Lecture 4: Learning DAG models from Data

As we saw yesterday $\vec{K}_n$ is an I-map for any distribution $\mathbb{P}$.
So we want a minimal I-map.

A goal of the course is to develop knowledge of algorithms for learning representations from data.

Recall $\mathbb{P}$ is <u>faithful</u> to DAG G if $\mathcal{CI}(\mathbb{P}) = \mathcal{CI}(G)$

Most algorithms assume faithfulness, "not such a strong assumption"

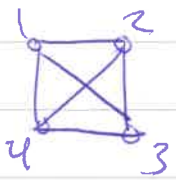
<u>Lemma:</u> The set of unfaithful distributions has Lebesgue measure zero

<u>Goal:</u> Assume $\mathbb{P}$ faithful to true causal structure G and recover G from data $\mathbb{D}$.

The PC-algorithm

Assume $CI(\mathbb{P}) = \{1 \perp\!\!\!\perp 2,\ 1 \perp\!\!\!\perp 4 \mid 3,\ 1 \perp\!\!\!\perp 4 \mid \{2,3\},\ 2 \perp\!\!\!\perp 4 \mid 3,\ 2 \perp\!\!\!\perp 4 \mid \{1,3\}\}$

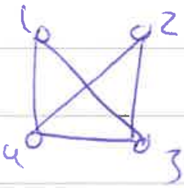
<u>Step 1</u> Learn skeleton



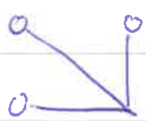
(0) start with K_m, complete graph on $[m]$

(1) Eliminate edges ij if $i \perp\!\!\!\perp j \mid \emptyset$

(2) For each ij still an edge & h adjacent to either i or j. (i.e. $h \in N(i) \cup N(j)$) eliminate ij if $i \perp\!\!\!\perp j \mid h$

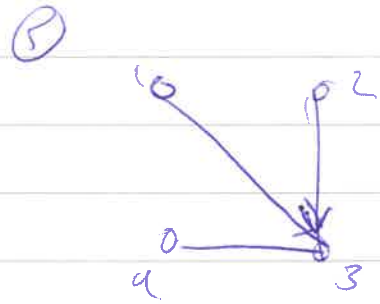
①


(3) Repeat (2) with larger sets. $\{h_1, h_2\} \subseteq N(i)$ or $\{h_1, h_2\} \subseteq N(j)$

②


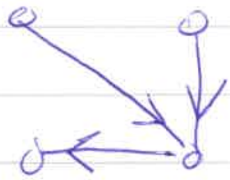
(4) Repeat (3) with larger neighborhoods until we reached $|N(i)|$ $\forall i$

Step 2: Learn v-structures



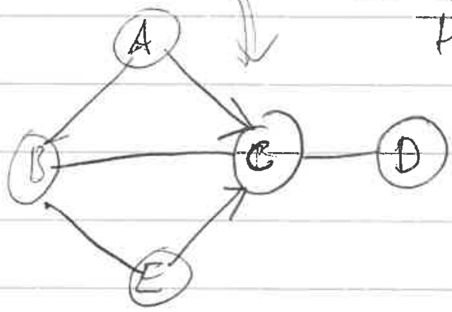
5 For each ~~triple~~ path <i,j,k>
i,k not adjacent orient path as $i \to j \leftarrow k$
if j was not in the conditioning set C
in previous step

6 For each triple $i \to j - k$ orient $j \to k$
i,k not adjacent
to avoid v-structure
Use also 4 configurations
from def. of strongly protected.



many more

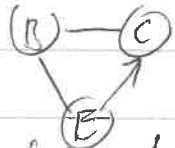
Example $CI = \{A \perp\!\!\!\perp E | B,\ \underbrace{D \perp\!\!\!\perp E | C,\ D \perp\!\!\!\perp A | C,\ D \perp\!\!\!\perp B | C}_{\text{these give D only connected to C}}\}$



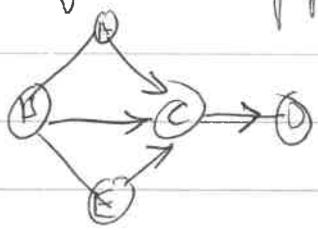
Must get:

C → D

Must counterorient at least one edge in the triangles



By fourth config on strongly protected it becomes



Advantages

- poly time for sparse graphs $O(n^{\Delta+1})$
- Recovers MEC if P faithful to some DAG
- software ∃

Disadv.

- When it is wrong can be very wrong, not stable.

Theorem 6 Suppose the data generating distribution P is faithful to G and we are using a test for conditional independence that perfectly answers any query as $|D| \to \infty$. Then any acyclic orientation of the output of the PC algorithm that introduces no new v-structures will be ME to G.

Proof: By assumption, whenever we test $X_A \perp\!\!\!\perp X_B \mid X_C$, as $|D| \to \infty$, we will learn that the statement holds precisely when it holds in P.

So need only check that we test the right statements needed to recover a DAG ME to G. So need to check that we learn the correct ~~skeleton~~ skeleton and v-structures.

problem 3 last hour yesterday

Skeleton: recall i and j adjacent in G $\iff$ i, j d-separated given either $pa_G(i)$ or $pa_G(j)$. In step 1, eventually the conditioning set is large enough that we test (wlog) $i \perp\!\!\!\perp j \mid pa_G(i)$, which holds in P, by faithfulness. $\Rightarrow$ learn correct skeleton.

v-structures: Recall $i - j - k$ a v-structure $\iff$ ① $\exists\, C \subset [m] \setminus \{i,k\}$ d-separating i and k in G such that $j \notin C$. $\iff$ ② all sets $C \subset [m] \setminus \{i,k\}$ containing j do not d-separate i and k.

Let $i \perp\!\!\!\perp k \mid C$ be statement from step 1 that lead to removal of edge $i - k$. [If $j \notin C$, by faithfulness of P, $i \to j \leftarrow k$ by ①. If $j \in C$, by ② and faithfulness, no v-structure in G.]

Similarly, by faithfulness, the orientation in step ⑦ will produce no additional v-structures.

So the output has exactly the skeleton & v-structures of G.

$\Rightarrow$ Any such orientation of it will be ME to G by VP. ∎

Assuming faithfulness and a consistent test for conditional independence, the PC algorithm will learn the true causal structure (up to Markov equivalence). However, we typically use hypothesis tests for testing CI, which are prone to error... causes accuracy problems.

PC algorithm is a constraint-based ~~alg~~ algorithm, relying on CI tests.

To avoid problems with accuracy we can instead use greedy (score-based) algorithms...

Greedy Equivalence Search (GES)

- Assign each DAG a score, such as the Bayesian Information Criterion (BIC):

$$\mathrm{BIC}(G, \mathbb{D}) := \underbrace{\log P(\mathbb{D} \mid \hat{\Theta}, G)}_{\text{log-likelihood of } \mathbb{D} \text{ given } G} - \underbrace{\frac{d}{2}\log(|\mathbb{D}|)}_{\text{penalization term based on \# of free parameters } d \text{ in } G}$$

Lemma ④ BIC is score equivalent; i.e. if G and $\mathcal{H}$ are Markov equivalent then $\mathrm{BIC}(G, \mathbb{D}) = \mathrm{BIC}(\mathcal{H}, \mathbb{D})$ for all data sets

- GES relies on the following generalization of the characterization of Markov equivalence proven in problem set 3.

- Write $G \leq \mathcal{H}$ if $CI(G) \supset CI(\mathcal{H})$ where $CI(G) := \{ A \perp\!\!\!\perp B \mid C \ : \ A, B \text{ d-separated given } C \text{ in } G \}$.

Theorem ⑦ (Chickering, 2002) Suppose $G \leq \mathcal{H}$. Then $\exists$ a sequence of edge reversals and edge deletions such that

(1) all edges reversed are covered; $i \to j$ where $pa_G(j) = pa_G(i) \cup \{i\}$.

(2) After each edge reversal or addition get a DAG $\hat{G}$ s.t. $\hat{G} \leq \mathcal{H}$

(3) After all edge reversals and additions $\hat{G} = \mathcal{H}$.

GES: Let $[G]$ denote MEC of G.

① Start at empty DAG $G := ([m], \emptyset)$.

② $E^+([G]) := \{$MECs of DAGs produced by adding single edge to (a DAG in) $[G]\}$
Pick $[\mathcal{H}] \in E^+([G])$ with highest BIC and set $G := \mathcal{H}$.

③ Repeat ② until no higher scoring MEC found.

④ $E^-([G]) := \{$MECs produced by removing single edge from DAG in $[G]\}$
Pick $[\mathcal{H}] \in E^-([G])$ with highest BIC and set $G := \mathcal{H}$.

⑤ Repeat step ④ until no higher scoring MEC found,

⑥ Return $[G]$.

Theorem ⑧ (Chickering, 2002) If P faithful to G, as $|\mathbb{D}| \to \infty$, the output of GES will be the MEC of G.

- While GES is more accurate, PC has polynomial time performance guarantees (see HW).

- Hybrid Algorithms: Combine score-based and constraint-based approaches to strike balance.