Causal Reasoning with Probabilistic Graphical Models

Jose M. Peña
Department of Computer and Information Science
Linköping University, Sweden

Chalmers CSE, January 26, 2017

Motivating Example: Simpson's Paradox

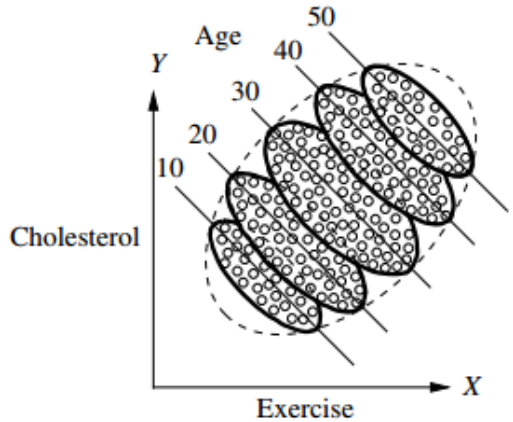


Figure 1.1: Results of the exercise-cholesterol study, segregated by age

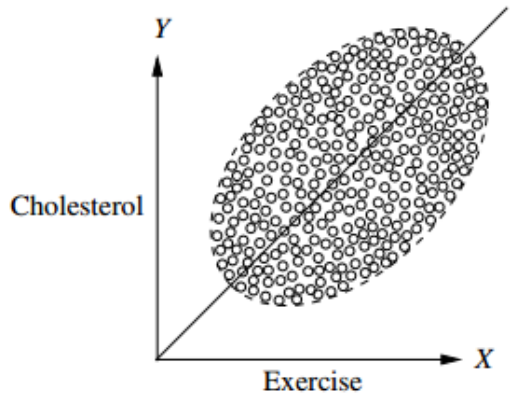


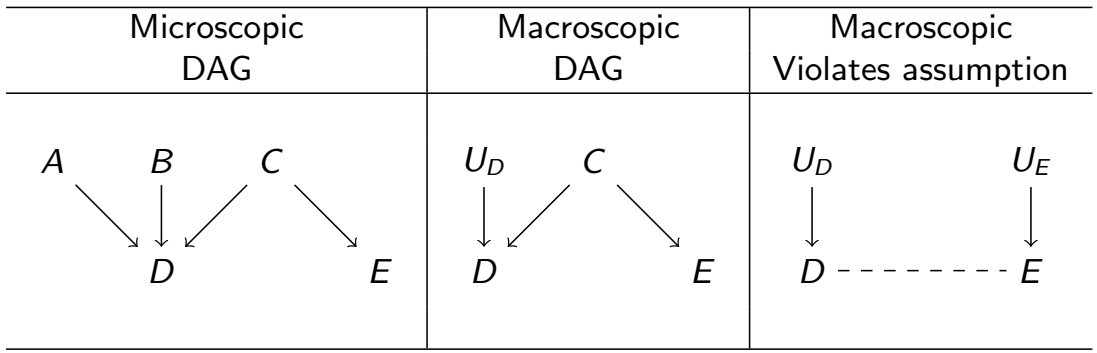
Figure 1.2: Results of the exercise-cholesterol study, unsegregated. The data points are identical to those of Figure 1.1, except the boundaries between the various age groups are not shown

- $p(Cholesterol|\mathbf{do}(Exercise)) = \sum_{age} p(Cholesterol|Exercise, age)p(age)$ since
  - we have a causal model: $\{E \to C, E \leftarrow A \to C\}$, and
  - we have a calculus: *do*-calculus.

Outline

Causal Models: Qualitative

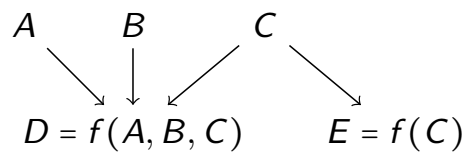
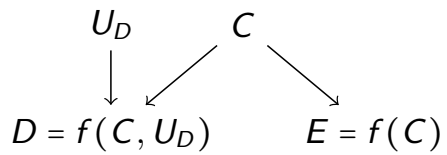
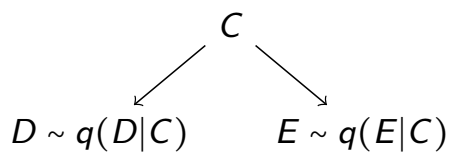
- At the microscopic level, every system is a causal system, **assuming** Reichenbach's principle of common cause: "No correlation without causation".
- Then, at the microscopic level, every system can be represented by a causal model.
- The structure of the causal model can be represented as a **directed and acyclic graph (DAG)**.



- At the macroscopic level, every system can also be represented by a causal model **if no unmodeled variable** is cause of two or more modeled variables. The unmodeled causes of a variable X are aggregated into an error variable U_X.

Causal Models: Quantitative

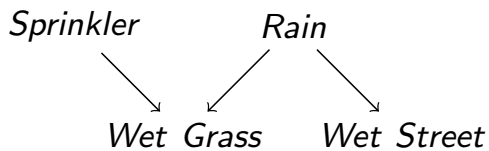
- **Assuming** Laplace's demon, every variable is a deterministic function of its causes at the microscopic level.

Microscopic DAG Functional	Macroscopic DAG Functional	Macroscopic DAG Probabilistic
		

- Then, every variable is a probabilistic function of its modeled causes at the macroscopic level.
- Both Reichenbach's principle of common cause and Laplace's demon have been disproven. Human reasoning seem to comply with both of them, though.
- Then, probabilistic DAGs (a.k.a **Bayesian networks**) may not be ontological but epistemological models.

Bayesian Networks: Definition

- A **Bayesian network (BN)** over a finite set of random variables $V = \{V_1, \ldots, V_n\}$ consists of
  - a DAG G whose nodes are the elements in V, and
  - parameter values θ_G specifying conditional probability distributions $q(V_i|pa_G(V_i))$.

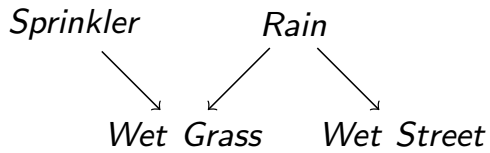
DAG	Parameter values for the conditional probability distributions
	$q(S) = (0.3, 0.7)$ $q(R) = (0.5, 0.5)$ $q(WG\|r_0, s_0) = (0.1, 0.9)$ $q(WG\|r_0, s_1) = (0.7, 0.3)$ $q(WG\|r_1, s_0) = (0.8, 0.2)$ $q(WG\|r_1, s_1) = (0.9, 0.1)$ $q(WS\|r_0) = (0.1, 0.9)$ $q(WS\|r_1) = (0.7, 0.3)$ $p(S, R, WG, WS) = q(S)q(R)q(WG\|S, R)q(WS\|R)$

- The BN represents a causal model of the system.
- The BN also represents a probabilistic model of the system, namely

$$p(V) = \prod_i q(V_i|pa_G(V_i)).$$

Bayesian Networks: Causal Reasoning

- What would be the state of the system if a random variable X is **forced** to take the state x, i.e. $p(V \setminus X | do(x))$?

Original	After $do(r_1)$
	
$q(S) = (0.3, 0.7)$ $q(R) = (0.5, 0.5)$ $q(WG \mid r_0, s_0) = (0.1, 0.9)$ $q(WG \mid r_0, s_1) = (0.7, 0.3)$ $q(WG \mid r_1, s_0) = (0.8, 0.2)$ $q(WG \mid r_1, s_1) = (0.9, 0.1)$ $q(WS \mid r_0) = (0.1, 0.9)$ $q(WS \mid r_1) = (0.7, 0.3)$ $p(S, R, WG, WS) = q(S)q(R)q(WG \mid S, R)q(WS \mid R)$	$q(S) = (0.3, 0.7)$ $q(WG \mid s_0) = (0.8, 0.2)$ $q(WG \mid s_1) = (0.9, 0.1)$ $q(WS) = (0.7, 0.3)$ $p(S, WG, WS) = q(S)q(WG \mid S)q(WS)$

- In words:
  - Remove X and all the edges from and to X from G.
  - Remove $q(X | pa_G(X))$.
  - Replace $q(V_i | pa_G(V_i))$ with $q(V_i | pa_G(V_i) \setminus X, x)$
  - Set $p(V \setminus X | do(x)) = \prod_i q(V_i | pa_G(V_i))$.

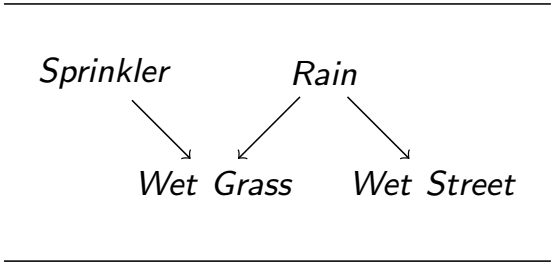
Bayesian Networks: Probabilistic Reasoning

- What is the state of a random variable Y if a random variable X is **observed** to be in the state x, i.e. $p(Y|x)$?
  - $p(Y|x) = \frac{p(Y,x)}{p(x)} = \frac{\sum_{V \setminus X \setminus Y} p(V \setminus X \setminus Y, Y, x)}{\sum_{V \setminus X} p(V \setminus X, x)}$
  - In the previous example,

$$
\begin{aligned}
p(WS|s) = \frac{p(WS,s)}{p(s)} &= \frac{\sum_{R,WG} p(s,R,WG,WS)}{\sum_{R,WG,WS} p(s,R,WG,WS)} \\
&= \frac{\sum_{R,WG} q(s)q(R)q(WG|s,R)q(WS|R)}{\sum_{R,WG,WS} q(s)q(R)q(WG|s,R)q(WS|R)} \\
&= \frac{q(s)\sum_R [q(R)q(WS|R)\sum_{WG} q(WG|s,R)]}{q(s)\sum_R [q(R)\sum_{WG}[q(WG|s,R)[\sum_{WS} q(WS|R)]]]}
\end{aligned}
$$

- Answering the question above is NP-hard.
- A BN allows to **efficiently** compute a posterior probability distribution from a prior probability distribution in the light of observations, hence the name.
- It is efficient because it takes advantage of the independences represented in the BN, e.g. via the junction tree or Lauritzen-Spiegelhalter algorithm.

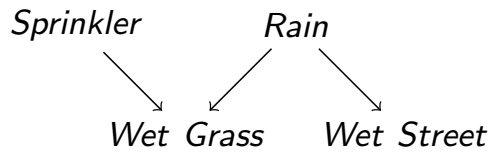
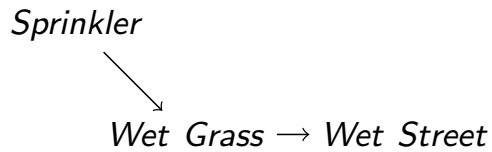
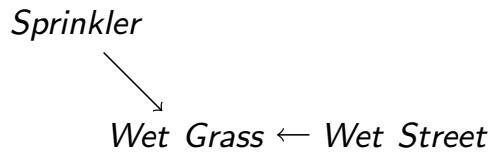
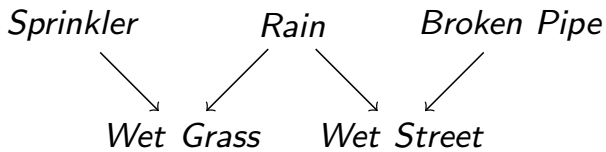
Bayesian Networks: Separation



- A BN (G, θ_G) represents an independence model via missing edges in G.
- A node B in a route ρ in G is called a **collider** node in ρ if $A \to B \leftarrow C$ is a subroute of ρ (note that maybe $A = C$).
- Given three disjoint subsets of nodes X, Y and Z, we say that X is **separated** from Y given Z in G (i.e. $X \perp_G Y|Z$) when every route ρ between a node in X and a node in Y is such that
  - some collider node in ρ is not in Z, or
  - some non-collider node in ρ is in Z.
- For instance, $S \perp_G \{R, WS\}|\varnothing$, $S \perp_G WS|\{WG, R\}|\varnothing$, $\{S, WG\} \perp_G WS|R$.
- The separation criterion is **sound**, i.e. $X \perp_G Y|Z \Rightarrow X \perp_p Y|Z$.
- The separation criterion is also **complete**, i.e. maybe $I(p) = \{X \perp_p Y|Z\} = \{X \perp_G Y|Z\} = I(G)$.
- Such so-called **faithful** probability distributions exist.

Bayesian Networks: Shortcomings

- Assume that the random variable *Rain* is not modeled, e.g. it is not observable.

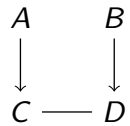
True causal model	BN	Causal model	Probabilistic model
		Wrong	Wrong
		Wrong	Right
	Any	Wrong	Wrong

- Solution ? Chain graphs.

MVR CG	Parameter values for the conditional probability distributions
	$q(S) = (0.3, 0.7)$ $q(WG, WS\|S) = \sum_R q(R)q(WG\|R, S)q(WS\|R)$ $p(S, WG, WS) = q(S)q(WG, WS\|S)$

Chain Graphs: Definition

LWF CG	Parameter values for the conditional probability distributions



$q(A) \equiv \psi_1^1(A)$

	$A=0$	$A=1$
	0.2	0.8

$q(B) \equiv \psi_2^1(B)$

	$B=0$	$B=1$
	0.3	0.7

$\psi_3^1(A,B)$

	$B=0$	$B=1$
$A=0$	1	2
$A=1$	4	3

$\psi_3^2(C,A)$

	$A=0$	$A=1$
$C=0$	5	6
$C=1$	8	7

$\psi_3^3(B,D)$

	$D=0$	$D=1$
$B=0$	9	10
$B=1$	11	12

$\psi_3^4(C,D)$

	$D=0$	$D=1$
$C=0$	13.3	14.4
$C=1$	15.5	16.6

$$p(A,B,C,D) = q(A)q(B)q(C,D|A,B) = \psi_1^1(A)\psi_2^1(B)\psi_3^1(A,B)\psi_3^2(C,A)\psi_3^3(B,D)\psi_3^4(C,D)$$

- A semi-directed cycle is a route of the form $V_1 \to V_2 \multimap \ldots \multimap V_n = V_1$.
- A connectivity component is a maximal connected set with only undirected/bidirected edges.
- A **chain graph (CG)** over a finite set of random variables $V = \{V_1, \ldots, V_n\}$ consists of
  - a graph G with possibly directed and **undirected/bidirected** edges whose nodes are the elements in V such that G has no semi-directed cycle, and
  - parameter values θ_G specifying conditional probability distributions $q(K_i|pa_G(K_i)) = \prod_j \psi_i^j$ where K_i is a connectivity component.
- The CG represents a causal model of the system.
- The CG also represents a probabilistic model of the system, namely

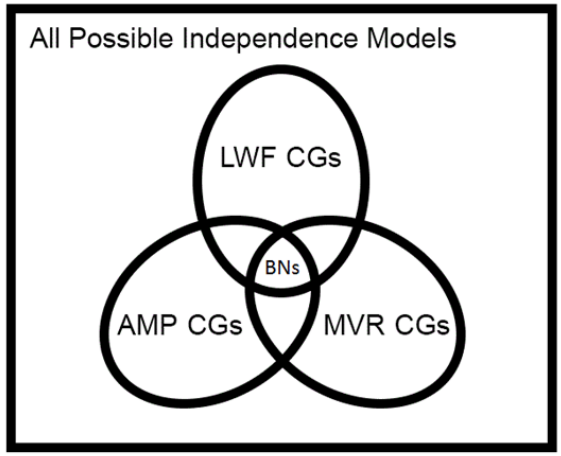
$$p(V) = \prod_i q(K_i|pa_G(K_i)) = \prod_i \prod_j \psi_i^j.$$

Chain Graphs: Separation

- A CG (G, θ_G) represents an independence model via missing edges in G.
- The separation criterion for CGs is the same as for BNs. Only the definition of collider node B differs:
  - BNs: $A \to B \leftarrow C$.
  - **Andersson-Madigan-Perlman (AMP) CGs**: $A \to B \circ\!\!- C$ or $A \circ\!\!- B \leftarrow C$.
  - **Multivariate regression (MVR) CGs**: $A \circ\!\!\to B \leftarrow\!\!\circ C$.
  - **Lauritzen-Wermuth-Frydenberg (LWF) CGs**: $A \to B_1 - \ldots - B_m \leftarrow C$.
- The separation criteria are sound and complete.

Chain Graphs: Why Different Interpretations ?

- None of them is a subset of another.



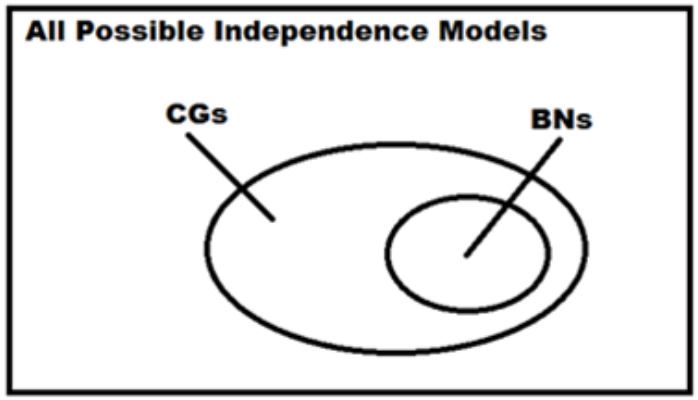
- Their pairwise intersections have been characterized.

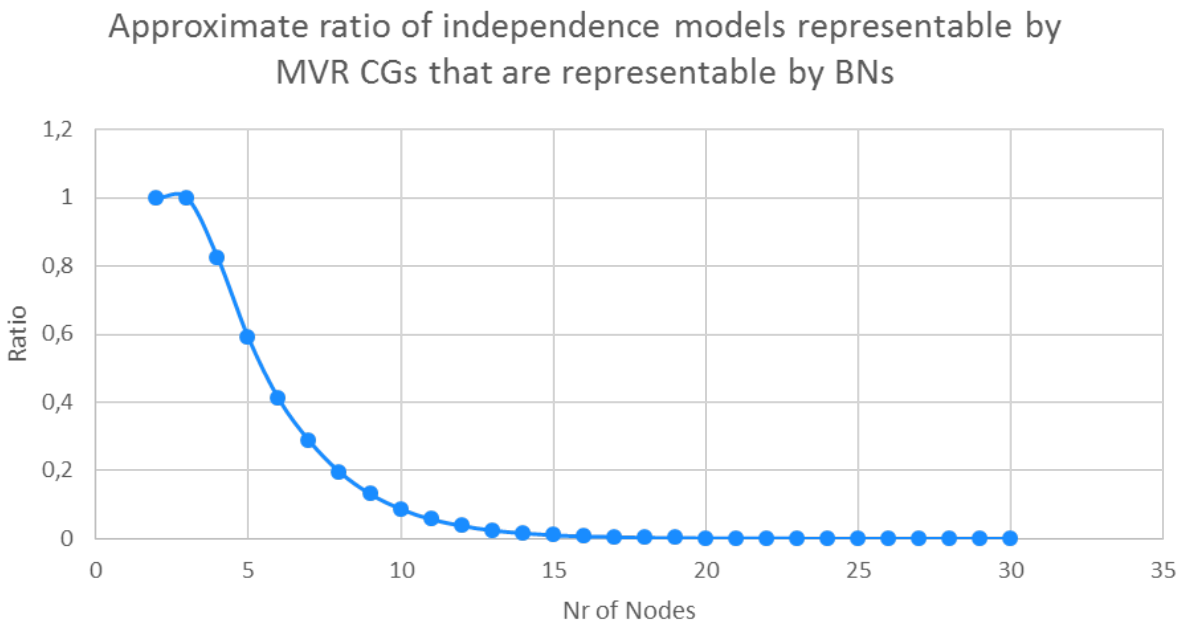
	LWF	AMP	MVR
LWF	-	Unidentified	$(G_{cl(K)})^M$ is chordal for all $K \in cc(G)$
AMP	G contains no k-biflag where $k \geq 2$ (Andersson et al. 2001)	-	G' does not contain any induced subgraph of the form $X - Y - Z$
MVR	G' contains no bidirected edge	G' contains no bidirected flag	

Given a CG G of the interpretation in the row, and a maximally oriented CG G' that is equivalent to G (i.e. $I(G) = I(G')$), there exists a CG H of the interpretation in the column such that G and H are equivalent if and only if the condition in the intersecting cell is fulfilled.

Chain Graphs: Why ?

- Any CG interpretation is much more expressive than BNs.





- The curves for LWF and AMP CGs are similar.

Chain Graphs: Learning under Faithfulness

Input: A probability distribution p that is **faithful** to an unknown AMP CG G.
Output: An AMP CG H st $I(H) = I(G)$.

1 Let H denote the complete undirected graph
2 Set $l = 0$
3 Repeat while $l \leq |V| - 2$
4 For each ordered pair of nodes A and B in H st $A \in ad_H(B)$ and $|[ad_H(A) \cup ad_H(ad_H(A))] \setminus B| \geq l$
5 If there is some $S \subseteq [ad_H(A) \cup ad_H(ad_H(A))] \setminus B$ st $|S| = l$ and $A \perp_p B | S$ then
6 Set $S_{AB} = S_{BA} = S$
7 Remove the edge $A - B$ from H
8 Set $l = l + 1$
9 Apply the rules R1-R4 to H while possible
10 Replace every edge $\rightarrowtail$ ($\rightarrowtail$) in H with $\rightarrow$ ($-$)

R1: $A \circ\!\!-\!\!\circ B \circ\!\!-\!\!\circ C$ $\Rightarrow$ $A \rightarrowtail B \circ\!\!-\!\!\circ C$, $\wedge\ B \notin S_{AC}$

R2: $A \rightarrowtail B \circ\!\!-\!\!\circ C$ $\Rightarrow$ $A \rightarrowtail B \rightarrowtail C$, $\wedge\ B \in S_{AC}$

R3: $A \rightarrowtail \cdots \rightarrowtail B$ $\Rightarrow$ $A \rightarrowtail \cdots \rightarrowtail B$

R4: A, B, C, D $\Rightarrow$ A, B, C, D, $\wedge\ A \in S_{CD}$

- Similar algorithms exist for LWF and MVR CGs.

Chain Graphs: Learning under Composition

- A probability distribution p satisfies the **composition property** assumption if
$$X \perp_p Y|Z \wedge X \perp_p W|Z \Rightarrow X \perp_p Y \cup W|Z.$$
- Not all the probability distributions satisfy the composition property, e.g. XOR. However, the property is satisfied by, among others,
  - all the regular Gaussian probability distributions, and
  - all the probability distributions resulting from marginalization and conditioning in a probability distribution that is faithful to some CG.
- Then, the composition property assumption is **milder** than the faithfulness assumption.
- A CG G **includes** a probability distribution p if $I(G) \subseteq I(p)$.
- Moreover, G is **inclusion optimal** with respect to p if
  - $I(G) \subseteq I(p)$, and
  - there is no CG F such that $I(G) \subset I(F) \subseteq I(p)$.

Chain Graphs: Learning under Composition

- We have proven that for any probability distribution p that satisfies the composition property, the algorithm below finds a LWF CG that is inclusion optimal with respect to p.

---

1 G :=Empty graph
2 Repeat until **all** the CGs that are equivalent to G have been considered
3 For every ordered pair of nodes X and Y
4 If $X \to Y$ is in G but $X \perp_p Y | Bd_G(Y) \setminus X$ then **remove** $X \to Y$ from G and go to line 2
5 If $X - Y$ is in G but $X \perp_p Y | Bd_G(Y) \setminus X$ and $X \perp_p Y | Bd_G(X) \setminus Y$ then **remove** $X - Y$ from G and go to line 2
6 If $X \to Y$ is not in G but adding $X \to Y$ to G results in a CG and $X \not\perp_p Y | Bd_G(Y)$ then **add** $X \to Y$ to G and go to line 2
7 If $X - Y$ is not in G but adding $X - Y$ to G results in a CG and $X \not\perp_p Y | Bd_G(Y)$ or $X \not\perp_p Y | Bd_G(X)$ then **add** $X - Y$ to G and go to line 2
8 Move to another CG that is equivalent to G by performing a random number of random **feasible mergings or splits** on G and thereby updating G
9 Return G

---

where $Bd_G(Y) = Pa_G(Y) \cup Ne_G(Y)$.

- **No** similar algorithm exists for AMP or MVR CGs.

Chain Graphs: Learning under no Assumption

- Learning MVR CGs via answer set programming (ASP). Similar algorithms exist for LWF and AMP CGs.

```
% input predicates
% nodes(N): N is the number of nodes
% set(X): X is the index of a set of nodes
% dep(X,Y,C,I,W) (resp. indep(X,Y,C,I,W)): the nodes X and Y are dependent (resp.
%                                          independent) given the set of nodes C
%                                          after having intervened on the node I

% nodes
node(X) :- nodes(N), X=1..N.                                          % rule 1

% edges
{ line(X,Y,0) } :- node(X), node(Y), X != Y.                          %      2
{ arrow(X,Y,0) } :- node(X), node(Y), X != Y.                         %      3
line(X,Y,I) :- line(X,Y,0), node(I), X != I, Y != I, I > 0.           %      4
line(X,Y,I) :- line(X,I,0), line(I,Y,0), node(I), X != Y, I > 0.
arrow(X,Y,I) :- arrow(X,Y,0), node(I), Y != I, I > 0.                 %      6
line(X,Y,I) :- line(Y,X,I).                                           %      7
:- arrow(X,Y,I), arrow(Y,X,I).                                        %      8

% directed acyclity
ancestor(X,Y) :- arrow(X,Y,0).                                        %      9
ancestor(X,Y) :- ancestor(X,Z), ancestor(Z,Y).
:- ancestor(X,Y), arrow(Y,X,0).                                       %      11

% set membership
inside_set(X,C) :- node(X), set(C), 2**(X-1) & C != 0.                %      12
outside_set(X,C) :- node(X), set(C), 2**(X-1) & C = 0.                %      13

% end_line/head/tail(X,Y,C,I) means that there is a connecting route
% from X to Y given C that ends with a line/arrowhead/arrowtail

% single edge route
end_line(X,Y,C,I) :- line(X,Y,I), outside_set(X,C).                   %      14
end_head(X,Y,C,I) :- arrow(X,Y,I), outside_set(X,C).
end_tail(X,Y,C,I) :- arrow(Y,X,I), outside_set(X,C).

% connection through non-collider
end_line(X,Y,C,I) :- end_line(X,Z,C,I), line(Z,Y,I), outside_set(Z,C).
end_line(X,Y,C,I) :- end_tail(X,Z,C,I), line(Z,Y,I), outside_set(Z,C).
end_head(X,Y,C,I) :- end_line(X,Z,C,I), arrow(Z,Y,I), outside_set(Z,C).
end_head(X,Y,C,I) :- end_head(X,Z,C,I), arrow(Z,Y,I), outside_set(Z,C).
end_head(X,Y,C,I) :- end_tail(X,Z,C,I), arrow(Z,Y,I), outside_set(Z,C).
end_tail(X,Y,C,I) :- end_tail(X,Z,C,I), arrow(Y,Z,I), outside_set(Z,C).

% connection through collider
end_line(X,Y,C,I) :- end_head(X,Z,C,I), line(Z,Y,I), inside_set(Z,C).
end_tail(X,Y,C,I) :- end_line(X,Z,C,I), arrow(Y,Z,I), inside_set(Z,C).
end_tail(X,Y,C,I) :- end_head(X,Z,C,I), arrow(Y,Z,I), inside_set(Z,C).    %      25

% derived non-separations
con(X,Y,C,I) :- end_line(X,Y,C,I), X != Y, outside_set(Y,C).          %      26
con(X,Y,C,I) :- end_head(X,Y,C,I), X != Y, outside_set(Y,C).
con(X,Y,C,I) :- end_tail(X,Y,C,I), X != Y, outside_set(Y,C).
con(X,Y,C,I) :- con(Y,X,C,I).                                         %      29

% satisfy all dependences
:- dep(X,Y,C,I,W), not con(X,Y,C,I).                                  %      30

% maximize the number of satisfied independences
:~ indep(X,Y,C,I,W), con(X,Y,C,I). [W,X,Y,C,I]                        %      31

% minimize the number of lines/arrows
:~ line(X,Y,0), X < Y. [1,X,Y,1]                                      %      32
:~ arrow(X,Y,0). [1,X,Y,2]                                            %      33

% show results
#show. #show line(X,Y) : line(X,Y,0), X < Y. #show arrow(X,Y) : arrow(X,Y,0).
```

```
nodes(3).   % three nodes
set(0..7).  % all subsets of three nodes

% observations
dep(1,2,0,0,1).
dep(1,2,4,0,1).
dep(2,3,0,0,1).
dep(2,3,1,0,1).
dep(1,3,0,0,1).
dep(1,3,2,0,1).

% interventions on the node 3
dep(1,2,4,0,3,1).
indep(2,3,0,3,1).
indep(2,3,1,3,1).
indep(1,3,0,3,1).
indep(1,3,2,3,1).
```

Acyclic Directed Mixed Graphs: Causal Reasoning

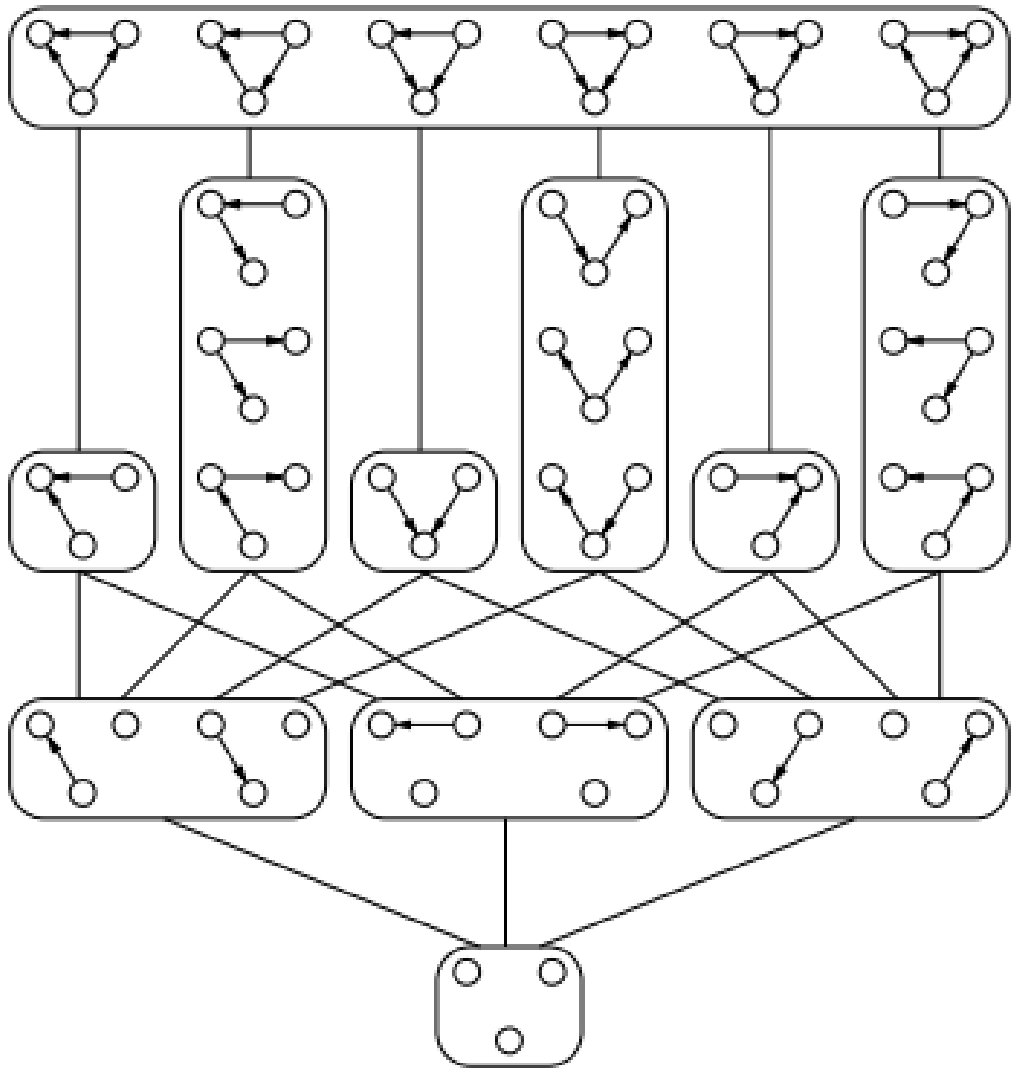
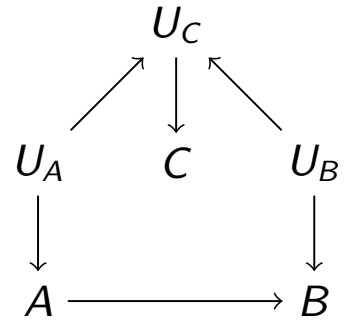
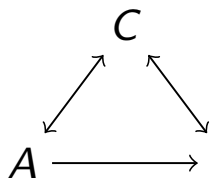
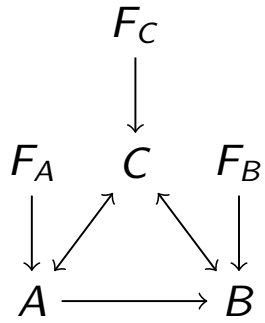


Figure 2: Hasse diagram of the space of Markov equivalence classes of Bayesian network structures over three variables.

Acyclic Directed Mixed Graphs: Causal Reasoning

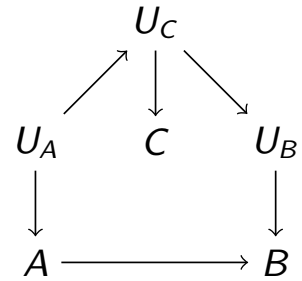
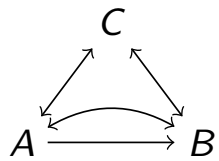
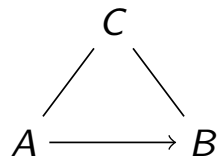
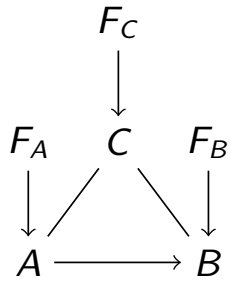
DAG D	Original ADMG G	Original ADMG G'
		

- $X \leftrightarrow Y$ in G iff X and Y have a **confounder** iff $U_X \not\perp_D U_Y | \varnothing$.
- Effect **identification** in G: $p(B|do(A)) = p(B|A)$.
- Identification is possible due to Pearl's *do*-calculus on G':
  - Rule 1 (insertion/deletion of observations):
    $p(Y|do(X), Z \cup W) = p(Y|do(X), W)$ if $Y \perp_{G'} Z | X \cup W || X$.
  - Rule 2 (intervention/observation exchange):
    $p(Y|do(X), do(Z), W) = p(Y|do(X), Z \cup W)$ if $Y \perp_{G'} F_Z | X \cup W \cup Z || X$.
  - Rule 3 (insertion/deletion of interventions):
    $p(Y|do(X), do(Z), W) = p(Y|do(X), W)$ if $Y \perp_{G'} F_Z | X \cup W || X$.

  Where $\cdot \perp_{G'} \cdot | \cdot || X$ denotes separation in G' after intervention on X:
  - Delete all the arrows into X.
  - Apply the separation criterion for MVR CGs.

Acyclic Directed Mixed Graphs: Causal Reasoning

- Assume **continuous** random variables and **additive** noise.

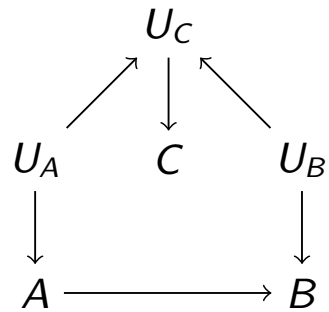
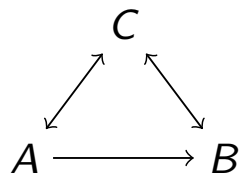
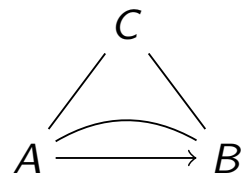
DAG D	Original ADMG G	**Alternative** ADMG H	**Alternative** ADMG H'
			

- $X - Y$ in H iff $U_X \not\perp_D U_Y | U \setminus \{U_X, U_Y\}$.
- Effect identification in G: Impossible.
- Effect identification in H: $p(B|do(A)) = \int p(B|A,C)p(C)\,dC$.
- Identification is possible due to *do*-calculus on H', but we need to **redefine** $\cdot \perp_{H'} \cdot \,|\, \cdot \,||X$:
  - Intervene on X.

1	Delete from H' all the edges $A \to B$ with $B \in X$
2	For each path $A - V_1 - \ldots - V_n - B$ in H' with $A, B \notin X$ and $V_1, \ldots, V_n \in X$
3	Add the edge $A - B$ to H'
4	Delete from H' all the edges $A - B$ with $B \in X$

  - Apply the separation criterion for AMP CGs.

Acyclic Directed Mixed Graphs: Causal Reasoning

DAG D	Original ADMG G	Alternative ADMG H
		

- Effect identification in G: $p(B|do(A)) = p(B|A)$.
- Effect identification in H: Impossible.
- Thus, original and alternative ADMGs are complementary.
- Is *do*-calculus **complete**, i.e. identifiable iff computable by *do*-calculus ?
  - Yes for original ADMGs.
  - Unknown yet for alternative ADMGs.

Acyclic Directed Mixed Graphs: Causal Reasoning

Algorithm: $ID(X, Y)$
Input: Two disjoint sets $X, Y \subseteq V$.
Output: An expression to compute $p(y|do(x))$ from $p(v)$ or FAIL.

1 Assume that V is partitioned into components $S_1, \ldots, S_k$ in G
2 Let $Z = An_{G_{V \setminus X}}(Y)$
3 Assume that Z is partitioned into components $Z_1, \ldots, Z_l$ in G^Z
4 For each Z_j do
5 Assume without loss of generality that $Z_j \subseteq S_j$
6 Compute $q(s_j)$ by Theorem 2
7 Compute $q(z_j)$ from $q(s_j)$ by calling $Identify(Z_j, S_j, q(s_j))$
8 If the call returns FAIL then stop and return FAIL
9 Return $p(y|do(x)) = \sum_{z \setminus y} \prod_j q(z_j)$

Algorithm: $Identify(C, S, q(s))$
Input: Two sets $C \subseteq S \subseteq V$. Moreover, G_C and G_S are both assumed to be composed of one single component.
Output: An expression to compute $q(c)$ from $q(s)$ or FAIL.

1 Let $W = An_{G_S}(C)$
2 Let $A = S \setminus W \setminus De_{G_S}(C)$
3 Compute $q(w|a) = \sum_{s \setminus a \setminus w} q(s) / \sum_{s \setminus a} q(s)$
4 Assume that C is contained in the component T of G^W
5 Compute $q(t|a)$ from $q(w|a)$ by Theorem 8
6 If C is an ancestral set in G_T then
7 Let $q(c|a) = \sum_{t \setminus c} q(t|a)$ and $q(a) = \sum_{s \setminus a} q(s)$
8 Return $q(c) = \sum_a q(c|a)q(a)$
9 Else return FAIL

Acyclic Directed Mixed Graphs: Causal Reasoning

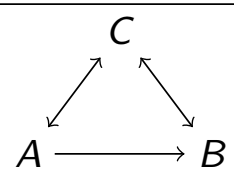
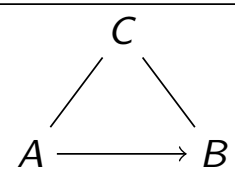
- Several sufficient (but **not** complete) graphical criteria for causal effect identification exist. For instance,

Theorem
A set of variables Z satisfies the back-door criterion relative to an ordered pair of variables (X, Y) in an ADMG G if

1. *Z contains no descendant of X in G, and*
2. *Z blocks all non-directed paths in G from X to Y.*

Moreover,

$$p(Y|do(X)) = \sum_z p(Y|X,z)p(z).$$

Original ADMG G	Alternative ADMG H
	

- Effect identification in G: $p(B|do(A)) = p(B|A)$.
- Effect identification in H: $p(B|do(A)) = \sum_c p(B|A,c)p(c)$.

Summary

- BNs: PGMs based on directed graphs + acyclic.
- CGs: PGMs based on mixed graphs + semi-directed acyclic.
- ADMGs: PGMs based on mixed graphs + directed acyclic.
- There exist three different interpretation of CGs (i.e. LWF, AMP and MVR CGs) and
  - none of them is a subset of another,
  - their pairwise intersections are characterized, and
  - each of them is much more expressive than BNs.
  - However, they are also more difficult to work with.
- There exist learning algorithms for
  - LWF, AMP and MVR CGs under the faithfulness assumption, and
  - LWF CGs under the milder composition property assumption, and
  - LWF, AMP and MVR CGs under no assumption but with poor scalability.
  - The last algorithm has been extended to ADMGs.
- Causal reasoning on CGs can be performed by applying *do*-calculus on
  - original ADMGs for MVR CGs, and
  - alternative ADMGs for AMP CGs.
- Topics not covered in this talk:
  - Cause of effects, counterfactuals, etc.
  - Causal reasoning for LWF CGs ?
  - Parameter learning for CGs and ADMGs: Iterative methods.
  - Probabilistic reasoning for AMP and MVR CGs.
  - Factorization according to CGs and ADMGs.

Selected Literature

- Bayesian networks:

  Koski, T. and Noble, J. A Review of Bayesian Networks and Structure Learning. Mathematica Applicanda, 2012.

- Chain graphs:

  Sonntag, D. Chain Graphs - Interpretations, Expressiveness and Learning Algorithms. PhD Thesis, 2016.

- Causal reasoning:

  Pearl, J. Reasoning with Cause and Effect. AI Magazine, 2002.

  Pearl, J., Glymour, M. and Jewell, N. P. Causal Inference in Statistics: A Primer. Wiley, 2016.

- Alternative acyclic directed mixed graphs:

  Peña, J. M. Alternative Markov and Causal Properties for Acyclic Directed Mixed Graphs. In UAI 2016.

- Software
  - GUI based: Hugin (see demo version), GeNIe.
  - R packages: bnlearn, gRain.

Thanks for your attention