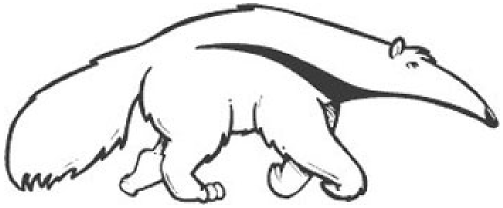
Algorithms for Causal Probabilistic Graphical Models

Class 5:

Causal Queries & Observational Data

Athens Summer School on AI
July 2024



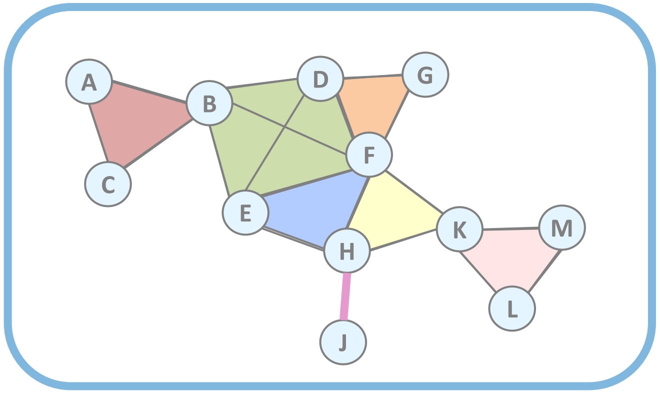
Prof. Rina Dechter
Prof. Alexander Ihler

BREN:ICS
INFORMATION AND COMPUTER SCIENCES

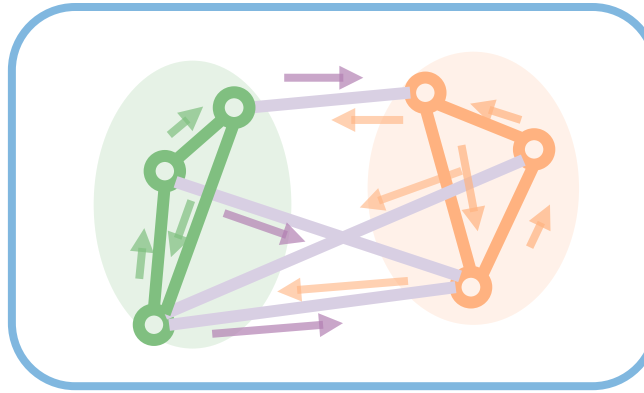
UNIVERSITY of CALIFORNIA IRVINE

Outline of Lectures

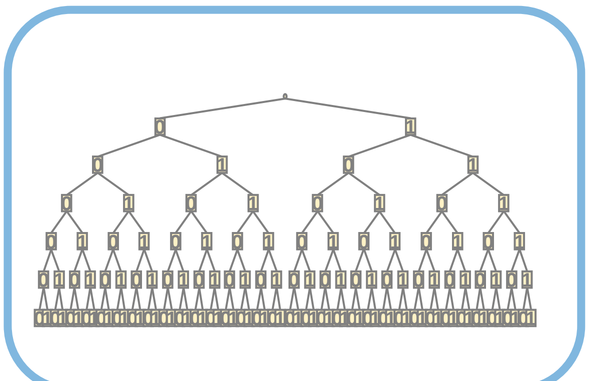
Class 1: Introduction & Inference



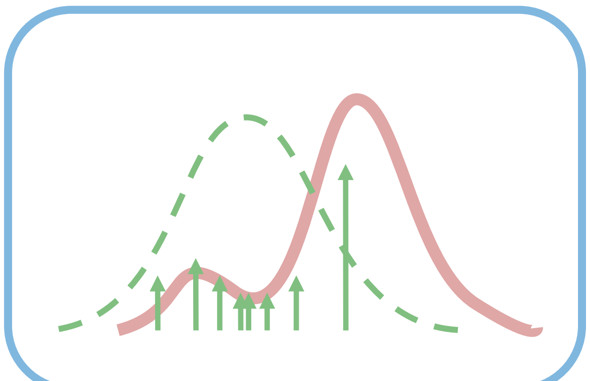
Class 2: Bounds & Variational Methods



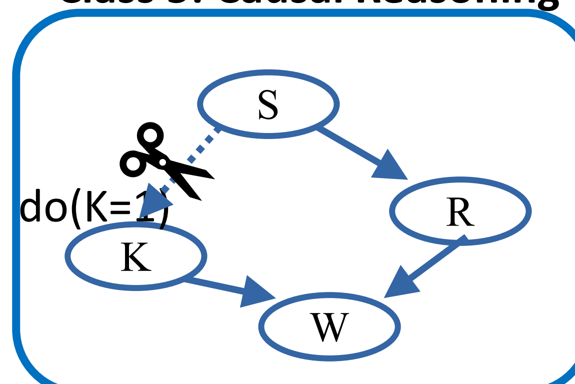
Class 3: Search Methods



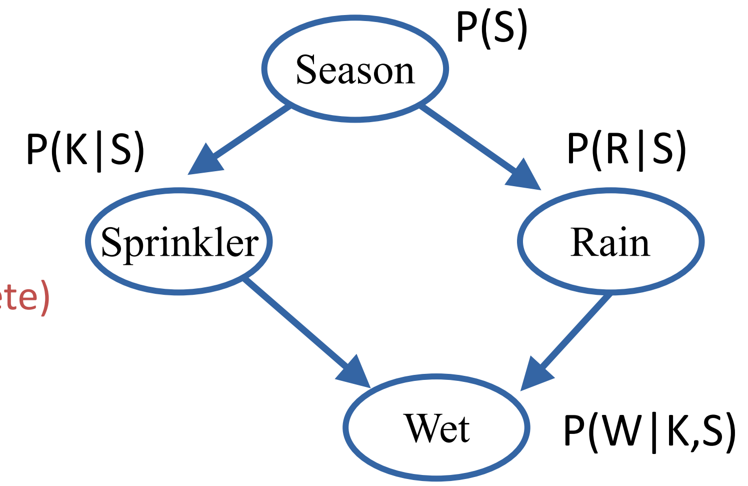
Class 4: Monte Carlo Methods



Class 5: Causal Reasoning



Graphical models



A *graphical model* consists of:

$X = \{X_1, \ldots, X_n\}$ -- variables

$D = \{D_1, \ldots, D_n\}$ -- domains (we'll assume discrete)

$F = \{f_{\alpha_1}, \ldots, f_{\alpha_m}\}$ -- functions or CPTs

and a *combination operator*

The *combination operator* defines an overall function from the individual factors,

e.g., "+" : $P(S,K,R,W) = P(S) \cdot P(K|S) \cdot P(R|S) \cdot P(W|K,S)$

Notation:

Discrete X_i values called "states"

"Tuple" or "configuration": states taken by a set of variables

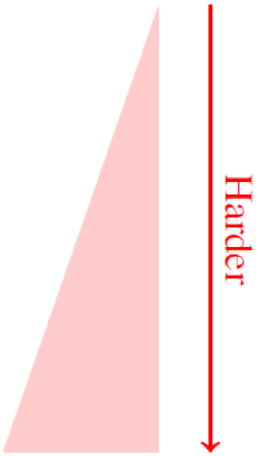
"Scope" of f: set of variables that are arguments to a factor f

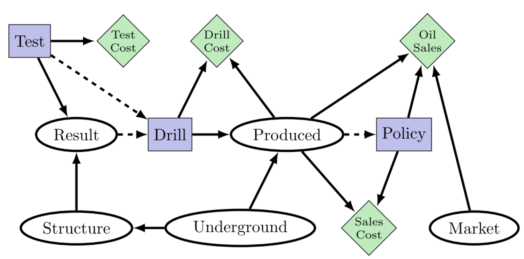
often index factors by their scope, e.g., $f_\alpha(X_\alpha), \quad X_\alpha \subseteq X$

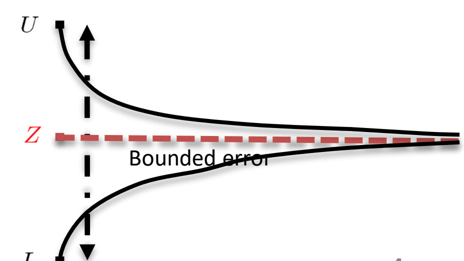
Probabilistic Reasoning Problems

- Exact inference time, space exponential in induced width
- Casual reasoning is a sum-inference task.

Max-Inference:	$f(x^*) = \max_x \prod_\alpha f_\alpha(x_\alpha)$
Sum-Inference: (e.g., causal effects)	$Z = \sum_x \prod_\alpha f_\alpha(x_\alpha)$
Mixed-Inference (MMAP):	$f_M(x_M^*) = \max_{x_M} \sum_{x_S} \prod_\alpha f_\alpha(x_\alpha)$
Mixed-Inference (MEU): (e.g., decisions, planning)	$\text{MEU} = \max_{D_1,\ldots,D_m} \sum_{X_1,\ldots X_n} (\prod_{P_i \in P} P_i) \times (\sum_{r_i \in R} r_i)$







Outline: Causal Inference

- Causal Models: Semantics
- **Causal Models: Queries**
- Identifiability
- Estimand Methods
- Learning Methods

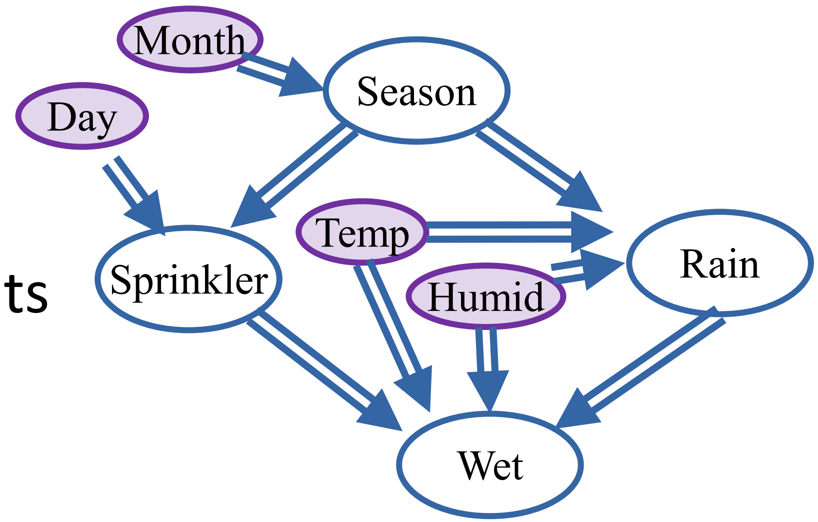
Pearl's Causal Hierarchy (PCH)

Level (Symbol)	Typical Activity	Typical Question	Examples
1 👀 Association $P(y \mid x)$	Seeing	What is? How would seeing X change my belief in Y?	What does a symptom tell us about the disease?
2 💪 Intervention $P(y \mid do(x), c)$	Doing	What if? What if I do X?	What if I take aspirin, will my headache be cured?
3 🧠 Counterfactual $P(y_x \mid x', y')$	Imagining, Retrospection	Why? What if I had acted differently?	Was it the aspirin that stopped my headache?

Structural Causal Models

- Endogenous (visible) variables V
  - Season, Sprinkler, Rain, Wet…
- Exogenous (latent) variables U
  - Temp, Humidity, Day, Month
- V are deterministic ($\Rightarrow$) given parents
  - $v_i = f_i(pa_i, u_i)$
- Randomness arises from U
  - $(u_1, \ldots, u_m) \sim p(U)$

Ex: Sprinkler



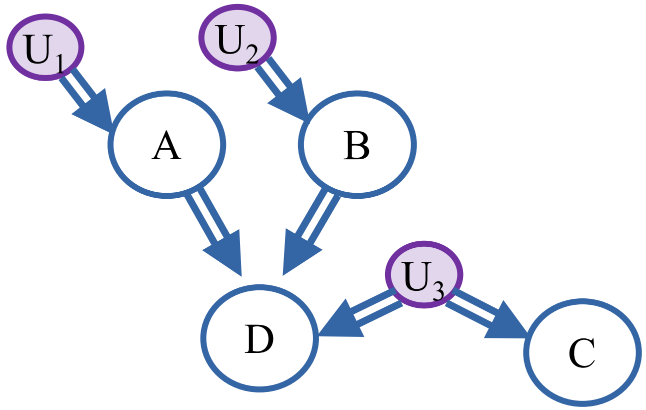
- We can only observe the variables V
  - SCM defines a **causal diagram**
    and the **observational distribution** p(V)

$$p(V) = \sum_{\mathbf{u}} p(\mathbf{u}) \prod_i p(V_i \,|\, pa_i, u_i)$$

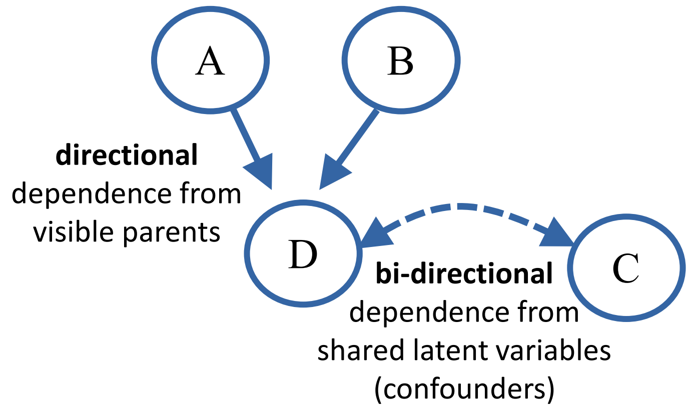
Causal Diagram

A graph over the visible variables V that describes their causal structure

Structural Causal Model



Causal Diagram



Special Cases

Markovian

- Each U_i has no parents, one child
  (equivalent to a Bayesian network)

Semi-Markovian

- Each U_i has no parents, ≤ 2 children

Observational Distribution

$$p(V) = \sum_{\mathbf{u}} p(\mathbf{u}) \prod_i p(V_i \,|\, pa_i, u_i)$$

visible and latent parents of V_i

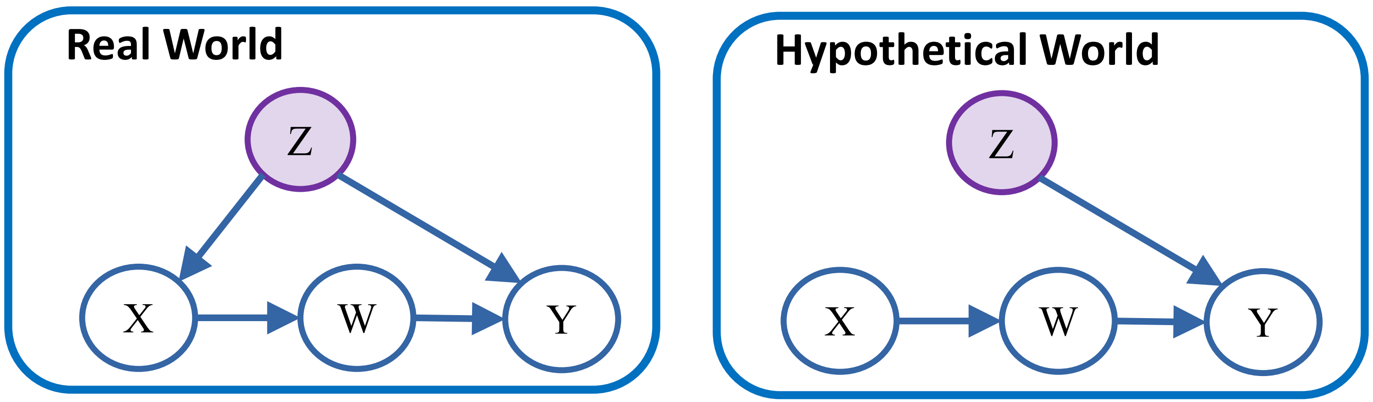
Outline: Causal Inference

- Causal Models: Semantics
- **Causal Models: Queries**
- Identifiability
- Estimand Methods
- Learning Methods

The Challenge of Causal Inference

"Causal Effect"

- How much does outcome Y change with X, if we vary X between two constants free of the influence of other (possibly unobserved) causes Z?



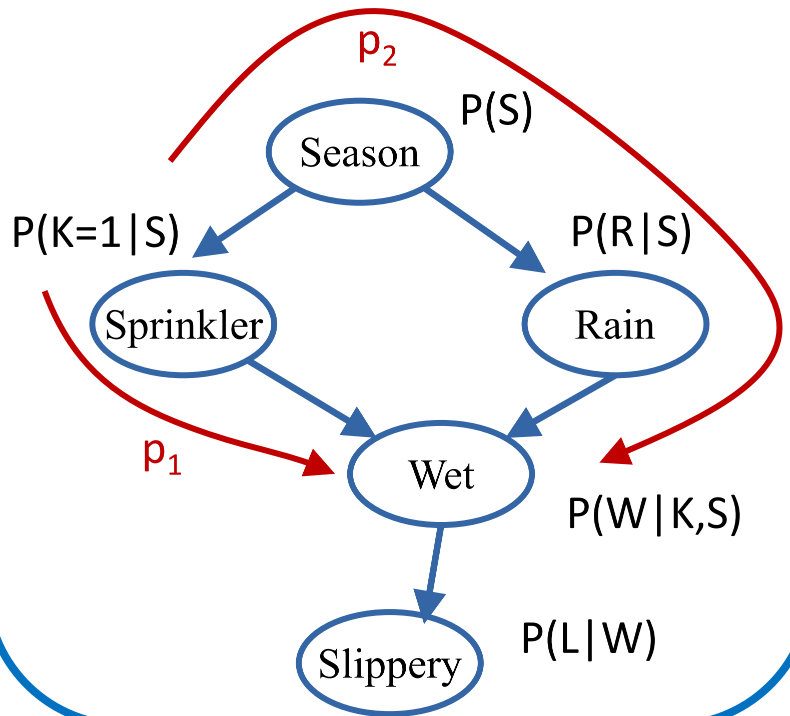
Z : age, sex
X : action
W : mediator
Y : outcome

- Randomized control experiments
  - Sample from hypothetical world directly
  - What if we cannot do this? (e.g., can't control X directly, or too much delay)
- Can we estimate using data only from the left model?

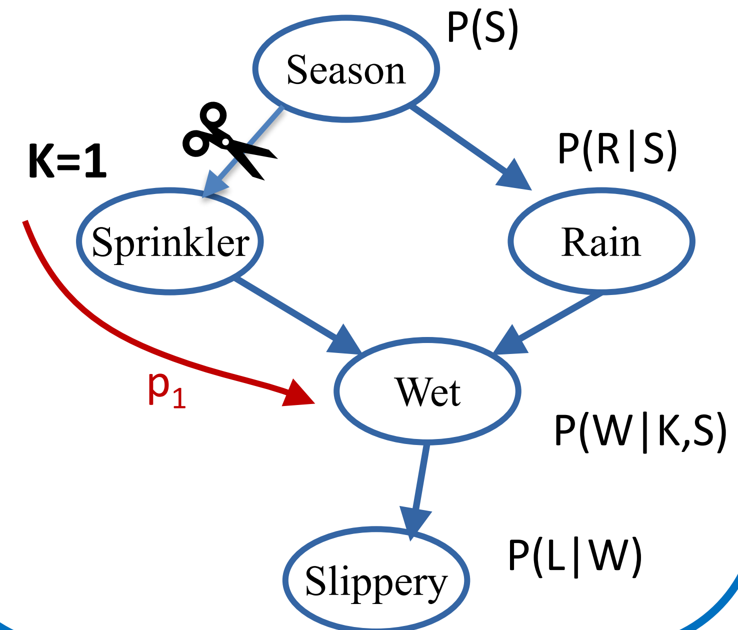
Computing Causal Effects

$$p(W|K=1) = ?$$

$$= \frac{\sum_{S,R} p(W|K=1,R)\, p(K=1|S)\, p(R|S)\, P(S)}{\sum_{S} p(K=1|S)\, p(S)}$$



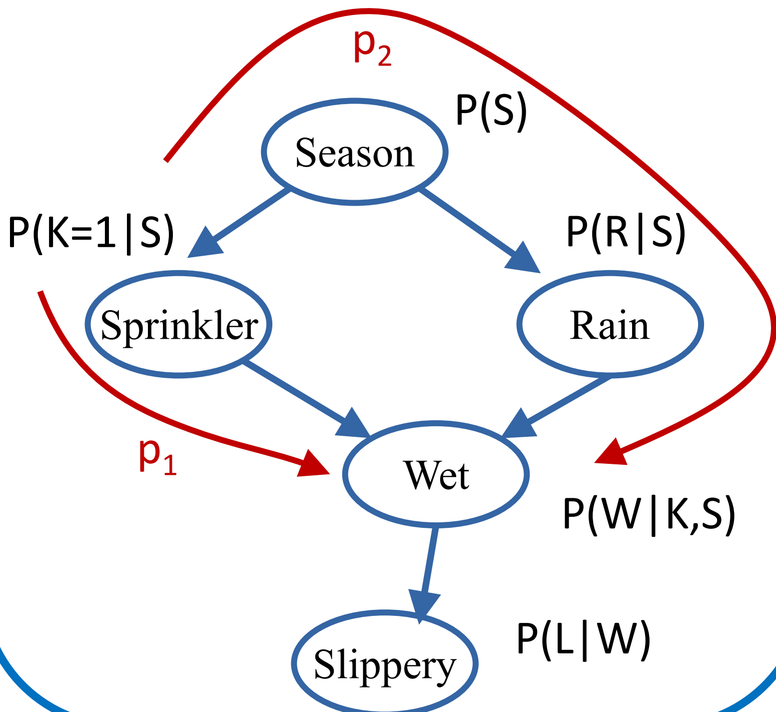
$$p(W|\mathrm{do}(K=1)) = ?$$



Computing Causal Effects

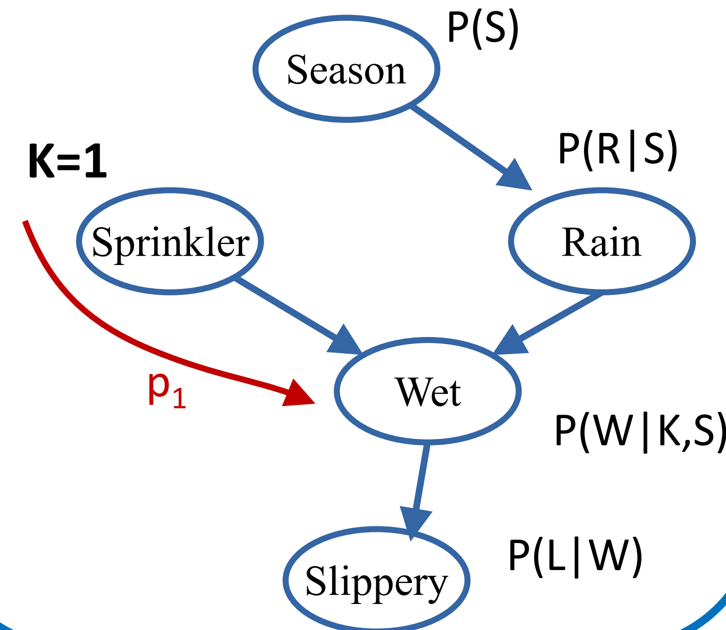
$$p(W|K=1) = ?$$

$$= \frac{\sum_{S,R} p(W|K=1,R)\, p(K=1|S)\, p(R|S)\, P(S)}{\sum_S p(K=1|S)\, p(S)}$$



$$p(W|\mathrm{do}(K=1)) = ?$$

$$= \frac{\sum_{S,R} p(W|K=1,R)\, \mathbb{1}(K=1)\, p(R|S)\, P(S)}{\mathbb{1}(K=1)}$$



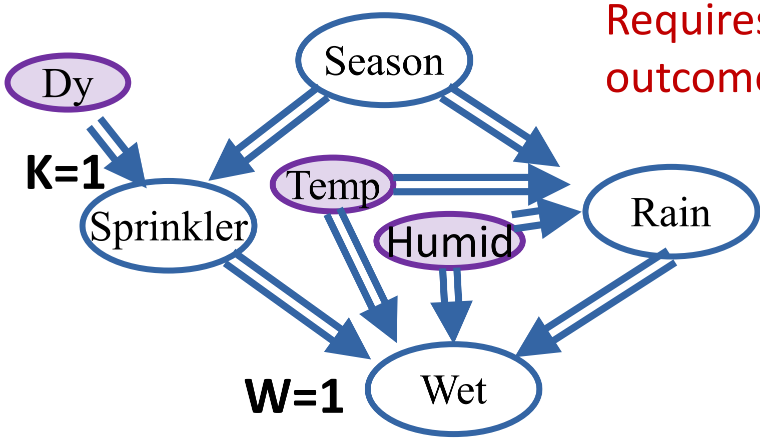
Counterfactual Queries

Counterfactual Query:

Probability of an event in contradiction with the observations

*What **would have happened** if the sprinkler had been turned off?*

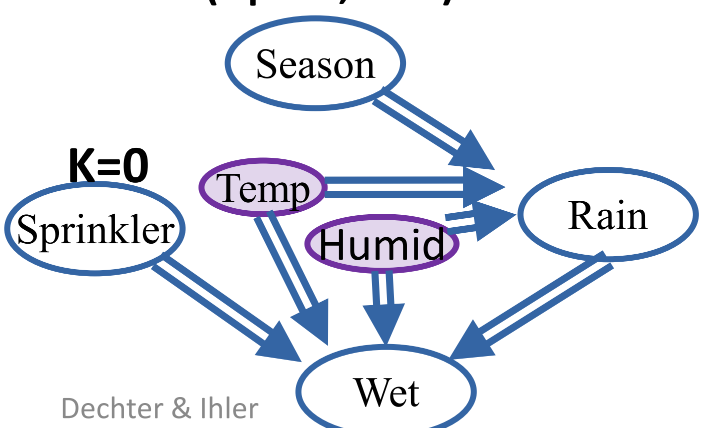
Requires that we transfer information about random outcomes that happened, to a different setting



Observe the sprinkler is on & grass is wet: (K=1,W=1)

What is the probability it would still be wet if we had turned the sprinkler off?

P(S|K=1,W=1)



Abduction: Observing K=1 tells us it is more likely to be summer;
Observing K=1,W=1 tells us it is not too hot & dry.

Action and Prediction: Then, apply this knowledge to compute the counterfactual:

Computing Counterfactuals

Given a model $M=\langle \mathbf{V},\mathbf{U},F,P(\mathbf{u})\rangle$, the conditional probability $P(\mathbf{Y}_x \mid \mathbf{z})$ can be evaluated using the following 3-step procedure:

1. (Abduction) Update $P(\mathbf{u})$ by the evidence $\mathbf{Z}=\mathbf{z}$ to obtain $P(\mathbf{u} \mid \mathbf{z})$.
2. (Action) Modify M with $do(\mathbf{X}=\mathbf{x})$ to obtain $F_{\mathbf{x}}$.
3. (Prediction) Use the model $\langle \mathbf{V},\mathbf{U},F_{\mathbf{x}},P(\mathbf{u} \mid \mathbf{z})\rangle$ to compute the probability of $\mathbf{Y}$.

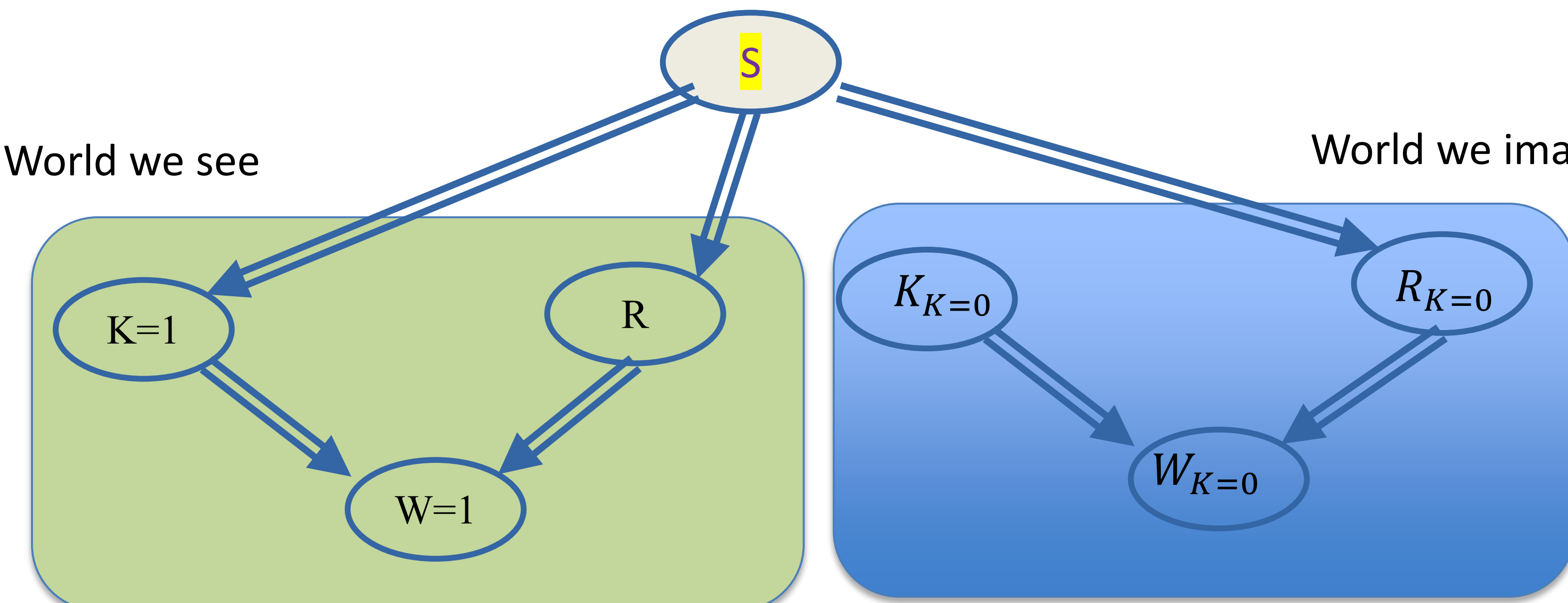
Counterfactual Queries

Ex: Observe the sprinkler is on & grass is wet: (K=1,W=1). What is the probability it would still be wet if we had turned the sprinkler off? Observing K=1 tells us it is

If we have the full model, Counterfactual queries can
Be answered by PGM methods over the twin network model (Classes 1-4)

ph G*

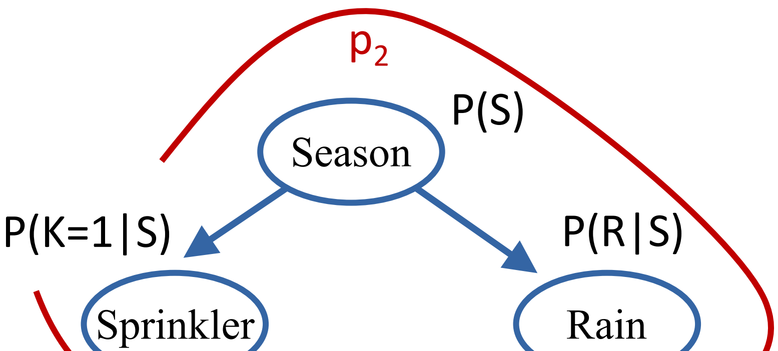
Compute $p(W_{K=0} \mid K=1, W=1)$ in M*



Computing Causal Effects

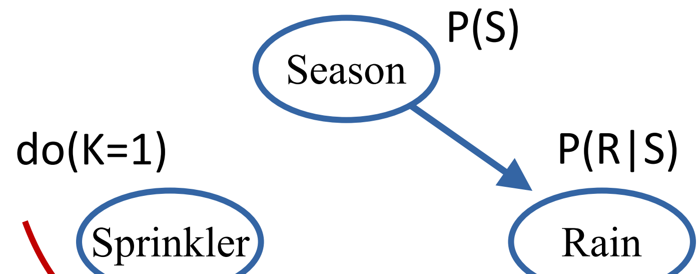
$$p(W|K=1) = ?$$

$$= \frac{\sum_{S,R} p(W|K=1,R)\, p(K=1|S)\, p(R|S)\, P(S)}{\sum_S p(K=1|S)\, p(S)}$$



$$p(W|\mathrm{do}(K=1)) = ?$$

$$= \frac{\sum_{S,R} p(W|K=1,R)\, \mathbb{1}(K=1)\, p(R|S)\, P(S)}{\mathbb{1}(K=1)}$$



If the model is **known**:

- Causal effects and counterfactual queries can be computed using inference
- (classes 1-4)

What if model is **unknown**?

- When is it possible to estimate the causal effect from observed data?
- When possible, how can we do it?

Outline: Causal Inference

- Causal Models: Semantics
- Causal Models: Queries
- **Identifiability**
- Estimand Methods
- Learning Methods

Identifiability

- When can we answer p(Y|do(X)) from observations?

Definition

We say a query p(Y|do(X)) is **identifiable** on graph G if, for any two distributions $p_1(V,U)$, $p_2(V,U)$ on G,

$$p_1(V) = p_2(V) \quad \Rightarrow \quad p_1(Y|do(X)) = p_2(Y|do(X))$$

- Intuition
  - If a query is not identifiable, it cannot be answered uniquely for **any** amount of data – no consistent estimator exists!
  - Conversely, if we can express p(Y|do(X)) in terms of p(V), the query must be identifiable.

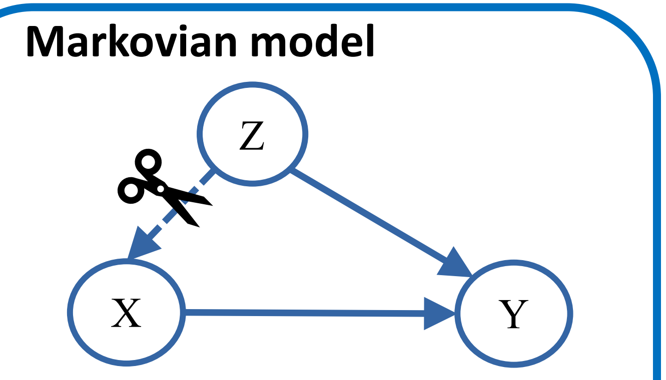
Let's look at a few useful special cases, before the general setting…

Identifiability: Markovian models

- For a Markovian graph G:
  - Causal effect p(Y|do(X)) is identifiable whenever X and all its parents are observed
  - In general,

$$p(Y|\mathrm{do}(X)) = \sum_Z p(Y|X, pa_X)\, p(pa_X)$$

We “adjust” for the values of pa_X!



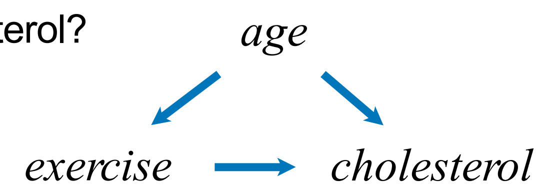
$$p(Y|\mathrm{do}(X)) = \sum_Z p(Y|X, Z)\, p(Z)$$

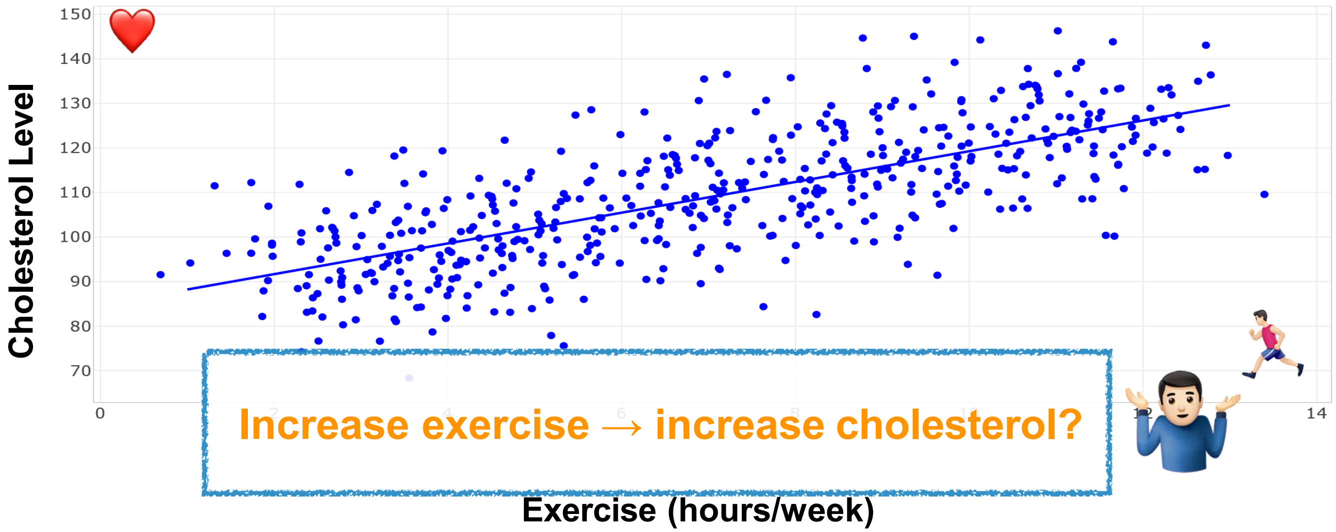
- Why is this necessary?
  - The problem of confounding

Ex: Confounding Bias

What's the causal effect of Exercise on Cholesterol?

What about $P(cholesterol \mid exercise)$?

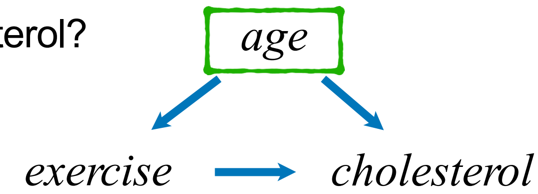


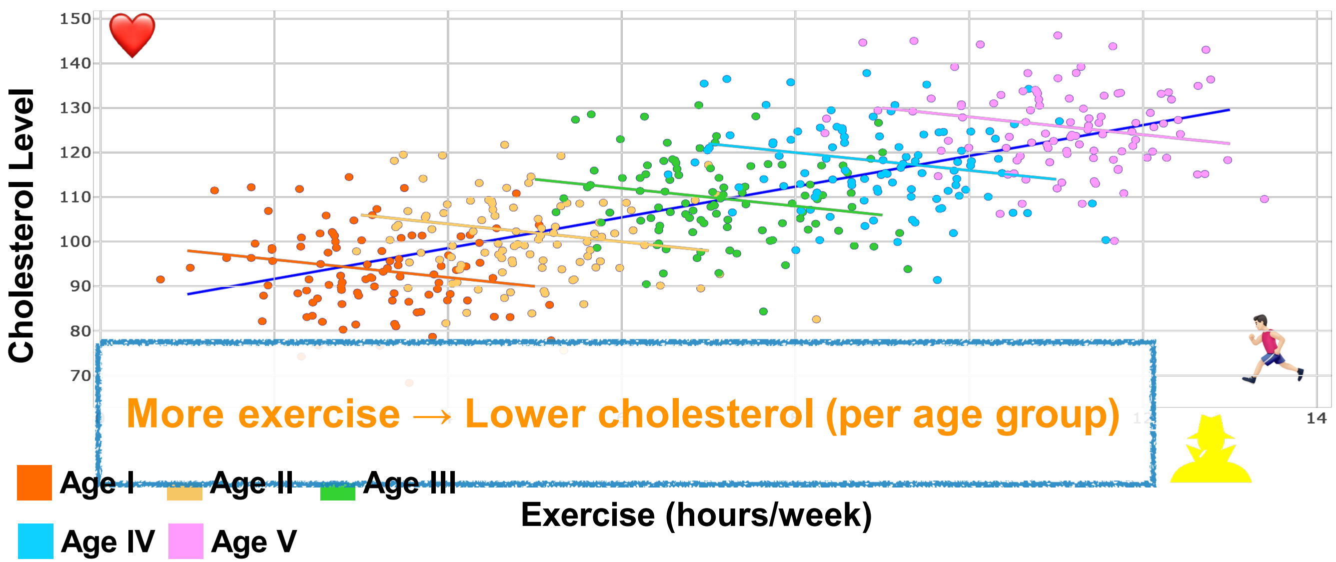


Ex: Confounding Bias

What's the causal effect of Exercise on Cholesterol?

What about *P(cholesterol | exercise)* ?





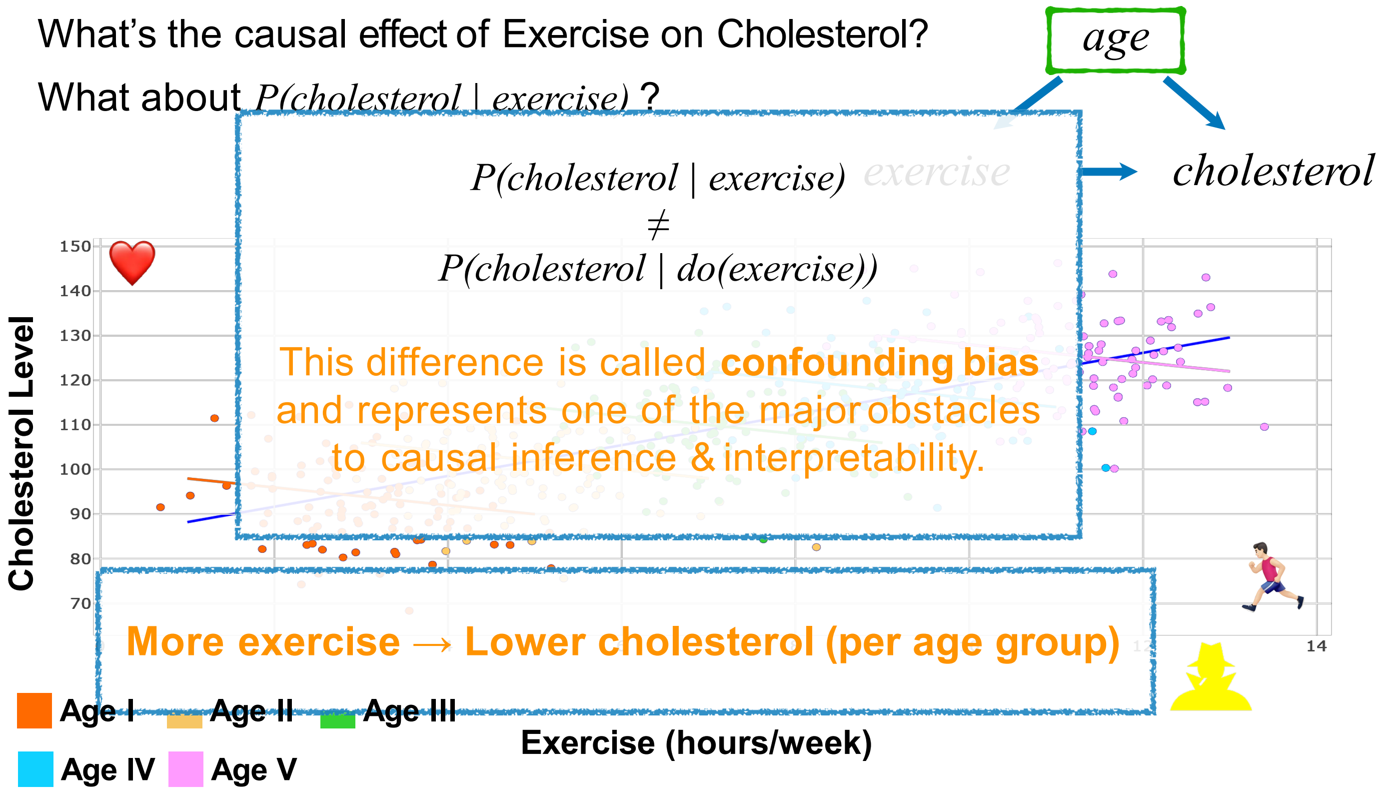
Ex: Confounding Bias

What's the causal effect of Exercise on Cholesterol?

What about *P(cholesterol | exercise)* ?

$$P(cholesterol \mid exercise) \neq P(cholesterol \mid do(exercise))$$

This difference is called **confounding bias** and represents one of the major obstacles to causal inference & interpretability.



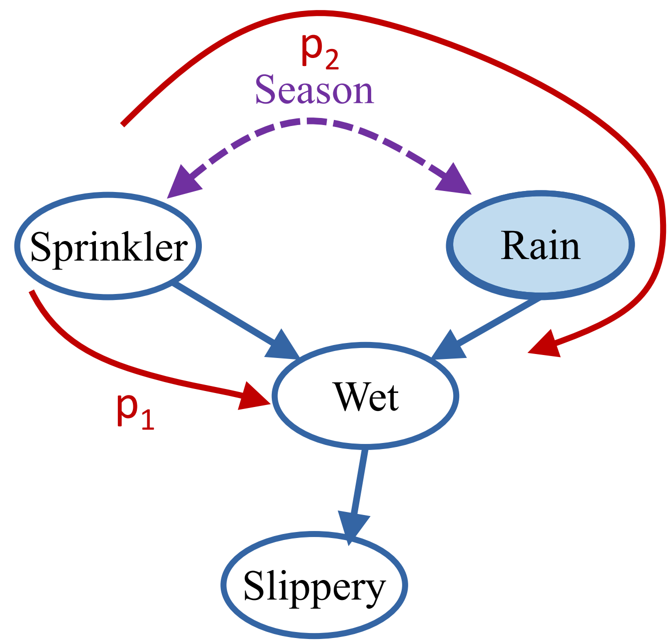
More exercise → Lower cholesterol (per age group)

Identifiability: Backdoor Criterion

- A set Z satisifies the **backdoor criterion** if
  - No Z_i in Z is a descendant of X
  - Z blocks every path between X,Y that has an arrow into X
- Then, $p(Y|\text{do}(X)) = \sum_{Z} p(Y|X,Z)\, p(Z)$

Ex: What if Season is latent?

- Z={Rain} for X=Sprinkler, Y=Wet
  - Conditioning on Rain blocks the non-causal path p_2
  - Leaves the causal path p_1 unaffected!

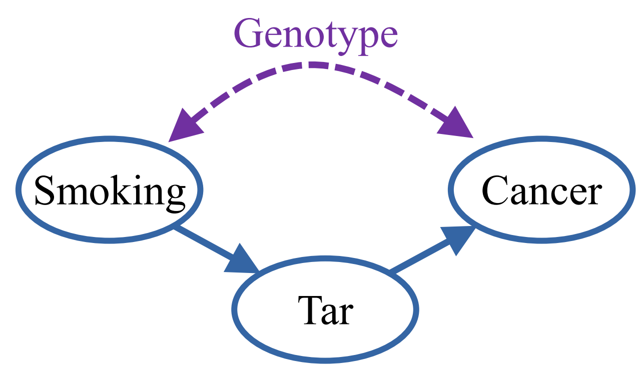


Identifiability: Frontdoor Criterion

- A set Z satisifies the **frontdoor criterion** if
  - Z intercepts all directed paths from X to Y
  - There is no unblocked backdoor path from X to Z
  - All backdoor paths from Z to Y are blocked by X
- Then, $p(Y|\text{do}(X)) = \sum_{Z} p(Z|X) \sum_{X'} p(Y|X', Z)\, p(X')$

Ex: Smoking & Cancer

- Z={Tar} for X=Smoking, Y=Cancer



$$p(Y|\text{do}(X)) = \sum_{Z} p(Z|X) \sum_{X'} p(Y|X', Z)\, p(X')$$

p(Z|do(X)) p(Y|do(Z))

”mediating variable”
in causation process

The Do-Calculus

- Semantics for rewriting expressions with do-operators

Theorem

The following transformations are valid for any do-distribution induced by a causal model M:

Rule 1: Adding/Removing Observations

$$p(y|\text{do}(x), \text{do}(z), w) = p(y|\text{do}(x), z, w) \qquad \text{if} \quad (Z \perp\!\!\!\perp Y \mid X, W)_{G_{\overline{X}\underline{Z}}}$$

Rule 2: Action/Observation Exchange

$$p(y|\text{do}(x), z, w) = p(y|\text{do}(x), w) \qquad \text{if} \quad (Z \perp\!\!\!\perp Y \mid W)_{G_{\overline{X}}}$$

Rule 3: Adding/Removing Actions

$$p(y|\text{do}(x), \text{do}(z), w) = p(y|\text{do}(x), w) \qquad \text{if} \quad (Z \perp\!\!\!\perp Y \mid X, W)_{G_{\overline{X}\,\overline{Z(W)}}}$$

where Z(W) is the set of Z-nodes that are not ancestors of any W-node in $G_{\overline{X}}$

If we can rewrite p(Y|do(X)) in terms of p(V), the query is identifiable!

Algorithmic approach for identification

Truncated Product in Semi-Markovian Models

The distribution generated by an intervention $do(\boldsymbol{X}=\boldsymbol{x})$ in a Semi-Markovian model M is given by the (generalized) truncated factorization product, namely,

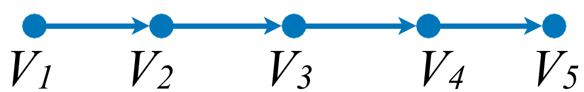
$$P(\mathbf{v} \mid do(\mathbf{x})) = \sum_{\mathbf{u}} \prod_{\{V_i \in \mathbf{V} \setminus \mathbf{X}\}} P\left(v_i \mid pa_i, u_i\right) P(\mathbf{u})$$

And the effect of such intervention on a set $\boldsymbol{Y}$ is

$$P(\mathbf{y} \mid do(\mathbf{x})) = \sum_{\mathbf{v} \setminus (\mathbf{y} \cup \mathbf{x})} \sum_{\mathbf{u}} \prod_{\{V_i \in \mathbf{V} \setminus \mathbf{X}\}} P\left(v_i \mid pa_i, u_i\right) P(\mathbf{u})$$

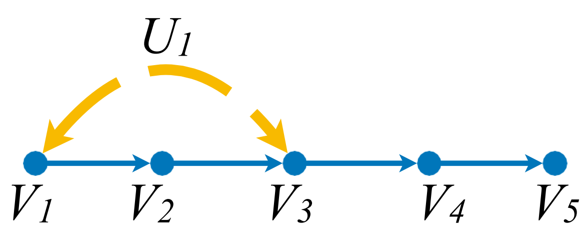
Factorizing the observed distribution

- Start from a simple Markovian model:



$$P(\mathbf{v}) = P(v_1)P(v_2 \mid v_1)P(v_3 \mid v_2)P(v_4 \mid v_3)P(v_5 \mid v_4)$$

- Let's add an unobservable U_1, that affects two observables, and breaking Markovianity:

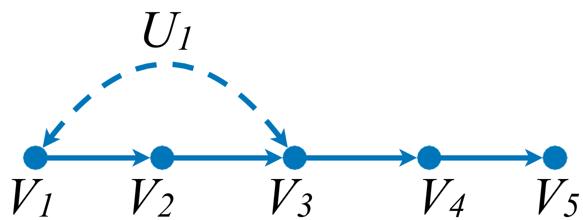


$$P(\mathbf{v}) = \sum_{u_1} P(u_1)\underline{P(v_1 \mid u_1)}P(v_2 \mid v_1)\underline{P(v_3 \mid v_2, u_1)}P(v_4 \mid v_3)P(v_5 \mid v_4)$$

$$= P(v_2 \mid v_1)P(v_4 \mid v_3)P(v_5 \mid v_4)\left(\underline{\sum_{u_1} P(u_1)P(v_1 \mid u_1)P(v_3 \mid v_2, u_1)} \right)$$

Factorizing the observed distribution

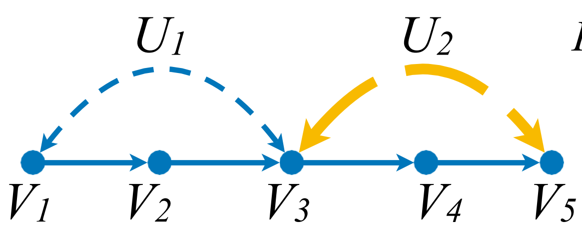
- From the previous model …



$$P(\mathbf{v}) = \sum_{u_1} P(u_1)P(v_1 \mid u_1)P(v_2 \mid v_1)P(v_3 \mid v_2, u_1)P(v_4 \mid v_3)P(v_5 \mid v_4)$$

$$= P(v_2 \mid v_1)P(v_4 \mid v_3)P(v_5 \mid v_4)\left(\sum_{u_1} P(u_1)P(v_1 \mid u_1)P(v_3 \mid v_2, u_1) \right)$$

- Add another unobservable U_2,

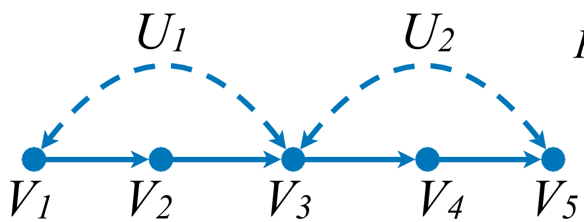


$$P(\mathbf{v}) = \sum_{u_1, u_2} P(u_1, u_2)P(v_1 \mid u_1)P(v_2 \mid v_1)P(v_3 \mid v_2, u_1, u_2)P(v_4 \mid v_3)P(v_5 \mid v_4, u_2)$$

$$= P(v_2 \mid v_1)P(v_4 \mid v_3)\left(\sum_{u_1, u_2} P(u_1, u_2)P(v_1 \mid u_1)P(v_3 \mid v_2, u_1, u_2)P(v_5 \mid v_4, u_2) \right)$$

Factorizing the observed distribution

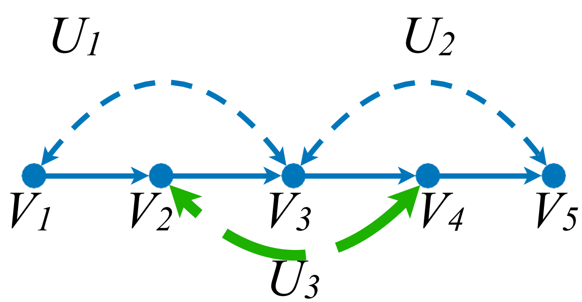
- From the previous model…



$$P(\mathbf{v}) = \sum_{u_1,u_2} P(u_1,u_2)P(v_1 \mid u_1)P(v_2 \mid v_1)P(v_3 \mid v_2,u_1,u_2)P(v_4 \mid v_3)P(v_5 \mid v_4,u_2)$$

$$= P(v_2 \mid v_1)P(v_4 \mid v_3)\left(\sum_{u_1,u_2} P(u_1,u_2)P(v_1 \mid u_1)P(v_3 \mid v_2,u_1,u_2)P(v_5 \mid v_4,u_2)\right)$$

- Let's add one more, U_3,

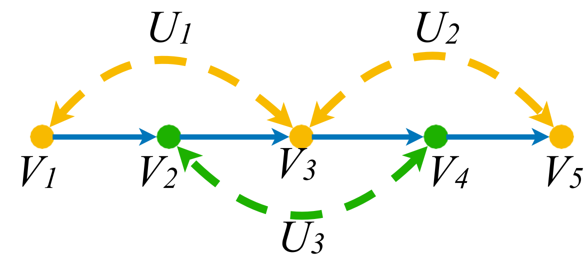


$$P(\mathbf{v}) = \sum_{u_1,u_2,u_3} P(u_1,u_2,u_3)P(v_1 \mid u_1)P(v_2 \mid v_1,u_3)P(v_3 \mid v_2,u_1,u_2)$$
$$P(v_4 \mid v_3,u_3)P(v_5 \mid v_4,u_2)$$

$$= \left(\sum_{u_3} P(u_3)P(v_2 \mid v_1,u_3)P(v_4 \mid v_3,u_3)\right)\left(\sum_{u_1,u_2} P(u_1,u_2)P(v_1 \mid u_1)P(v_3 \mid v_2,u_1,u_2)P(v_5 \mid v_4,u_2)\right)$$

C-Factors

- Recall our example



$$P(\mathbf{v}) = \left(\underbrace{\sum_{u_3} P(u_3)P(v_2 \mid v_1, u_3)P(v_4 \mid v_3, u_3)} \right) \left(\underbrace{\sum_{u_1,u_2} P(u_1, u_2)P(v_1 \mid u_1)P(v_3 \mid v_2, u_1, u_2)P(v_5 \mid v_4, u_2)} \right)$$

- These factors made of sums may be long to write in terms of $P(\mathbf{v},\mathbf{u})$. However, their structure follows from the topology of the diagram, then we can abstract this concept out by defining a new function Q:

$$Q[\mathbf{C}](\mathbf{c}, pa_{\mathbf{c}}) = \sum_{u(\mathbf{C})} P(u(\mathbf{C})) \prod_{V_i \in \mathbf{C}} P(v_i \mid pa_i, u_i) \qquad \text{where} \qquad U(\mathbf{C}) = \bigcup_{V_i \in \mathbf{C}} U_i$$

Then $P(\mathbf{v})$ can be re-written as

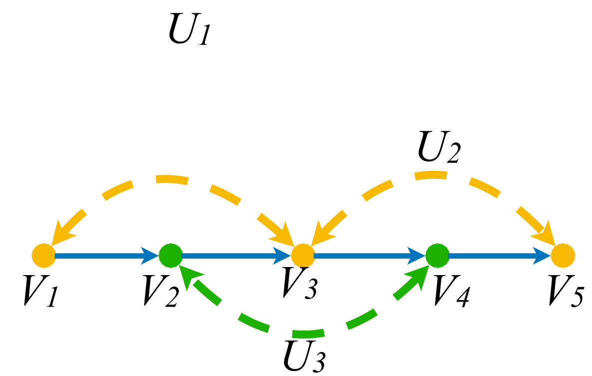
$$P(\mathbf{v}) = \underbrace{Q[V_2, V_4](v_2, v_4, v_1, v_3)}\underbrace{Q[V_1, V_3, V_5](v_1, v_3, v_5, v_2, v_4)}$$

C-Factors

- For convenience $Q[\mathbf{C}](\mathbf{c},pa_c)$ can be written just as $Q[\mathbf{C}]$

- Then, for our example, we can just write

$$P(\mathbf{v}) = Q[V_2, V_4]Q[V_1, V_3, V_5]$$



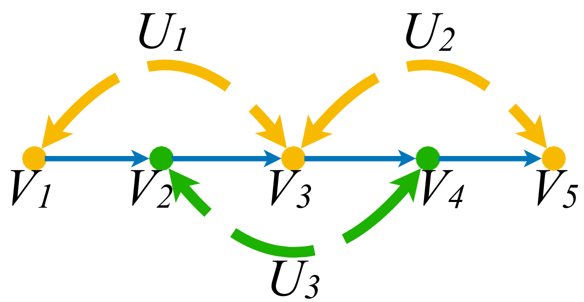
- Note that for the whole set of variables V

$$Q[\mathbf{V}] = \sum_{\mathbf{u}} P(\mathbf{u}) \prod_{V_i \in \mathbf{V}} P(v_i \mid pa_i, u_i) = P(\mathbf{v})$$

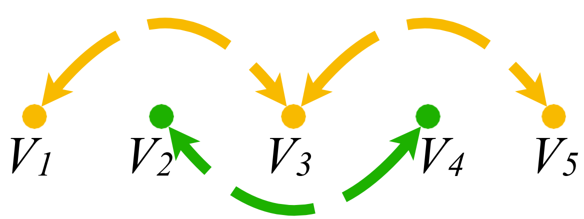
- For consistency define $Q[\varnothing]=1$

Confounding Components

C-components: A partition of the observed variables where any 2 variables connected by a path of bi-directed edges is in the same component.



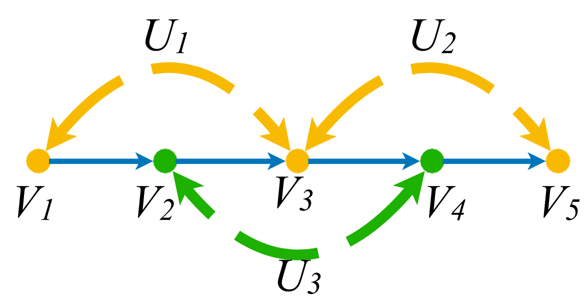
- V_1 is in the same c-component as V_3,
- V_3 is in the same c-component as V_5,
- By extension, V_1 is in the same c-component as V_5 too.
- V_2 is in the same c-component as V_4.



- To see it easily, consider the graph induced over the bidirected edges!
- Obs. The C-Component relation defines a partition over the observable variables, hence it is *Reflexive*, *Symmetric* and *Transitive*.

C-Component Factorization

- The distribution $P(\mathbf{v})$ factorizes into c-factors associated with the c-components of the graph.



$$Q_1 = \{V_2, V_4\} \quad Q_2 = \{V_1, V_3, V_5\}$$

$$P(\mathbf{v}) = \left(\sum_{u_3} P(u_3) P(v_2 \mid v_1, u_3) P(v_4 \mid v_3, u_3) \right) \left(\sum_{u_1, u_2} P(u_1, u_2) P(v_1 \mid u_1) P(v_3 \mid v_2, u_1, u_2) P(v_5 \mid v_4, u_2) \right)$$

$$P(\mathbf{v}) = Q[V_2, V_4]\, Q[V_1, V_3, V_5]$$

C-Component Factorization

- For any $\boldsymbol{H} \subseteq \boldsymbol{V}$, consider a graph $G_{\boldsymbol{H}}$.
- Let $H_1, H_2, \ldots, H_k$ be the c-components of $G_{\boldsymbol{H}}$.
- Then

$$Q[\mathbf{H}] = \prod_j Q[H_j]$$

And, the Q factor of any c component can be computed from Q(H)

C-factor Algebra - Summary

We have two basic operations over c-factors

1. Reduce to an ancestral set

$$Q[\mathbf{W}] = \sum_{\mathbf{c}\setminus\mathbf{w}} Q[\mathbf{C}]$$

If $\boldsymbol{W}$ is ancestral in $G_{\boldsymbol{C}}$

2. Factorize into c-components

$$Q[\mathbf{H}] = \prod_{j} Q[H_j]$$

Where $H_1, \ldots H_k$, are the c-components in $G_{\boldsymbol{H}}$

The Identification Algorithm

- Given G and the query variables $\boldsymbol{X}, \boldsymbol{Y}$

$$P(\mathbf{y} \mid do(\mathbf{x})) = \sum_{\mathbf{v} \setminus (\mathbf{x} \cup \mathbf{y})} Q[\mathbf{V} \setminus \mathbf{X}]$$

$$= \sum_{\boldsymbol{d} \setminus \mathrm{y}} Q[\boldsymbol{D}]$$

where $\boldsymbol{D} = An(\boldsymbol{Y})$ in $G_{\overline{\mathbf{X}}}$

- Suppose the graph $G_{\boldsymbol{D}}$ has C-components $\mathbf{D}_1, \mathbf{D}_2, \ldots, \mathbf{D}_k$, then

$$P(\mathbf{y} \mid do(\mathbf{x})) = \sum_{\mathbf{d} \setminus \mathbf{y}} \prod_{i} Q[\mathbf{D}_i]$$

The Identification Algorithm

- If we can identify each $Q[D1], Q[D2], \ldots, Q[Dk]$, from $P(\mathbf{v}) = Q[V] = Q[C_1] \ldots Q[C_m]$, we obtain an expression equal to $P(\mathbf{y}|do(\mathbf{x}))$. An algorithm for computing Q[C] from Q[T] for C, T being c-components:

```
Identify(C, T, Q, G)
1. Let A=An(C) in G_T.
2. If A=C return Q[C]=Σ_{t\c}Q.
3. If A=T return Fail.
4. Let A_i be the c-comp of G_A that contains C.
   Get Q[A_i] from Q.
5. Return Identify(C, A_i, Q[A_i], G).
```

Completeness

Theorem [Huang and Valtorta, 2008]

The causal effect $P(\mathbf{y}|do(\mathbf{x}))$ is identifiable from causal diagram G and $P(\mathbf{v})$ *if and only if* each of the C-factors $Q[\mathbf{D_i}$ is identifiable by `Identify(Di,Ci,Q[Ci],G)`.

Where $\mathbf{C}_i$ is the C-component of G containing $\mathbf{D}_i$.

Examples of Estimand Expressions

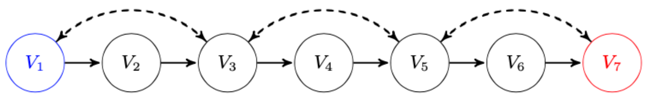
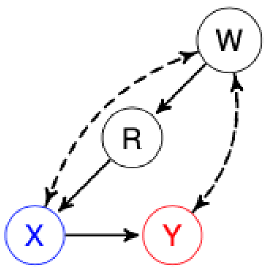


Figure 1: Chain Model with 7 observable variables and 3 latent variables

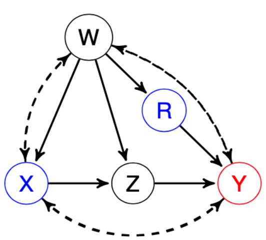
$$P(V_7 \mid do(V_1)) = \sum_{V_2,V_3,V_4,V_5,V_6} P(V_6 \mid V_1,V_2,V_3,V_4,V_5)P(V_4 \mid V_1,V_2,V_3)P(V_2 \mid V_1)$$
$$\times \sum_{V_1'} P(V_7 \mid V_1',V_2,V_3,V_4,V_5,V_6)P(V_5 \mid V_1',V_2,V_3,V_4)P(V_3 \mid V_1',V_2)P(V_1')$$



(a) Model 1

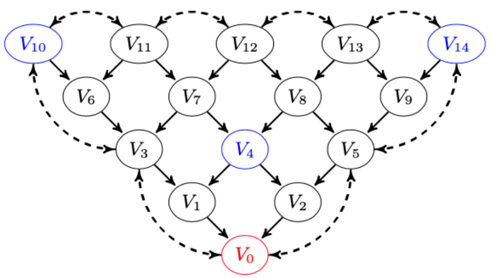
Estimand Expressions for Models 1 & 8 .

Model	Estimate of $P(Y \mid do(X))$
1	$\frac{\sum_W P(X,Y \mid R,W)P(W)}{\sum_W P(X \mid R,W)P(W)}$
8	$\sum_{R,W,Z} P(Z \mid R,W,X)P(R \mid W) \sum_x P(Y \mid R,W,x,Z)P(x \mid R,W)P(W)$



(c) Model 8

Examples of Estimand Expressions



(b) Cone Cloud, $n = 15$ (15-CC)

$$
\begin{aligned}
P(V_0|V_{14}, V_{10}, V_4) = & \\
\sum_{V_1,V_2,V_3,V_5,V_6,V_7,V_8,V_9,V_{11},V_{12},V_{13},v'_{10},V'_{11},V'_{12},V'_{13},V'_{14}} & P(V_2|V_4, V_5, V_7, V_8, V_9, V_{11}, V_{12}, V_{13}, V_{14})\times \\
& P(V_9|V_{13}, V_{14})P(V_8|V_{12}, V_{13})P(V_1|V_3, V_4, V_6, V_7, V_8, V_{10}, V_{11}, V_{12}, V_{13})\times \\
& P(V_7|V_{11}, V_{12})P(V_6|V_{10}, V_{11})P(V_{11}, V_{12}, V_{13})\times \\
& P(V_0|V_1, V_2, V_3, V_4, V_5, V_6, V_7, V_8, V_9, V'_{10}, V_{11}, V_{12}, V_{13}, V'_{14})\times \\
& P(V_5|V_1, V_3, V_4, V_6, V_7, V_8, V_9, V'_{10}, V_{11}, V_{12}, V_{13}, V'_{14})\times \\
& P(V'_{14}|V_1, V_3, V_4, V_6, V_7, V_8, V'_{10}, V_{11}, V_{12}, V_{13})\times \\
& P(V_3, V_{13}|V_6, V_7, V'_{10}, V_{12}, V_{13})P(V'_{10}|V_7, V_{11}, V_{12})P(V_{11}, V_{12}) \qquad (7)
\end{aligned}
$$

An estimand often corresponds to inference over a Bayesian network
Which is sometime very dense.

The treewidth of the above example is sqrt of n, when n is the number of variables

So, is evalution Exp(w)?.

Outline: Causal Inference

- Causal Models: Semantics
- Causal Models: Queries
- Identifiability
- **Estimand Methods**
- Learning Methods

The Plug-in estimate

- The Plug-in methods uses the “empirical distributions extracted from the data to estimate observed probabilistic quantities.
- Complexity of generating a table is O(|D|).
- Complexity of evaluation is exponential in the hyper-tree width.
- Computation can explore the graph and sparseness of the probabilistic quantities.

Empirical Factors, Sparse Representation

Table 80: 6 Variables with domain size 3

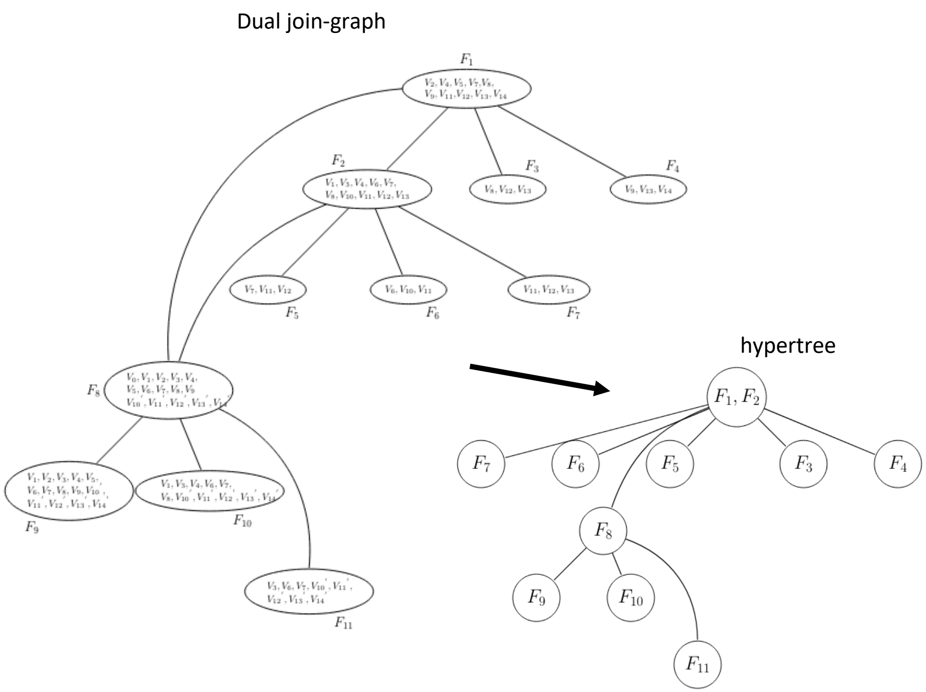
(a) Data Table

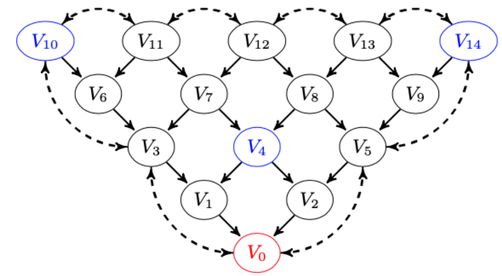
A	B	C	D	E	F
0	1	2	0	1	2
1	2	0	1	2	0
2	0	1	2	0	1
1	0	2	1	0	2
2	1	0	2	1	0
0	1	2	0	1	2
1	2	0	1	2	0
2	0	1	2	0	1
1	0	2	1	0	2
0	1	2	0	1	2
1	2	0	1	2	0
2	0	1	2	0	1
1	0	2	1	0	2
2	1	0	2	1	0
0	0	0	0	0	0
1	2	1	0	2	0
0	1	0	2	1	1
0	0	0	0	0	0
2	0	1	2	0	1
0	2	2	1	0	1

(b) Sparse Factor Table

A	B	C	D	E	F	Probability
0	1	2	0	1	2	0.20
1	2	0	1	2	0	0.20
2	0	1	2	0	1	0.20
1	0	2	1	0	2	0.15
2	1	0	2	1	0	0.10
0	0	0	0	0	0	0.10
1	2	1	0	2	0	0.05
0	1	0	2	1	1	0.05
0	2	2	1	0	1	0.05

Estimands Tree-Decomposition (Cones)





(b) Cone Cloud, $n = 15$ (15-CC)

$$
\begin{aligned}
P(V_0|V_{14}, V_{10}, V_4) = &\\
\sum_{V_1,V_2,V_3,V_5,V_6,V_7,V_8,V_9,V_{11},V_{12},V_{13},v'_{10},V'_{11},V'_{12},V'_{13},V'_{14}} & P(V_2|V_4, V_5, V_7, V_8, V_9, V_{11}, V_{12}, V_{13}, V_{14})\times \\
& P(V_9|V_{13}, V_{14})P(V_8|V_{12}, V_{13})P(V_1|V_3, V_4, V_6, V_7, V_8, V_{10}, V_{11}, V_{12}, V_{13})\times \\
& P(V_7|V_{11}, V_{12})P(V_6|V_{10}, V_{11})P(V_{11}, V_{12}, V_{13})\times \\
& P(V_0|V_1, V_2, V_3, V_4, V_5, V_6, V_7, V_8, V_9, V'_{10}, V_{11}, V_{12}, V_{13}, V'_{14})\times \\
& P(V_5|V_1, V_3, V_4, V_6, V_7, V_8, V_9, V'_{10}, V_{11}, V_{12}, V_{13}, V'_{14})\times \\
& P(V'_{14}|V_1, V_3, V_4, V_6, V_7, V_8, V'_{10}, V_{11}, V_{12}, V_{13})\times \\
& P(V_3, V_{13}|V_6, V_7, V'_{10}, V_{12}, V_{13})P(V'_{10}|V_7, V_{11}, V_{12})P(V_{11}, V_{12})
\end{aligned}
\tag{7}
$$

hw=2,w=14

Hyper-tree width: is the mximum number of functions placed in any cluster of a tree-decomposition

Complexity of Plug-In Scheme

Theorem: The complexity of evaluating an extimands whose expression has a tree-decomposition having hyper tree-width hw is O(n t^{hw}) if it has no denominators, where n = number of variables, t is the data size and k is the variables domain size. The complexity is also exponential in the tree-width is O(n k^{w}).

In all the examples we saw hw=1,2. w=1 or O($\sqrt{n}$).

But what about statistical accuracy?

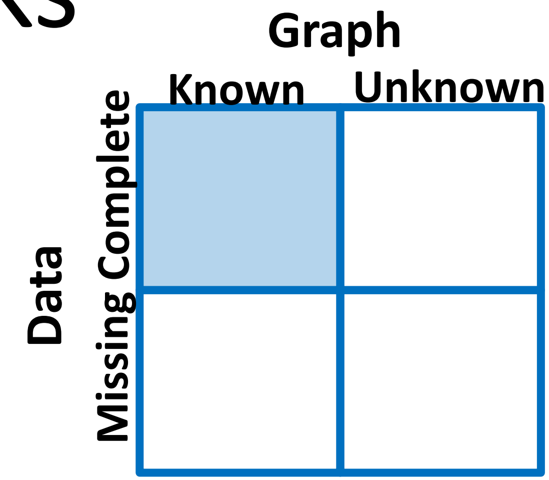
Note: The Plug-in can be viewed as using *maximum –likelihood* learning when data is fully observed over a Bayesian network graph extracted from the estimand.

Outline: Causal Inference

- Causal Models: Semantics
- Causal Models: Queries
- Identifiability
- Estimand Methods
- **Learning Methods**

Learning Bayesian networks

- Maximum Likelihood estimation
  - Select model that makes the data most probable
- For discrete X_i & no shared parameters
  - ML estimates are empirical probabilities

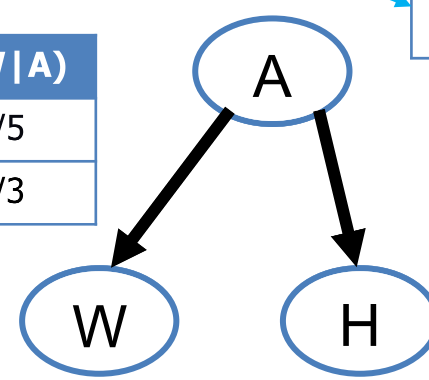


Observed Data

A	W	H
0	0	0
0	0	0
0	1	0
0	1	1
0	0	0
1	0	1
1	1	0
1	1	1

P(A)
3/8

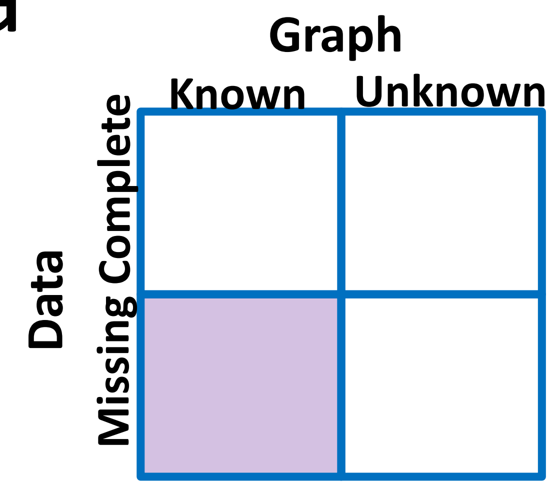
A	P(W|A)
0	2/5
1	2/3



A	P(H|A)
0	1/5
1	2/3

Learning with missing data

- Latent / hidden variables
  - Value is never observed
  - No unique model (e.g., symmetry)
  - No closed form solution; iterative ML
- More general missing values?
  - May depend on the reason for missingness!

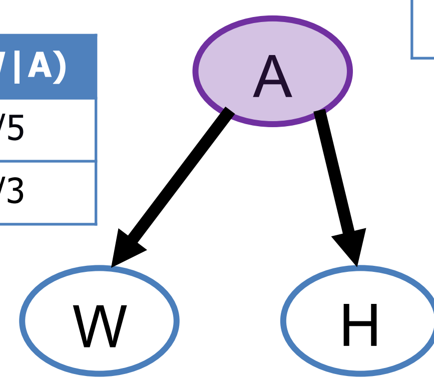


Observed Data

A	W	H
?	0	0
?	0	0
?	1	0
?	1	1
?	0	0
?	0	1
?	1	0
?	1	1

P(A)
3/8

A	P(W|A)
0	2/5
1	2/3



A	P(H|A)
0	1/5
1	2/3

Learning-Based Approach

Motivation: Use PGM algorithms for Causal Reasoning

$$P(V_7 \mid do(V_1)) = \sum_{V_2,V_3,V_4,V_5,V_6} P(V_6 \mid V_1,V_2,V_3,V_4,V_5) P(V_4 \mid V_1,V_2,V_3) P(V_2 \mid V_1)$$
$$\times \sum_{V_1'} P(V_7 \mid V_1',V_2,V_3,V_4,V_5,V_6) P(V_5 \mid V_1',V_2,V_3,V_4) P(V_3 \mid V_1',V_2) P(V_1')$$

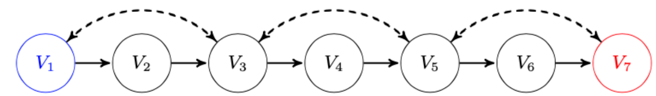


Figure 1: Chain Model with 7 observable variables and 3 latent variables

Motivation: Bucket-elimination on this network has tree-width 3 while the observational distribution has a tree-width of 7 and hypertree width of 1

Learning Idea:
Input: G, P(V)
1. Learn a full causal BN, from G and samples from P(V) using **EM**.
2. Truncate the learned model B into B_x.
3. Compute $P(\boldsymbol{Y})$ by Bucket elimination over B_x and return.
End Algorithm

Accuracy: EM is a mximum likelihood learning scheme that converges to a local maxima.
Complexity of Inference of both learning and inference is exponential in the **tree-width.**

Theorem: Given a model M yielding observational distribution P(V) and graph G, then any causal Bayesian Network over G having the same P(V), will agree with M **on any identifiable** causal effect query P(Y|do(X)).

Learning-Based Approach

What about the latent variables? Their domains?

Fit a good domain size for latent variables using the BIC score.

$$BIC_{\mathcal{B},\mathcal{D}} = -2 \cdot LL_{\mathcal{B},\mathcal{D}} + p \cdot \log(|\mathcal{D}|)$$

Algorithm 3: EM4CI

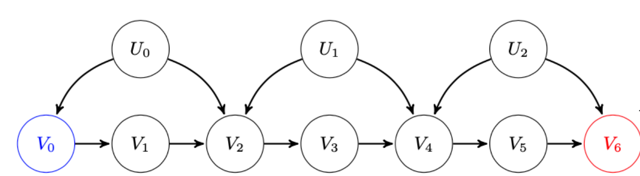
input : A causal diagram $\mathcal{G} = \langle \boldsymbol{U} \cup \boldsymbol{V}, E \rangle$, $\boldsymbol{U}$ latent and $\boldsymbol{V}$ observables; $\mathcal{D}$ samples from $P(\boldsymbol{V})$;
Query $Q = P(\mathbf{Y} \mid do(\mathbf{X} = \mathbf{x}))$.

output: Estimated $P(\mathbf{Y} \mid do(\mathbf{X} = \mathbf{x}))$

// k= latent domain size, $BIC_{\mathcal{B}} = BIC$ score of $\mathcal{B}$, $\mathcal{D}$,
// $LL_{\mathcal{B}}$ is the log-likelihood of $\mathcal{B}$, $\mathcal{D}$
Step 1: Initialize: $BIC_{\mathcal{B}} \leftarrow \inf$,
If $\neg$identifiable$(\mathcal{G}, Q)$, terminate.
For $k = 2, ...,$ to upper bound, **do**
  $(LL_{\mathcal{B}_{new}}) \leftarrow \max_{LL}\{\{\text{EM}(\mathcal{G}, \mathcal{D}, k) \mid i = 1, 2, \ldots, 10\}\}$;
  Calculate $BIC_{\mathcal{B}_{new}}$ from $LL_{\mathcal{B}_{new}}$
  If $BIC_{\mathcal{B}_{new}} \leq BIC_{\mathcal{B}}$,
    $\mathcal{B} \leftarrow \mathcal{B}_{new}$, $BIC_{\mathcal{B}} \leftarrow BIC_{\mathcal{B}_{new}}$
  else, break.
Endfor
$\mathcal{B}_{\mathbf{X}=\mathbf{x}} \leftarrow$ generate truncated CBN from $\mathcal{B}$.
return $\leftarrow$ evaluate $P_{\mathcal{B}_{\mathbf{X}=\mathbf{x}}}(\mathbf{Y})$

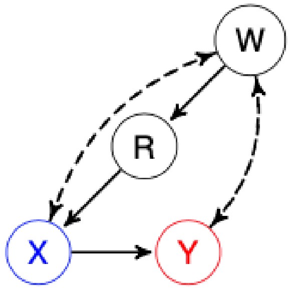
latent domains, CPT's parameters

U= {1,2,3}



P(V6|V5,U2)

Empirical Evaluation: Small Graphs

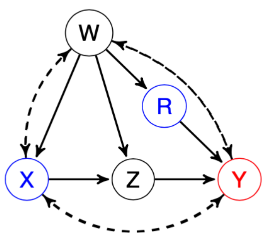


(a) Model 1

Small Synthetic models

Estimand Expressions for Models 1 & 8 .

Model	Estimate of $P(Y \mid do(X))$
1	$\frac{\sum_W P(X,Y\mid R,W)P(W)}{\sum_W P(X\mid R,W)P(W)}$
8	$\sum_{R,W,Z} P(Z\mid R,W,X)P(R\mid W)\sum_x P(Y\mid R,W,x,Z)P(x\mid R,W)P(W)$



(c) Model 8

Results for EM4CI and Plug-in estimates on $P(Y=y|do(X=0))$, $(d,k)=(2,10)$, k_{lrn} is the learned domain sizes of latent variables.

	100 Samples				1,000 Samples			
	EM4CI			Plugin	EM4CI			Plugin
Model	(LL,BIC)	(mad,time(s))	k_{lrn}	(mad,time)	(LL,BIC)	(mad,time(s))	k_{lrn}	(mad,time)
1	(-79,167)	(0.00348, 0.13)	2	(0.015, 0.016)	(-716,1445)	(0.003475, 1.5)	2	(0.002, 0.21)
8	(-121,262)	(0.0373, 0.61)	2	(0.288 , 0.016)	(-1290, 2609)	(0.0083, 6.6)	2	(0.154, 0.248)

Empirical Evaluation: Large Synthetic

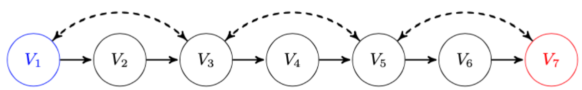
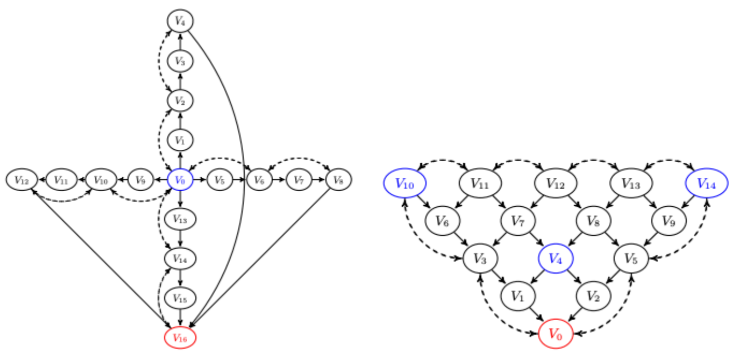


Figure 1: Chain Model with 7 observable variables and 3 latent variables



(a) Diamond, $n = 17$ **(b)** Cone Cloud, $n = 17$

More accuracy for learning in all cases but 1

Table 78: Results for EM4CI and Plug-In on $P(Y|do(\mathbf{X}))$ $(d, k) = (4, 10)$

(a) 1,000 samples

Model	Query	k_{lrn}	EM4CI mad	EM4CI Learn-time(s)	EM4CI inf-time(s)	Plug-In mad	Plug-In time(s)
5-CH	$P(V_4\|do(V_0))$	4	0.0902	3.5	0.0001	0.1509	2.3
9-CH	$P(V_8\|do(V_0))$	4	0.1204	11.5	0.0002	0.1516	2.4
25-CH	$P(V_24\|do(V_0))$	2	0.0070	77.7	0.0003	0.0959	6.1
49-CH	$P(V_48\|do(V_0))$	4	0.0005	161.2	0.0007	0.0319	17.8
99-CH	$P(V_98\|do(V_0))$	6	0.0093	413.4	0.0023	0.0611	88.1
9-D	$P(V_8\|do(V_0))$	2	0.0719	24.6	0.0002	0.1832	3.4
17-D	$P(V_{16}\|do(V_0))$	6	0.0542	202.3	0.0006	0.0700	4.5
65-D	$P(V_{64}\|do(V_0))$	4	0.0074	432.4	0.0012	0.1716	232.5
6-CC	$P(V_0\|do(V_5))$	4	0.0088	23.5	0.0001	0.0156	2.3
15-CC	$P(V_0\|do(V_{14}))$	4	0.0147	60.8	0.0001	0.0659	4.5
45-CC	$P(V_0\|do(V_{14}, V_{36}, V_{44}))$	6	0.0097	199.2	2.7429	0.1509	18.6

(b) 10,000 samples

Model	Query	k_{lrn}	EM4CI mad	EM4CI Learn-time(s)	EM4CI inf-time(s)	Plug-In mad	Plug-In time(s)
5-CH	$P(V_4\|do(V_0))$	4	0.0508	17.3	0.0001	0.0537	2.5
9-CH	$P(V_8\|do(V_0))$	4	0.0236	150.0	0.0002	0.1074	3.1
25-CH	$P(V_24\|do(V_0))$	6	0.0068	697.1	0.0005	0.0714	26.4
49-CH	$P(V_48\|do(V_0))$	10	0.0017	2412.6	0.0036	0.0160	133.7
99-CH	$P(V_98\|do(V_0))$	6	0.0028	3887.9	0.0022	0.0433	850.6
9-D	$P(V_8\|do(V_0))$	4	0.0611	390.7	0.0002	0.1481	3.0
17-D	$P(V_{16}\|do(V_0))$	6	0.0360	1849.6	0.0007	0.0582	8.4
65-D	$P(V_{64}\|do(V_0))$	4	0.0022	4787.2	0.0013	0.1376	2258.5
6-CC	$P(V_0\|do(V_5))$	6	0.0138	116.9	0.0003	0.0136	2.7
15-CC	$P(V_0\|do(V_{14}))$	4	0.0022	489.5	0.0043	0.0321	10.9
45-CC	$P(V_0\|do(V_{14}, V_{36}, V_{44}))$	6	0.0026	1833.7	2.757	0.1561	105.8

Dependence on Model Size

Trajectory over Cone Instances **Trajectory over Chain Instances** **Trajectory over Diamond Instances**

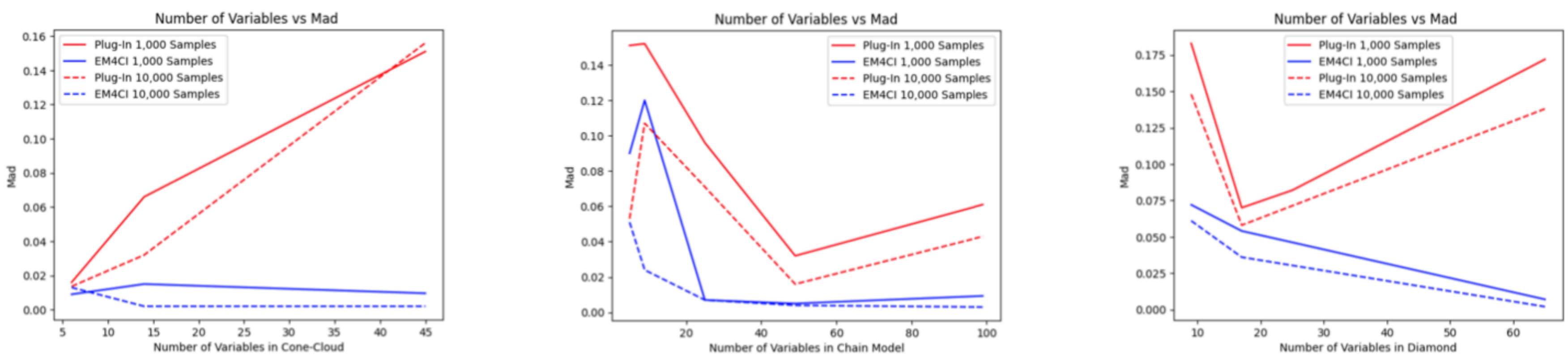


Figure 4: Comparing the accuracy of EM4CI and Plug-In. While both methods improve with more samples (solid to dashed lines), the error (mad) of EM4CI is smaller, even when compared to Plug-In with more samples.

Empirical Evaluation: Results

- **4 Real Networks**
- “Alarm”, “A”, “Barley”, and “Win95pts” network

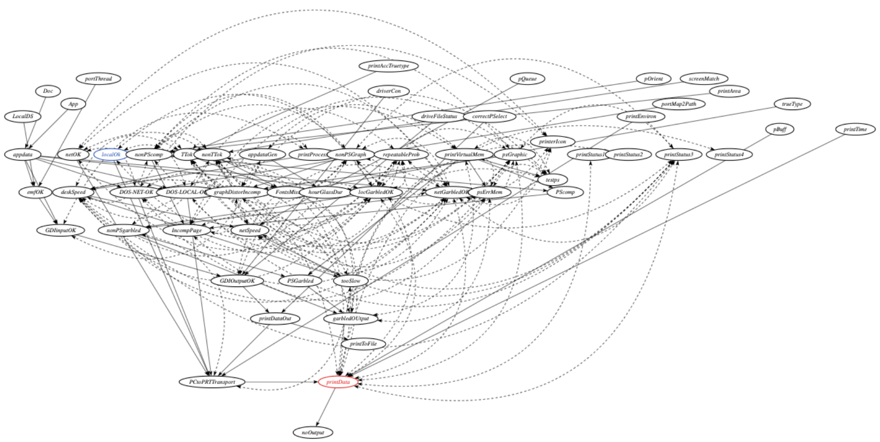


Figure: A Model

Table 8: Plug-In & EM4CI results on the **A Network**
$|\boldsymbol{V}| = 46;\ |\boldsymbol{U}| = 8;\ d = 2;\ k = 2$ treewidth ≈ 16

(a) Plug-In

	1,000 Samples		*10,000 Samples*	
Query	mad	time(s)	mad	time(s)
$P(V_{51}\|\text{do}(V_{10}))$	0.0584	8.0	0.0114	55.7
$P(V_{51}\|\text{do}(V_{14}))$	0.0319	8.3	0.0056	51.3
$P(V_{51}\|\text{do}(V_{41}))$	0.0255	13.9	0.0092	48.3
$P(V_{51}\|\text{do}(V_{45}))$	0.0496	9.8	0.0206	49.1

(b) EM4CI

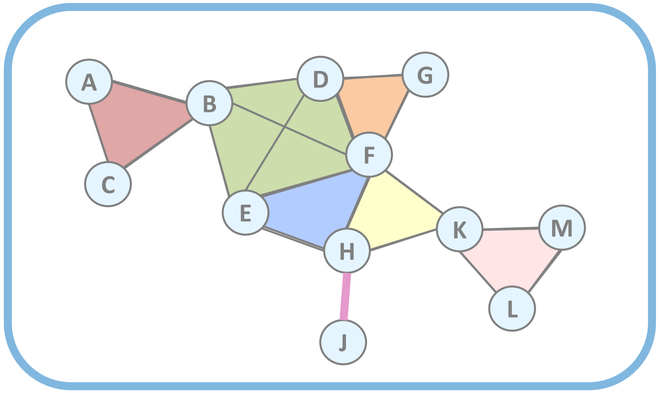
	1,000 Samples		*10,000 Samples*	
Learning	$time = 71(s)$	$k_{lrn} = 4$	$time(s) = 541$	$k_{lrn} = 4$
Inference: Query	mad	time(s)	mad	time(s)
$P(V_{51}\|\text{do}(V_{10}))$	0.0139	0.0012	0.0083	0.0012
$P(V_{51}\|\text{do}(V_{14}))$	0.0143	0.0047	0.0086	0.0046
$P(V_{51}\|\text{do}(V_{41}))$	0.0147	0.0042	0.0079	0.0041
$P(V_{51}\|\text{do}(V_{45}))$	0.0140	0.0031	0.0082	0.0030

Summary: Causal queries

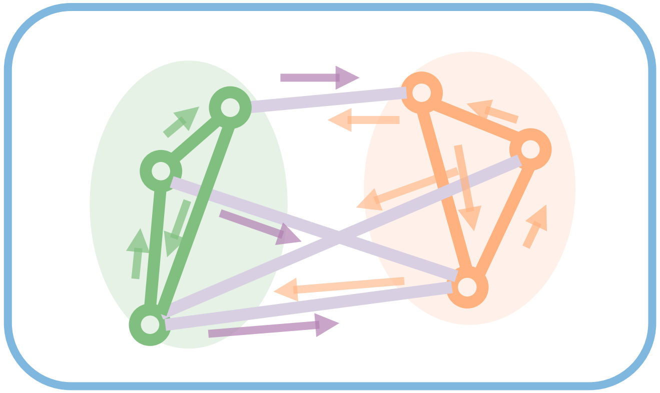
- Causal Bayesian networks and SCM encodes causal assumptions explicitly.
- Causal effects and counterfactual queries can be computed from the full causal model by PGM methods.
- Given only the causal diagram and observational data queries can be evaluated if identifiable.
- Causal effect queries can be done by statistical estimation of estimands (defined by observation quantities).
- Estimands can be generated by Backdoor, Frontdoor, do-calculous. The ID algorithm.
- Model completion by learning is a promising alternative for causal inference that exploit PGM methods.

Summary of Lectures

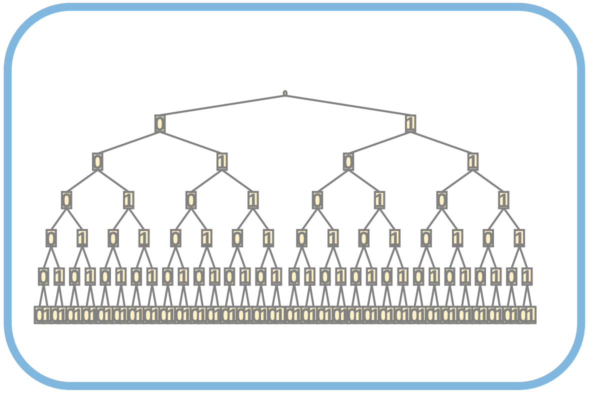
Class 1: Introduction & Inference



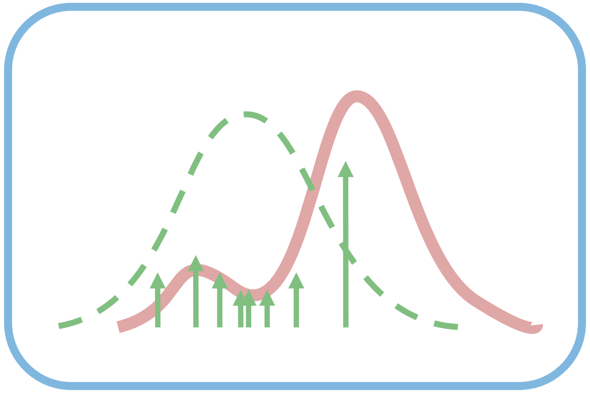
Class 2: Bounds & Variational Methods



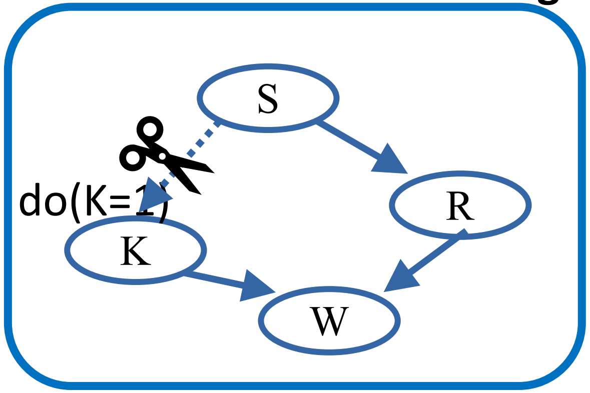
Class 3: Search Methods



Class 4: Monte Carlo Methods



Class 5: Causal Reasoning



Software

- My software page

pyGMs : Python Toolbox for Graphical Models by Alexander Ihler.

UAI Probabilistic Inference Competitions

- **2006**

(aolib)

- **2008**

(aolib)

- **2012**

(daoopt)

- **2014**

(daoopt)

(daoopt)

(merlin)

Marginal Map

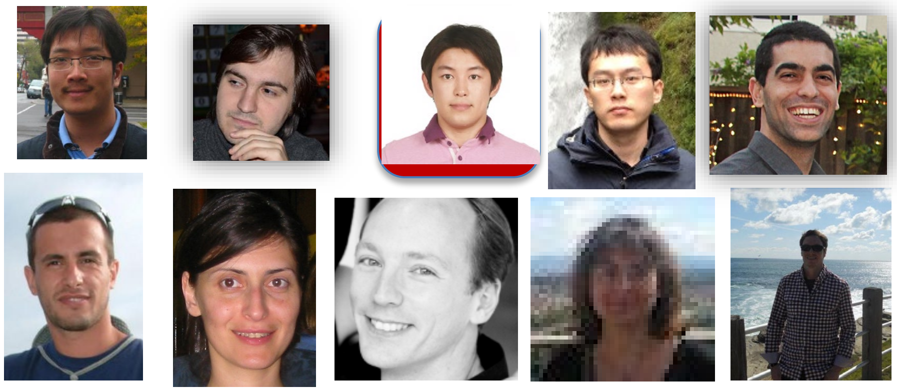
New UAI Competition

- [UAI Competition 2022]

Solver	20sec	1200sec	3600sec
uai14-pr	61.7	96.8	96.7
ibia-pr	53.6	96.6	97.1
AbstractionSampling	78.9	91.7	93.9
lbp-pr	90.3	89.9	90.2

Thank You !

For publication see:
http://www.ics.uci.edu/~dechter/publications.html



Rina Dechter
Alex Ihler
Kalev Kask
Irina Rish
Bozhena Bidyuk
Robert Mateescu
Radu Marinescu
Vibhav Gogate
Lars Otten
Natalia Flerova
Andrew Gelfand
William Lam
Filjor Broka
Junkyu Lee
Qi Lou
Bobak Pezeshki