Structure Learning in Bayesian Networks: The PC Algorithm

Mohammad Ali Javidian

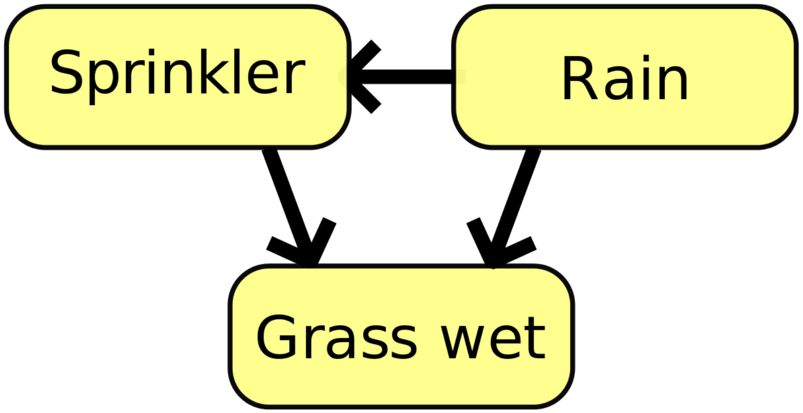
April 2020

Overview

- Bayesian Networks and Learning
- Structure Learning in Bayesian Networks
- Markov Condition, Faithfulness and Causal Sufficiency Assumptions
- Markov Equivalence in Bayesian Networks
- PC algorithm
- Complexity
- Discussion

Bayesian Networks and Learning

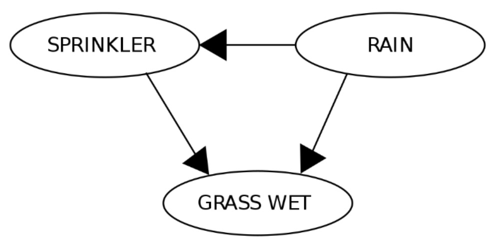
- A Bayesian network consists of two main components:
  1) a *directed acyclic graph*,



Bayesian Networks and Learning

- A Bayesian network consists of two main components:
  1) a *directed acyclic graph*,
  2) a joint probability distribution (To each variable A with parents B1, . . . , Bn, a *__c__onditional __p__robability __t__able* P(A | B1, . . . , Bn) is attached.)

	SPRINKLER	
RAIN	T	F
F	0.4	0.6
T	0.01	0.99



RAIN	
T	F
0.2	0.8

		GRASS WET	
SPRINKLER	RAIN	T	F
F	F	0.0	1.0
F	T	0.8	0.2
T	F	0.9	0.1
T	T	0.99	0.01

Structure Learning in Bayesian Networks

1) Constraint-based structure learning:
   - Bayesian network as a representation of independencies
   - Based on conditional independence tests in the data
   - Example: PC[1], Grow-Shrink[2], Incremental Association based on the Markov blanket discovery (IAMB)[3]
2) Score-based structure learning:
   - Learning as a *model selection* problem
   - Example: Hill-Climbing[4], Tabu Search[4]
3) Hybrid algorithms:
   - Example: Max-Min Hill-Climbing (MMHC)[5]

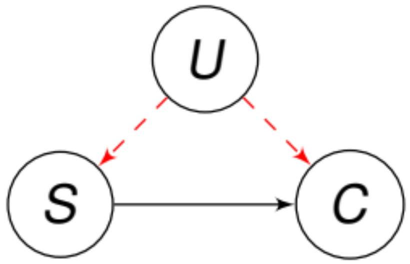
Markov Condition & Faithfulness Assumption

- The **Markov condition** is said to hold for a *DAG G =(V, E)* and a probability distribution *P(V)* if every variable *T* is statistically independent of its graphical non-descendants (the set of vertices for which there is no directed path from *T*) conditional on its graphical parents in *P*.
- Implication: $\forall X, Y \in V, \forall Z \subseteq V \setminus \{X, Y\} : (X \perp\!\!\!\perp_d Y|Z \implies X \perp\!\!\!\perp_p Y|Z)$.
- The **faithfulness assumption** states that the only conditional independencies to hold are those specified by the Markov condition, formally: $\forall X, Y \in V, \forall Z \subseteq V \setminus \{X, Y\} : (X \not\!\perp\!\!\!\perp_d Y|Z \implies X \not\!\perp\!\!\!\perp_p Y|Z)$.
- Implication: faithfulness is assumed to prove that the learned graph is correct.

Causal Sufficiency Assumption

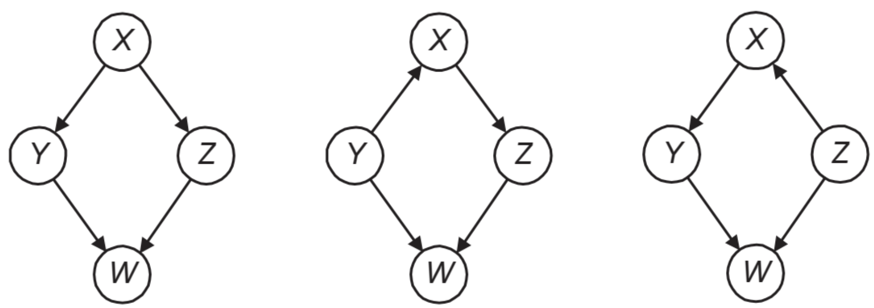
- there are no *unmeasured common causes* and no *unmeasured selection variables*.
- Example:
  - The simplest model:

  - Fisher's Smoking and Lung Cancer Model:



Markov Equivalence in Bayesian Networks

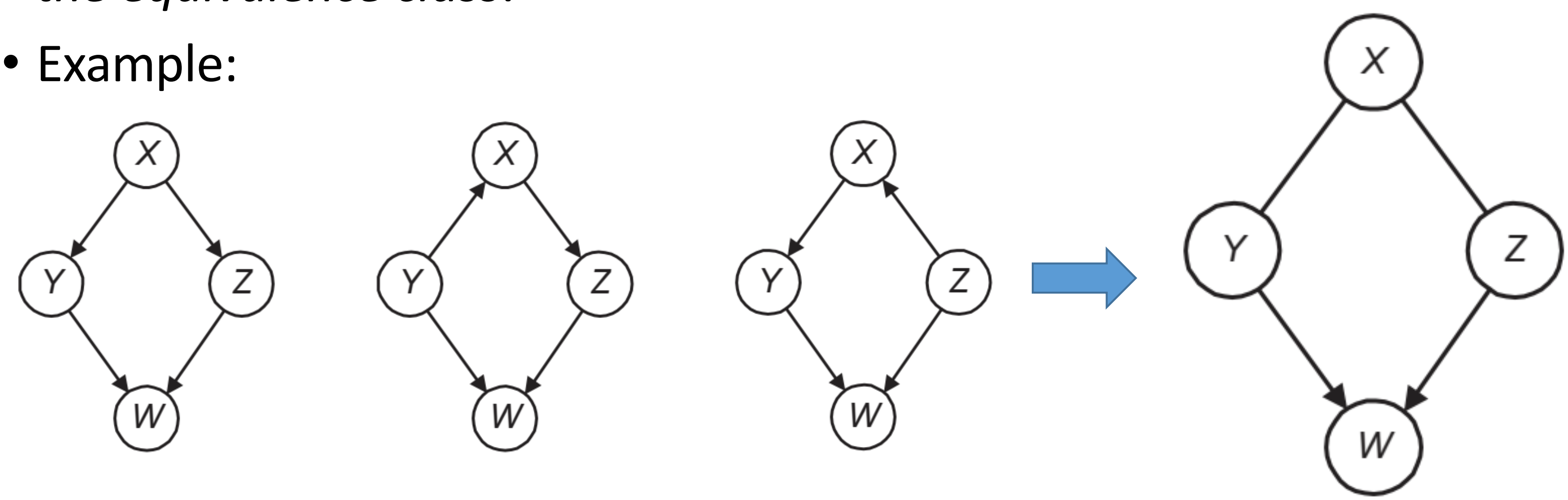
- Two DAGs are **Markov equivalent** if and only if, based on the Markov condition, they entail the *same conditional independencies*.
- Example (Quiz):



- Theorem. Two DAGs *G1* and *G2* are *Markov equivalent* if and only if they have the same links[skeleton] (edges without regard for direction) and the same set of unshielded colliders (v-structures).

Markov Equivalence in Bayesian Networks

- we define a **DAG pattern** for a Markov equivalence class to be the graph that has the *same links as the DAGs* in the equivalence class and has *oriented all and only the edges common to all of the DAGs in the equivalence class*.
- Example:



PC algorithm

Algorithm 1: The PC algorithm for learning DAGs

Input: A set V of nodes and a probability distribution p faithful to an unknown DAG G and an ordering order(V) on the variables.
Output: DAG pattern H.

Let H denote the complete undirected graph over V;
/* Skeleton Recovery */
for $i \leftarrow 0$ **to** $|V_H| - 2$ **do**
    while *possible* **do**
        Select any ordered pair of nodes u and v in H such that $u \in ad_H(v)$ and $|ad_H(u) \setminus v| \geq i$ using order(V);
        /* $ad_H(x) := \{y \in V | x \longrightarrow y, y \longrightarrow x,$ or $x \text{---} y\}$ */
        if *there exists* $S \subseteq (ad_H(u) \setminus v)$ *s.t.* $|S| = i$ *and* $u \perp\!\!\!\perp_p v | S$ *(i.e., u is independent of v given S in the probability distribution p)* **then**
            Set $S_{uv} = S_{vu} = S$;
            Remove the edge $u \text{---} v$ from H;
        end
    end
end
/* v-structure Recovery */
for *each separator* S_{uv} **do**
    if $u \text{---} w \text{---} v$ *appears in the skeleton and* w *is not in* S_{uv} **then**
        Determine a v-structure $u \longrightarrow w \longleftarrow v$;
    end
end
return H;

PC algorithm (Skeleton Recovery)

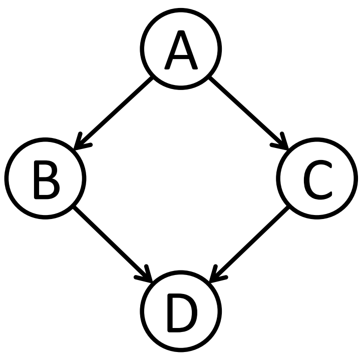
Naïve approach:

- construct an undirected graph *H* such that vertices $u - v$ if and only if no set S_{uv} can be found such that $u \perp_p v | S_{uv}$.
- To determine whether there is a set separating *u* and *v*, we might search all 2^{n-2} subsets of all *n* random variables excluding *u* and *v*.
- So, the complexity for investigating each possible edge in the skeleton is $O(2^n)$ and hence the complexity for constructing the skeleton is $O(n^2 2^n)$, where *n* is the number of vertices in the DAG.

for $i \leftarrow 0$ **to** $|V_H| - 2$ **do**
    while *possible* **do**
        Select any ordered pair of nodes u and v in H such that $u \in ad_H(v)$ and $|ad(u) \setminus v| \geq i$;
        /* $ad_H(x) := \{y \in V | x \rightarrow y, y \rightarrow x,$ or $x - y\}$ */
        if *there exists* $S \subseteq (ad_H(u) \setminus v)$ *s.t.* $|S| = i$ *and* $u \perp\!\!\!\perp_p v | S$ *(i.e., u is independent of v given S in the probability distribution p)* **then**
            Set $S_{uv} = S_{vu} = S$;
            Remove the edge $u - v$ from H;
        end
    end
end

PC algorithm (Skeleton Recovery)

- Example: alphabetical ordering



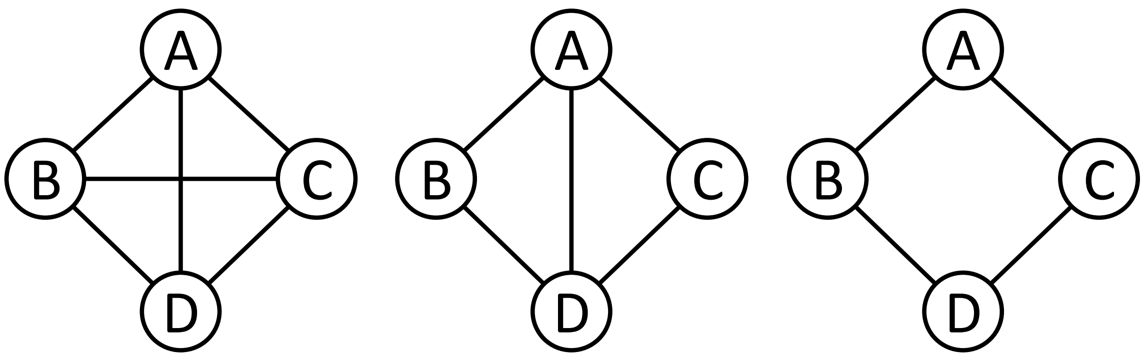
A1 | c

	A	B	C	D
1	c	b	a	d
2	-4.37815	3.250093	1.171418	-1.8429
3	-2.0754	1.744617	0.175751	-0.84926
4	1.252989	-2.76795	0.421092	1.448017
5	-1.54862	1.817924	1.299873	0.220929
6	4.193677	-4.55141	-2.17857	1.546651
7	1.296974	-0.3538	0.721286	0.483316
8	-2.39266	0.194322	0.750423	-0.4586
9	-2.32502	0.614471	0.683146	0.993996
10	1.472147	-1.71747	-1.47815	-0.96295

- Obtained conditional independence tests: $B \perp_p C | A$, $A \perp_p D | (B, C)$.

PC algorithm (Skeleton Recovery)

- $B \perp_p C|A$, $A \perp_p D|(B, C)$.



for $i \leftarrow 0$ **to** $|V_H| - 2$ **do**
  while *possible* **do**
    Select any ordered pair of nodes u and v in H such that $u \in ad_H(v)$ and $|ad(u) \setminus v| \geq i$;
    /* $ad_H(x) := \{y \in V | x \rightarrow y, y \rightarrow x, \text{ or } x - y\}$ */
    if *there exists* $S \subseteq (ad_H(u) \setminus v)$ *s.t.* $|S| = i$ *and* $u \perp\!\!\!\perp_p v | S$ *(i.e., u is independent of v given S in the probability distribution p)* **then**
      Set $S_{uv} = S_{vu} = S$;
      Remove the edge $u - v$ from H;
    end
  end
end

Start With $H = K_4$, after i = 0

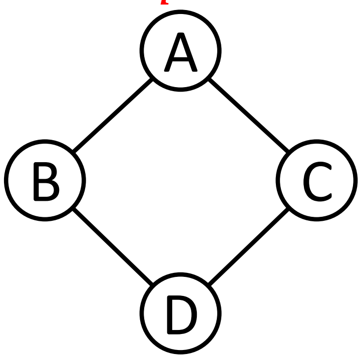
after i = 1 $B \perp_p C|A$

after i = 2 $A \perp_p D|(B, C)$

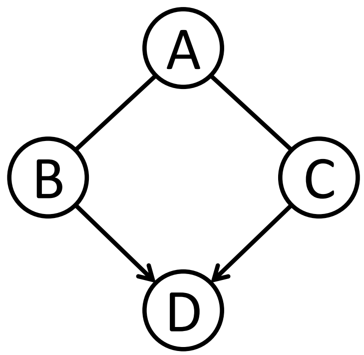
v-structure Recovery

- $B \perp_p C | A, A \perp_p D | (B, C).$

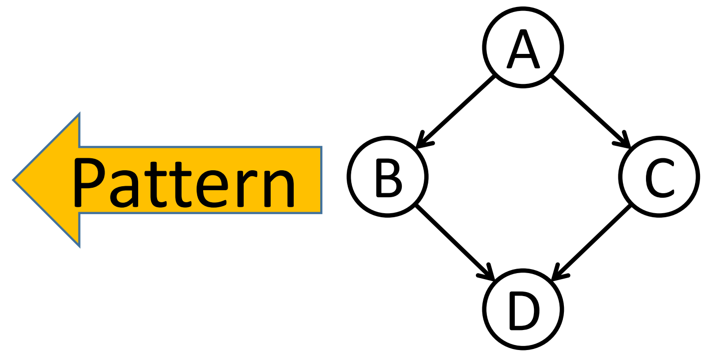
```
for each separator S_uv do
    if u — w — v appears in the skeleton and w is not in S_uv then
        Determine a v-structure u ⟶ w ⟵ v;
    end
end
return H;
```



after Skeleton recovery phase



$A \in S_{BC}$
$D \notin S_{BC}$



Complexity

- The complexity of the algorithm for a graph *G* is bounded by the largest degree in *G*. Let *k* be the maximal degree of any vertex and let *n* be the number of vertices. Then in the worst case the number of **c**onditional **i**ndependence (CI) tests required by the algorithm is bounded by:

$$2\binom{n}{2}\sum_{i=0}^{k}\binom{n-2}{i} \leq \frac{n^2(n-2)^k}{(k-1)!}$$

- PC algorithm has a worst-case running time of $O(n^{k+2})$.

Discussion

1) PC and Order-dependence problem: the output can depend on the order in which the variables are given.
   - ➢Stable (order-independent) PC algorithm
2) Scalability problem with PC algorithm:
   - ➢Parallel PC algorithm
3) Can we relax the causal sufficiency assumption?
   - ➢Yes, Fast Causal Inference (FCI) algorithm
4) Can we weaken the faithfulness assumption?
   - ➢Yes, Greedy Equivalence Search (GES) an inclusion optimal structure learning algorithm