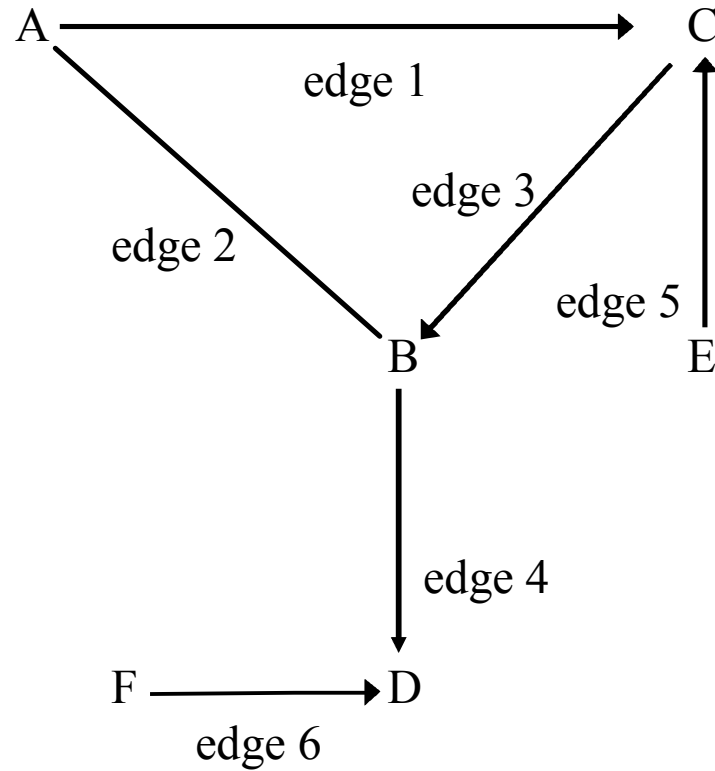


$$\mathbf{A}_0 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ \alpha_{210} & 1 & \alpha_{230} & \alpha_{240} & 0 & 0 \\ \alpha_{310} & 0 & 1 & 0 & \alpha_{350} & 0 \\ 0 & 0 & 0 & 1 & 0 & \alpha_{460} \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Model 4 Data-generating Graph



Model 4 Reference Graph



$$\mathbf{A}_0 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ \alpha_{210} & 1 & \alpha_{230} & 0 & 0 & 0 \\ \alpha_{310} & 0 & 1 & 0 & \alpha_{350} & 0 \\ 0 & \alpha_{420} & 0 & 1 & 0 & \alpha_{460} \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

2. Monte Carlo Simulation Results

Model 1 simulation outcomes by signal strength

<i>Signal strength</i> [†]		<i>Outcomes as fraction of possible outcomes</i>						
<i>By link</i> [‡]	<i>Average</i> [*]	<i>Link errors</i>						
		<i>Correct</i>	<i>Skeleton correct</i>	<i>Committed</i>	<i>Omitted</i>	<i>Reversed</i>	<i>Unresolved</i>	<i>Total</i>
<i>LLL</i>	1.00	0.46	0.60	0.10	0.71	0.00	0.28	0.99
<i>MLL</i>	1.83	0.47	0.71	0.08	0.50	0.00	0.47	0.98
<i>HLL</i>	3.00	0.48	0.71	0.07	0.52	0.01	0.45	0.97
<i>MLM</i>	2.67	0.50	0.81	0.07	0.31	0.02	0.60	0.93
<i>MMM</i>	3.50	0.54	0.91	0.06	0.11	0.07	0.67	0.84
<i>HML</i>	3.83	0.55	0.82	0.07	0.29	0.03	0.53	0.85
<i>HMM</i>	4.67	0.63	0.93	0.07	0.08	0.11	0.50	0.69
<i>HHL</i>	5.00	0.65	0.83	0.05	0.27	0.03	0.32	0.62
<i>HHM</i>	5.83	0.73	0.95	0.06	0.04	0.15	0.29	0.49
<i>HHH</i>	7.00	0.83	0.98	0.05	0.00	0.15	0.13	0.29

[†]Signal strength measured as the expected z -statistic.

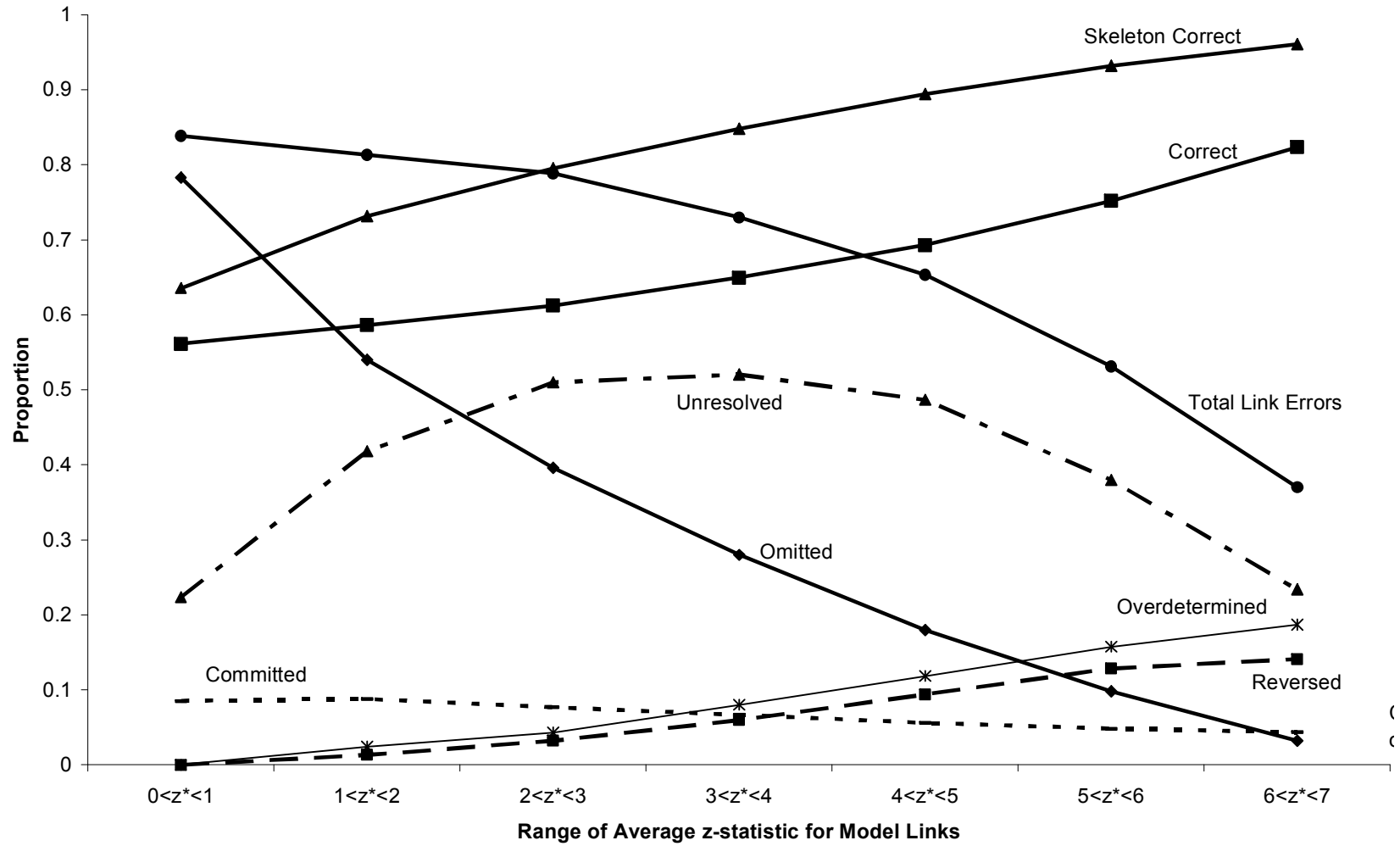
[‡] L = low ($0 \leq z^* < 2$), M = medium ($2 \leq z^* < 5$), H = high ($5 \leq z^* \leq 9$); each letter indicates one link (e.g., *HML* means link 1 is high, link 2 medium, link 3 low).

^{*}Average expected z -statistic over three links.

Model 2 simulation outcomes by signal strength

<i>Signal Strength</i> [†]		<i>Outcomes as fraction of possible outcomes</i> [*]										
(1)	(2)	(3)	(4)	(5)	<i>Link errors for links 1 and 2</i>				<i>Link errors for link 3</i>			
					(6)	(7)	(8)	(9)	(10)	(11)	(12)	(13)
<i>By link</i> [‡]	<i>Average</i> [*]	<i>Skeleton</i>										
		<i>Correct</i>	<i>correct</i>	<i>Committed</i>	<i>Omitted</i>	<i>Reversed</i>	<i>Unresolved</i>	<i>Total</i>	<i>Omitted</i>	<i>Reversed</i>	<i>Unresolved</i>	<i>Total</i>
<i>LLL</i>	1.00	0.46	0.59	0.09	0.72	0.00	0.28	1.00	0.76	0.00	0.23	1.00
<i>LLM</i>	1.83	0.46	0.68	0.10	0.43	0.00	0.55	0.98	0.78	0.01	0.21	1.00
<i>MLL</i>	1.83	0.46	0.68	0.09	0.42	0.01	0.56	0.99	0.77	0.01	0.21	1.00
<i>LLH</i>	3.00	0.47	0.70	0.09	0.37	0.01	0.59	0.97	0.78	0.01	0.20	0.99
<i>HLL</i>	3.00	0.47	0.70	0.08	0.39	0.01	0.58	0.98	0.79	0.01	0.20	1.00
<i>MML</i>	2.67	0.47	0.79	0.08	0.44	0.01	0.51	0.97	0.12	0.04	0.83	0.99
<i>HML</i>	3.83	0.48	0.81	0.07	0.40	0.01	0.55	0.96	0.12	0.05	0.81	0.98
<i>HHL</i>	5.00	0.48	0.83	0.06	0.41	0.01	0.54	0.96	0.00	0.05	0.93	0.98
<i>MLM</i>	2.67	0.49	0.79	0.08	0.11	0.01	0.80	0.93	0.79	0.01	0.19	0.99
<i>MMM</i>	3.50	0.51	0.90	0.08	0.13	0.03	0.75	0.90	0.12	0.06	0.76	0.94
<i>HLM</i>	3.83	0.53	0.81	0.08	0.07	0.01	0.72	0.80	0.80	0.02	0.15	0.96
<i>MLH</i>	3.83	0.54	0.81	0.07	0.06	0.01	0.73	0.80	0.79	0.01	0.16	0.96
<i>MMH</i>	4.67	0.57	0.93	0.06	0.06	0.03	0.70	0.79	0.13	0.06	0.66	0.85
<i>HMM</i>	4.67	0.57	0.92	0.06	0.08	0.04	0.66	0.78	0.11	0.08	0.65	0.84
<i>HHM</i>	5.83	0.57	0.94	0.05	0.09	0.04	0.66	0.79	0.00	0.08	0.74	0.82
<i>HLH</i>	5.00	0.66	0.83	0.06	0.00	0.01	0.46	0.47	0.81	0.01	0.09	0.92
<i>HMH</i>	5.83	0.73	0.95	0.06	0.00	0.02	0.43	0.45	0.13	0.07	0.37	0.57
<i>HHH</i>	7.00	0.74	0.97	0.06	0.00	0.04	0.42	0.45	0.00	0.06	0.42	0.48

Figure 5. Outcomes by Average Signal Strength for Model 3



VI. A Bootstrap Procedure for Evaluating Causal Graphs

- Monte Carlo simulations not general.
- Central question: *How reliable is this graph?*
- Addressed in Demiralp, Hoover, and Perez, *Oxford Bulletin of Economics and Statistics*, 2008, forthcoming.

➤ Bootstrap Procedure:

1. Estimate ordinary VAR and retain $\hat{\mathbf{U}}_t$.
2. Draw columns with replacement from $\hat{\mathbf{U}}_t$;
3. Use these bootstrap errors and the estimated coefficients ($\hat{\mathbf{B}}(L)$) of the VAR to generate simulated data:

$$\tilde{\mathbf{Y}}_t = \hat{\mathbf{B}}(L)\tilde{\mathbf{Y}}_{t-1} + \tilde{\mathbf{U}}_t.$$

4. Filter the simulated data ($\tilde{\mathbf{Y}}_t$) and run the PC algorithm, recording edge-by-edge results;
5. Repeat a large number of times.

➤ How well does the bootstrap procedure work?

Figure 6
Monte Carlo versus Bootstrap for the Model 1 Skeleton

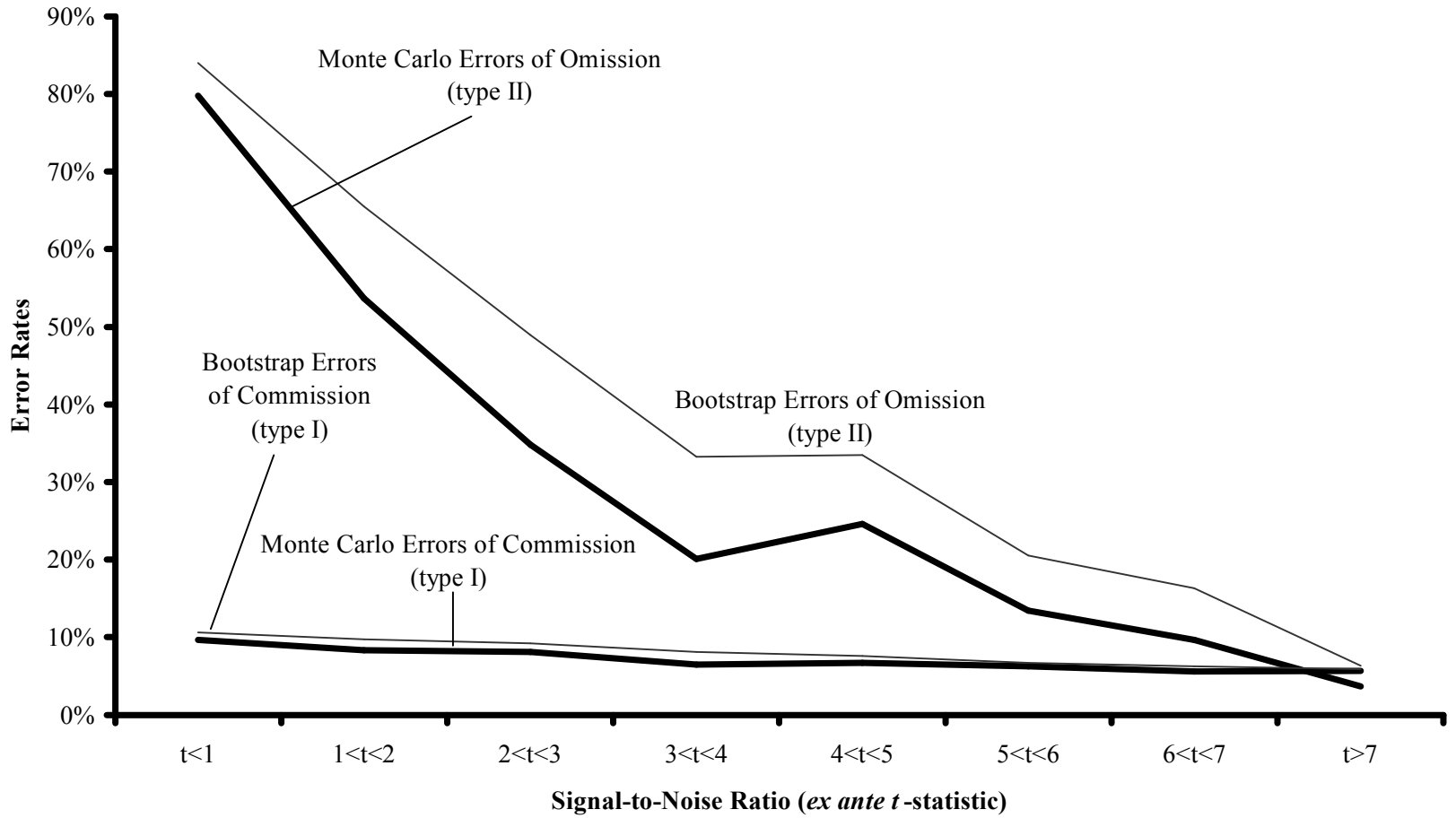


Figure 7
Monte Carlo versus Bootstrap for the Model 1 Unshielded Colliders

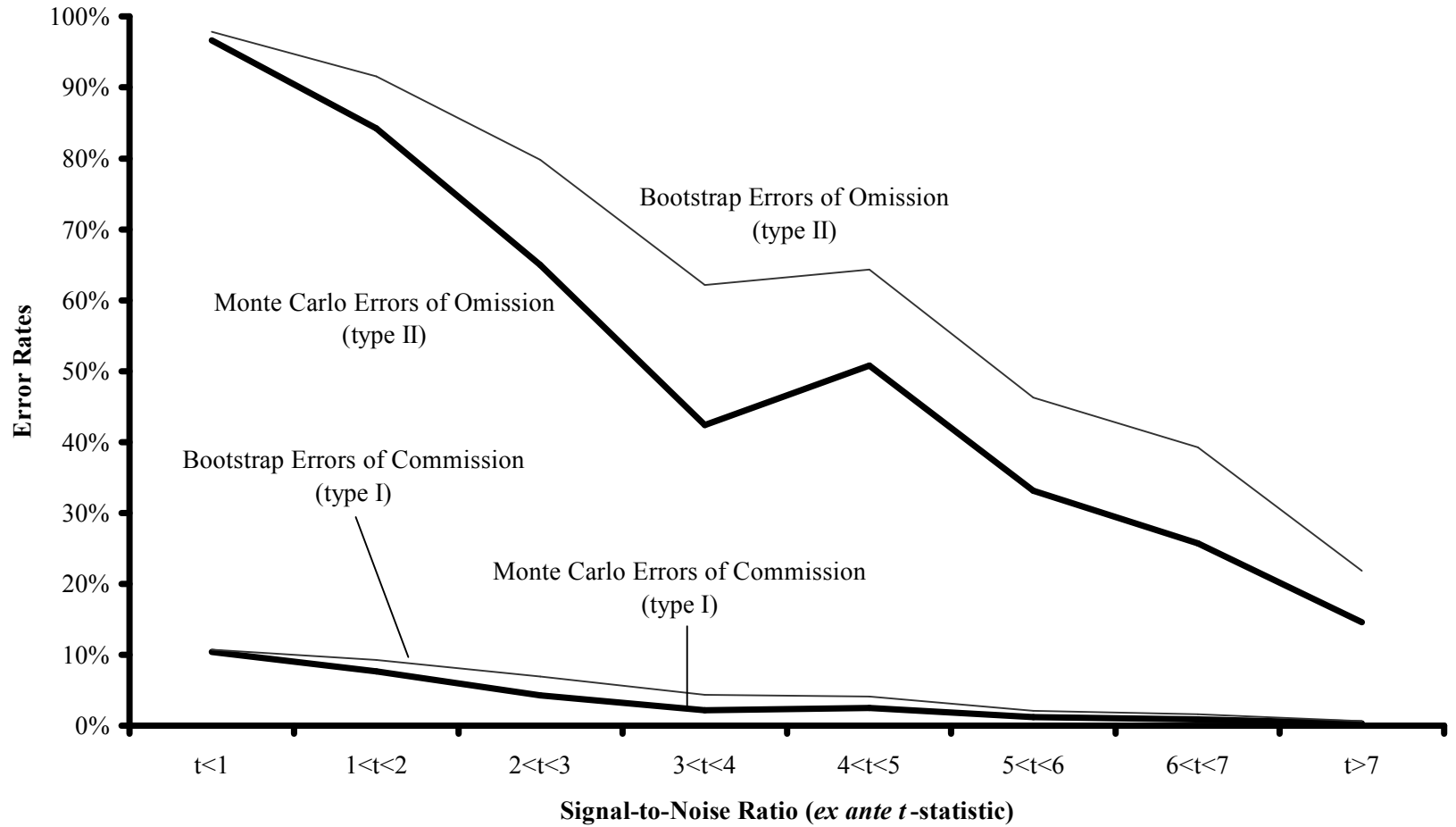


Table 2
How Well Does the Bootstrap Mimic the Monte Carlo Simulations?

Difference between Bootstrap and Monte Carlo Simulations (percentage points)										
Signal-to-Noise Ratios (measured by <i>ex ante</i> <i>t</i> -statistics)										
Model	Errors of:	$t < 1$	$1 < t < 2$	$2 < t < 3$	$3 < t < 4$	$4 < t < 5$	$5 < t < 6$	$6 < t < 7$	$t > 7$	mean
Skeleton										
Model 1	Commission	0.9	1.4	1.1	1.6	0.9	0.5	0.7	0.2	0.9
Model 2	Commission	1.5	1.1	0.9	1.2	1.3	0.9	0.5	0.2	0.9
Model 2A	Commission	12.4	11.8	11.0	9.7	8.5	7.5	6.7	5.8	9.2
Model 3	Commission	1.1	1.1	1.0	1.0	0.6	0.7	0.5	0.6	0.8
Model 4	Commission	1.1	1.0	1.0	0.8	0.8	0.6	0.6	0.3	0.8
Model 1	Omission	4.3	11.8	14.1	13.1	8.8	7.1	6.6	2.7	8.6
Model 2	Omission	4.3	9.6	13.0	13.9	8.7	6.5	3.3	2.2	7.7
Model 2A	Omission	-7.6	-1.6	2.0	2.6	0.4	-0.1	0.2	0.6	-0.4
Model 3	Omission	7.3	11.7	14.0	10.1	6.8	6.1	5.0	3.0	8.0
Model 4	Omission	6.1	10.9	14.0	10.0	6.7	6.2	4.3	2.7	7.6
Unshielded Colliders										
Model 1	Commission	0.3	1.7	2.6	2.1	1.6	0.9	0.7	0.3	1.3
Model 2	Commission	-0.1	0.2	-0.5	-1.5	-0.9	-0.4	1.2	2.6	0.1
Model 2A	Commission	-7.6	-1.6	2.0	2.6	0.4	-0.1	0.2	0.6	-0.4
Model 3	Commission	-0.3	-0.5	-0.7	-0.4	0.0	0.3	0.5	0.8	0.0
Model 4	Commission	-0.2	-0.6	-0.9	-0.4	-0.1	0.1	0.7	0.7	-0.1
Model 1	Omission	1.2	7.4	14.8	19.8	13.5	13.1	13.5	7.3	11.3
Model 2	Omission	1.8	5.5	11.9	17.6	16.1	12.5	7.9	8.1	10.2
Model 2A	Omission	-3.0	0.8	6.8	10.0	7.5	7.3	9.2	10.4	6.1
Model 3	Omission	2.7	9.0	15.2	13.5	10.8	11.8	8.9	8.1	10.0
Model 4	Omission	1.2	6.7	15.5	14.3	11.0	11.3	9.4	4.9	9.3

End

of

Lecture 2