Optimization and Machine Learning

Lecture 4: IP models to learn acyclic graphs

Sanjeeb Dash
IBM Research
(co-lecturer Parikshit Ram)

JPOC 13 Summer School, June 26-28, 2023
University of Clermont-Ferrand

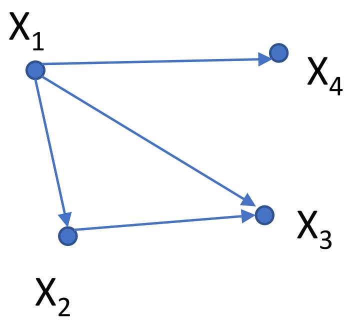
Lecture 4 Outline

- Brief review of IP to learn rules/knowledge graphs
- Bayesian Network structure learning
- Integer Programming Formulation to find optimal scores
- Latent variables and IP methods
- Numerical Experiments

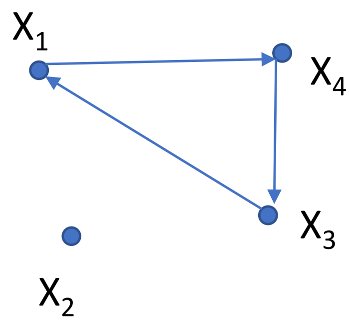
Bayesian Network Structure Learning

Bayesian Network: Directed acyclic graph (DAG) representing conditional probability relationships between variables

dag



directed cycle



$$P(X_1, X_2, X_3, X_4) = P(X_4|X_1)P(X_3|X_1, X_2)P(X_2|X_1)P(X_1)$$

BNSL Problem - Learn DAG from data:

DP methods: Koivisto, Sood ’04, Silander, Myllymäki ’06

A* search: Yuan, Malone ’13

Branch-and-bound: Campos, Ji ’11

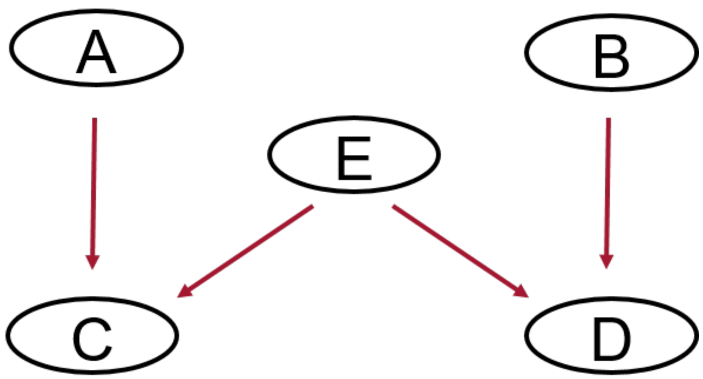
IP based solver GOBNILP: Bartlett, Cussens ’13, ’17

GOBNILP is a state-of-the-art method: Malone et. al. ’17

Causal Graphs/Causal BN

- Graphical Models where directed edges represent causal relationships
- DAG encodes *structural equations*

Directed Acyclic Graph

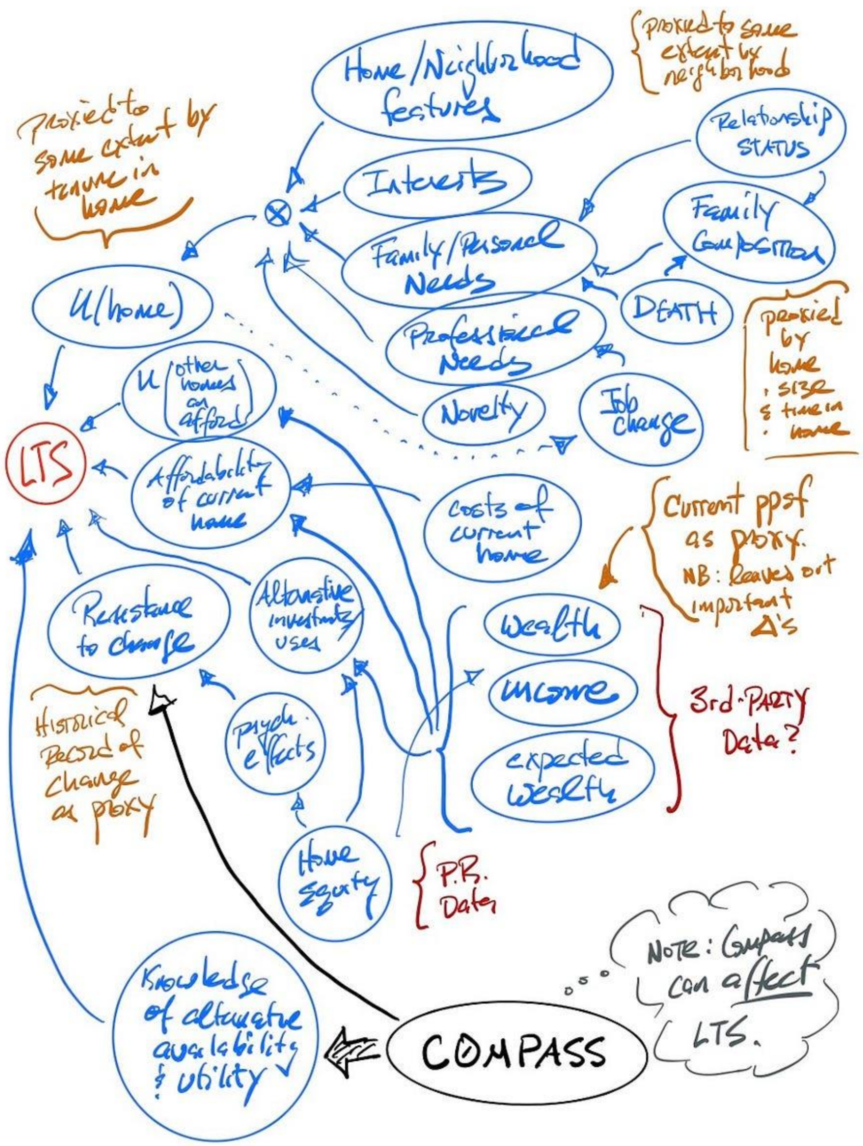


(Linear) Structural equations

$$\Leftrightarrow \begin{cases} x_A = \epsilon_A \\ x_B = \epsilon_B \\ x_C = b_{CA}x_A + b_{CE}x_E + \epsilon_C \\ x_D = b_{DB}x_B + b_{DE}x_E + \epsilon_D \\ x_E = \epsilon_E \end{cases}$$

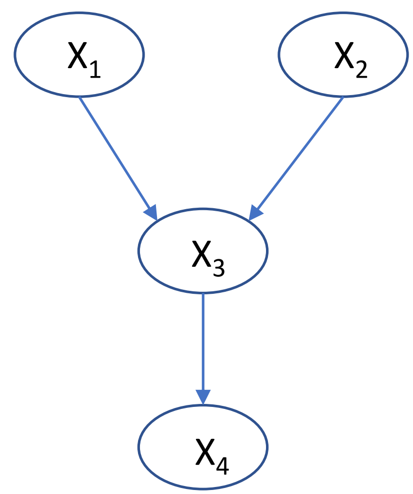
In a BN, $X \to Y \to Z$ and $X \leftarrow Y \leftarrow Z$ are indistinguishable.

Creating causal graphs



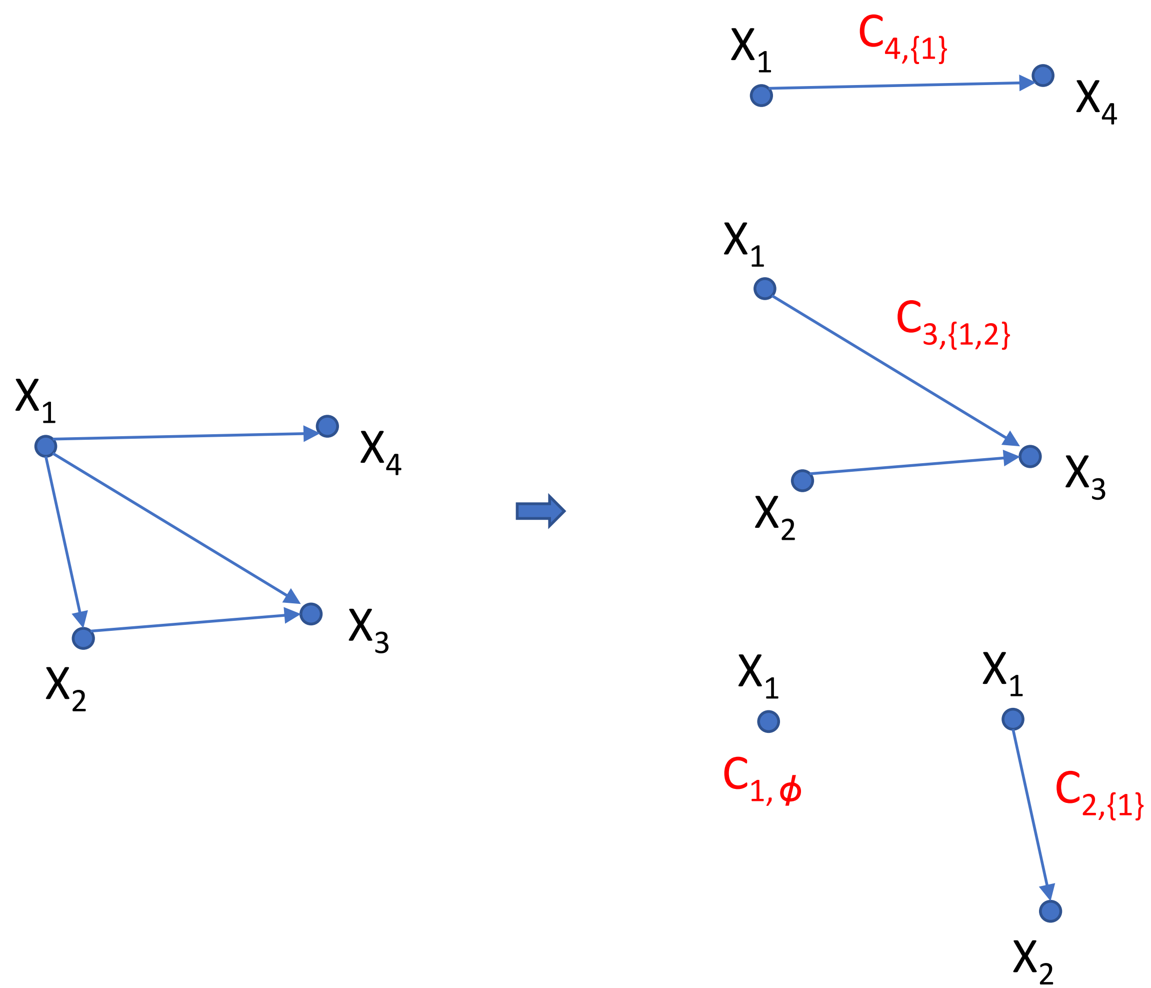
Foster, Ipeirotis 2022

X_1	X_2	X_3	X_4
1	0	1	0
0	1	1	1
1	1	1	0
0	1	1	1
0	0	0	1



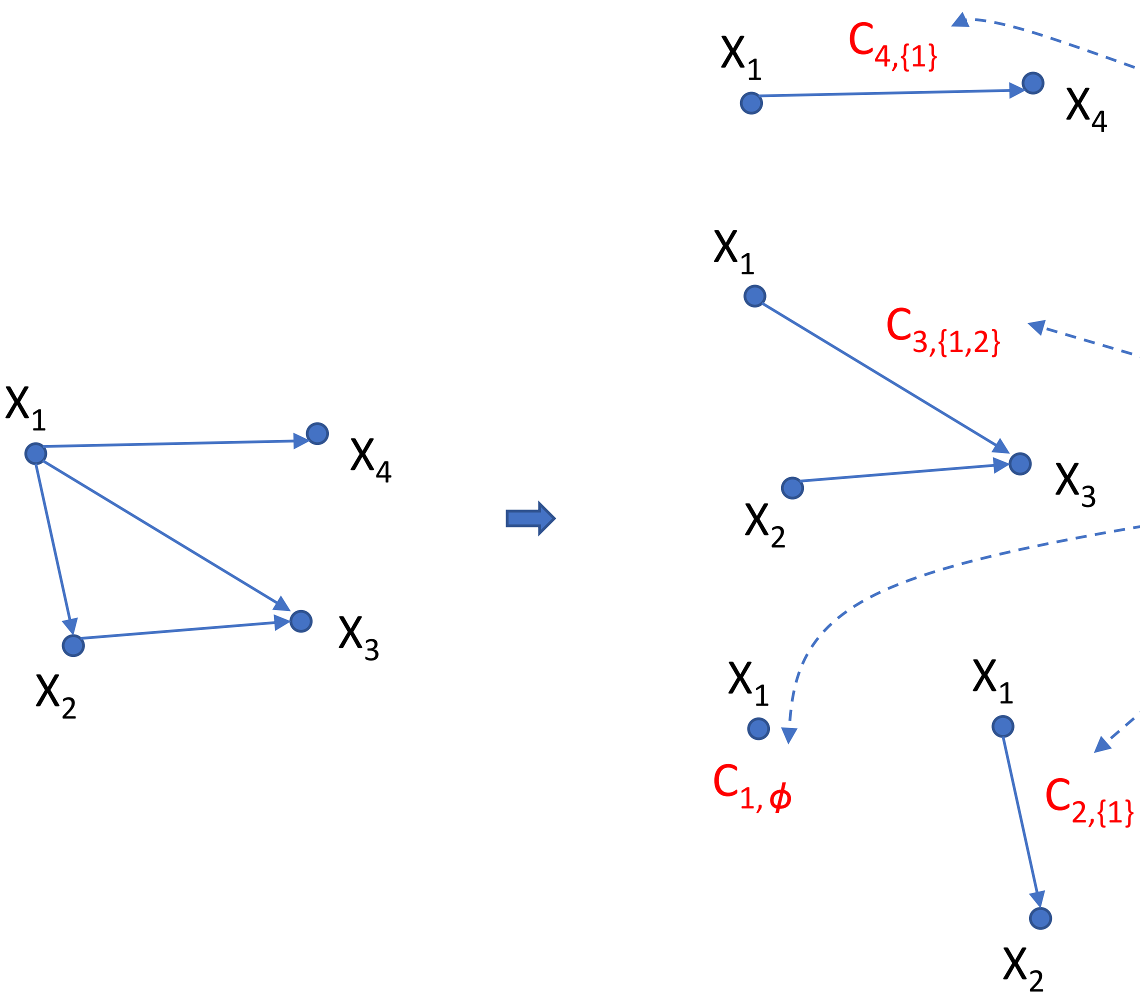
Score decompositions for BNSL

Score of DAG is sum of scores of “in-stars” (inward directed star)



Score calculation

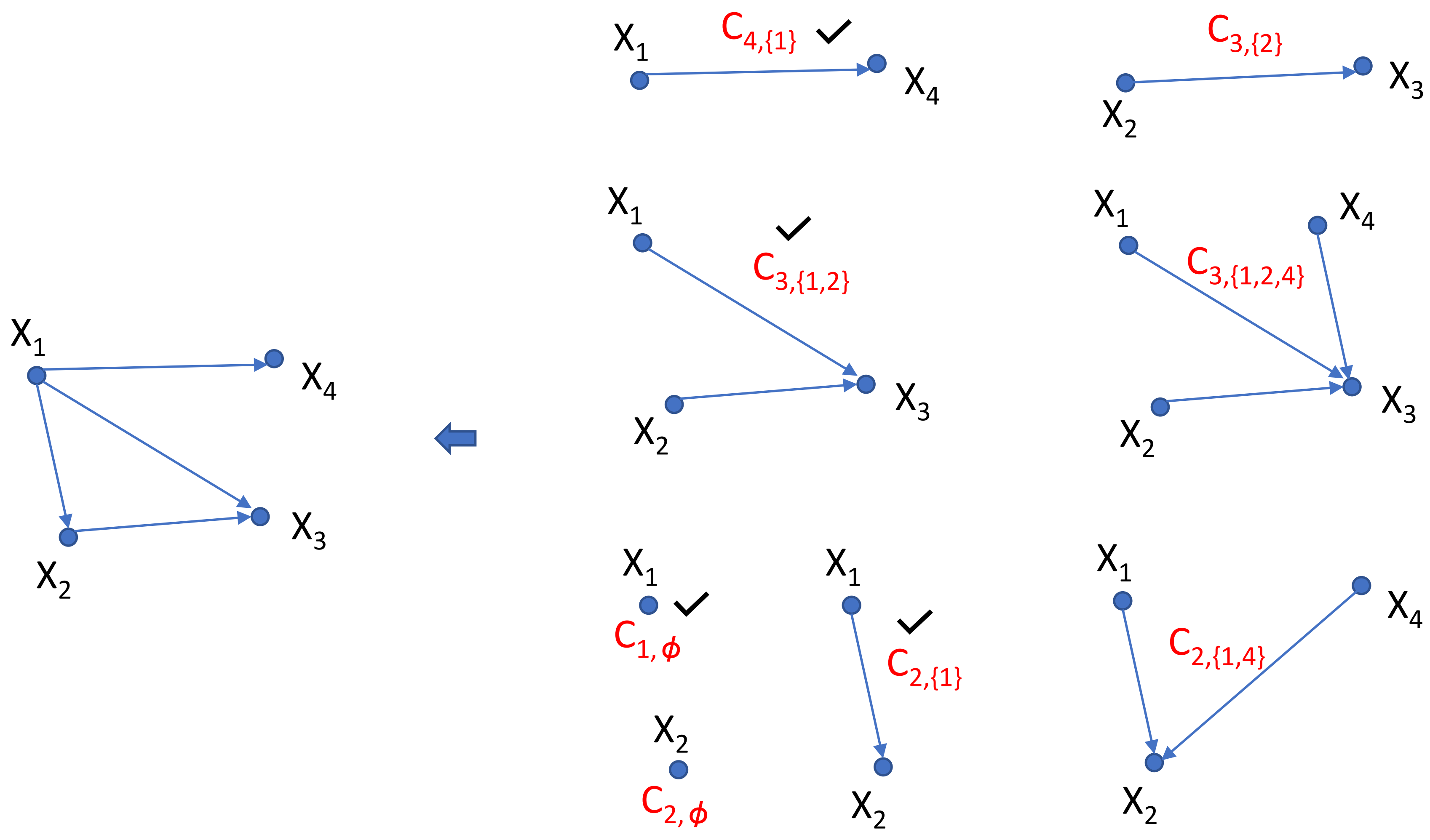
Score of each“in-star” is calculated from data



X_1	X_2	X_3	X_4
1	0	1	1
0	1	0	1
1	0	1	1
0	1	0	1
1	0	1	1

MIP for score based approach

MIP has one variable per in-star, equations choosing one in-star per node, and *cluster inequalities* preventing cycles.



Opt. formulations

Notation: Node set - $V = \{1, \ldots, n\}$, $P(i)$ = set of parent sets of i.

MIP (parent set variables):

$$\max \sum_{i \in V} \sum_{P \in P(i)} c_{i,P} z_{i,P}$$

$$\sum_{P \in P(i)} z_{i,P} = 1, \ \ \forall i \in V$$

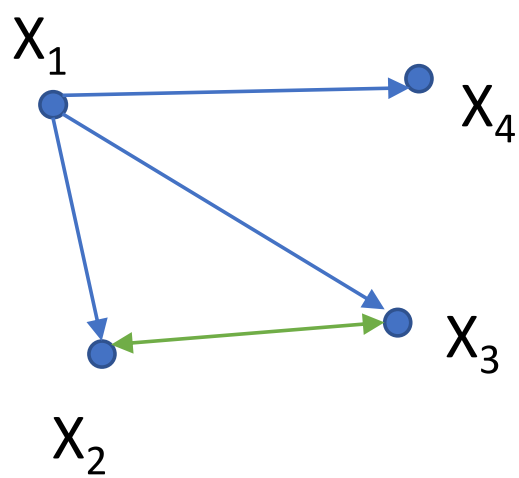
$$\sum_{i \in S, P \cap S = \emptyset} z_{i,P} \geq 1, \ \ \forall S \subseteq V \ \ *$$

$$z_{i,P} \in \{0, 1\}$$

Jaakkola, Sontag, Globerson, Meila ’10: cluster constraints(*)
Bartlett, Cussens ’13, 17: IP + software (GOBNILP)
Grotschel, Junger, Reinelt ’85: Acyclic subgraph polytope

Latent Variables

Goal: Learn causal network structures in the presence of latent vars.

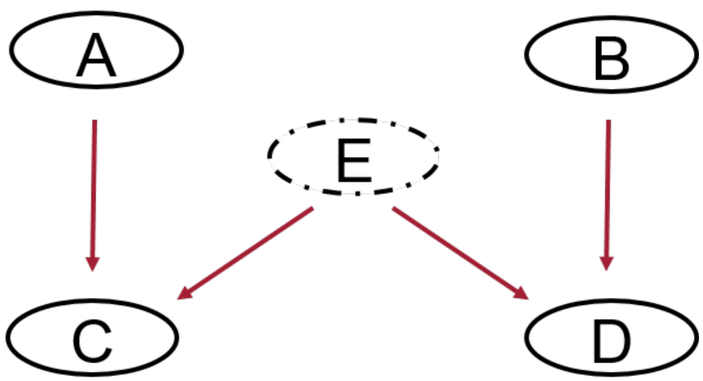


We use **ancestral acyclic directed mixed graphs** (with directed + bidirected edges) as models of data with latent confounders.

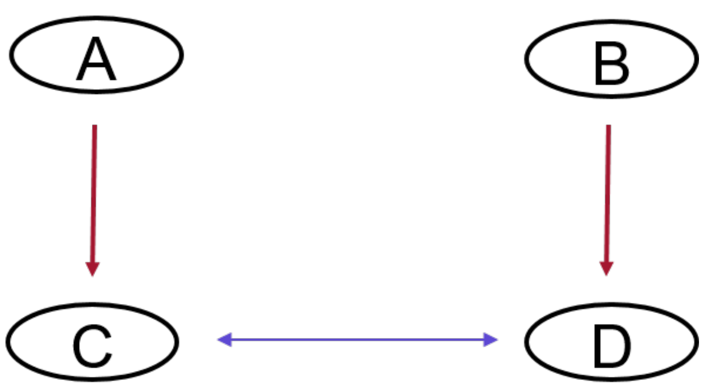
Chen, Dash, Gao '21: MIP formulation & first exact score-based method to find optimal AADMG for continuous Gaussian variables.

Ancestral graphs (AGs)

- DAGs are not closed under marginalization!



Ancestral graphs (Richardson and Spirtes '02)

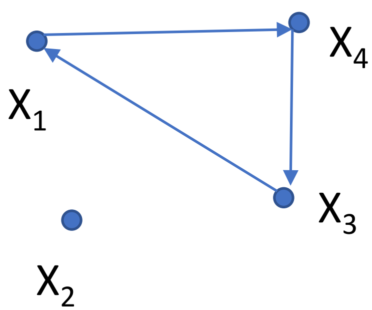


- Include all DAGs and are closed under marginalization
- Properties:

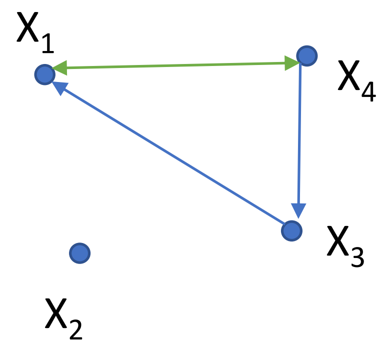
No directed cycles
$(a \to b \to \ldots \to a)$
No almost directed cycles
$(a \leftrightarrow b \to c \to \ldots \to a)$

Forbidden structures

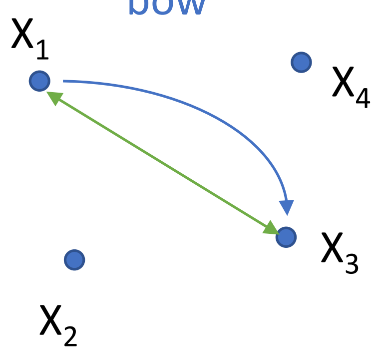
directed cycle



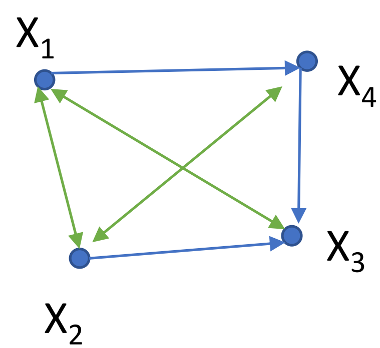
almost directed cycle



bow



rooted arborescence +
bidirected component



Learning methods

Constraint-based methods:

► Apply conditional independence test on the data to infer the graph structure: FCI (Sprites et al., ’00), cFCI (Ramsey et al., ’12)

Score-based methods:

► Optimize a scoring criterion that measures the likelihood of the graph: GSMAG (Triantafillou and Tsamardinos, ’16)

Hybrid methods:

► Use both a scoring criterion and conditional independence tests: M^{3}HC (Tsirlis et al., ’18), SPo (Bernstein et al., ’20), CCHM (Chobtham and Constantinou, ’20)

Current score-based and hybrid methods are all greedy or local search algorithms!

Scoring a DMG

- The BIC score (Schwarz ’78) for graph $\mathcal{G}$ is given by

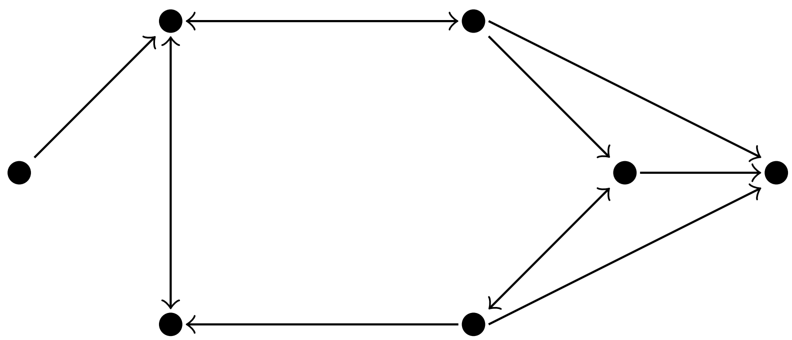
$$\text{BIC}_{\mathcal{G}} = 2\ln(l_{\mathcal{G}}(\hat{\Sigma})) - \ln(N)(2|V| + |E|)$$

- The maximum log-likelihood $\ln(l_{\mathcal{G}}(\hat{\Sigma}))$ can be decomposed by c-components in $\mathcal{G}$ (Nowzohour et al., ’17)

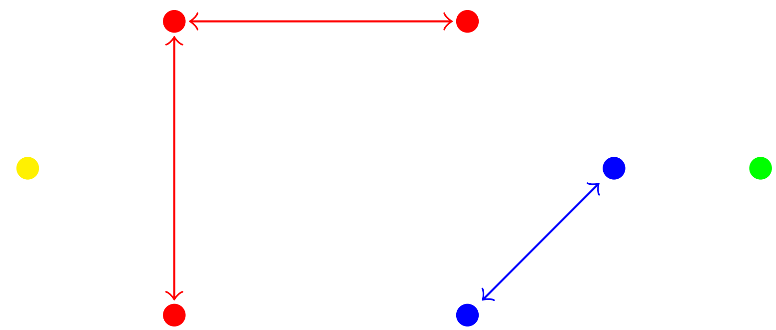
$$\ln(l_{\mathcal{G}}(\hat{\Sigma})) = -\frac{N}{2}\sum_{D\in\mathcal{D}}\Big[|D|\ln(2\pi) + \log(\frac{|\hat{\Sigma}_{\mathcal{G}_D}|}{\prod_{j\in pa_{\mathcal{G}}(D)}\hat{\sigma}^2_{Dj}}) + \frac{N-1}{N} tr(\hat{\Sigma}^{-1}_{\mathcal{G}_D} S_D - |pa_{\mathcal{G}}(D)\setminus D|)\Big]$$

district = component defined by bidirected edges
c-component = district + in-edges per node in district

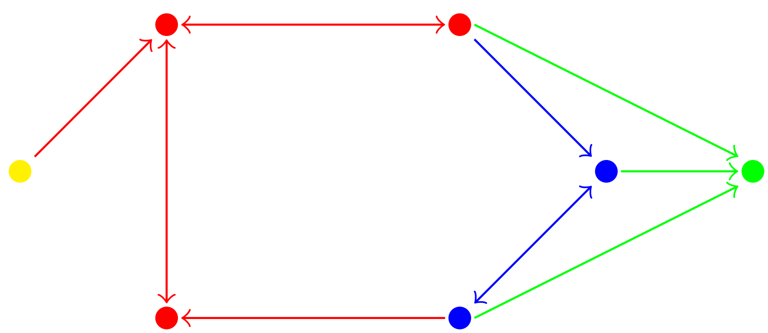
Decomposition into c-components



Ancestral ADMG



Districts

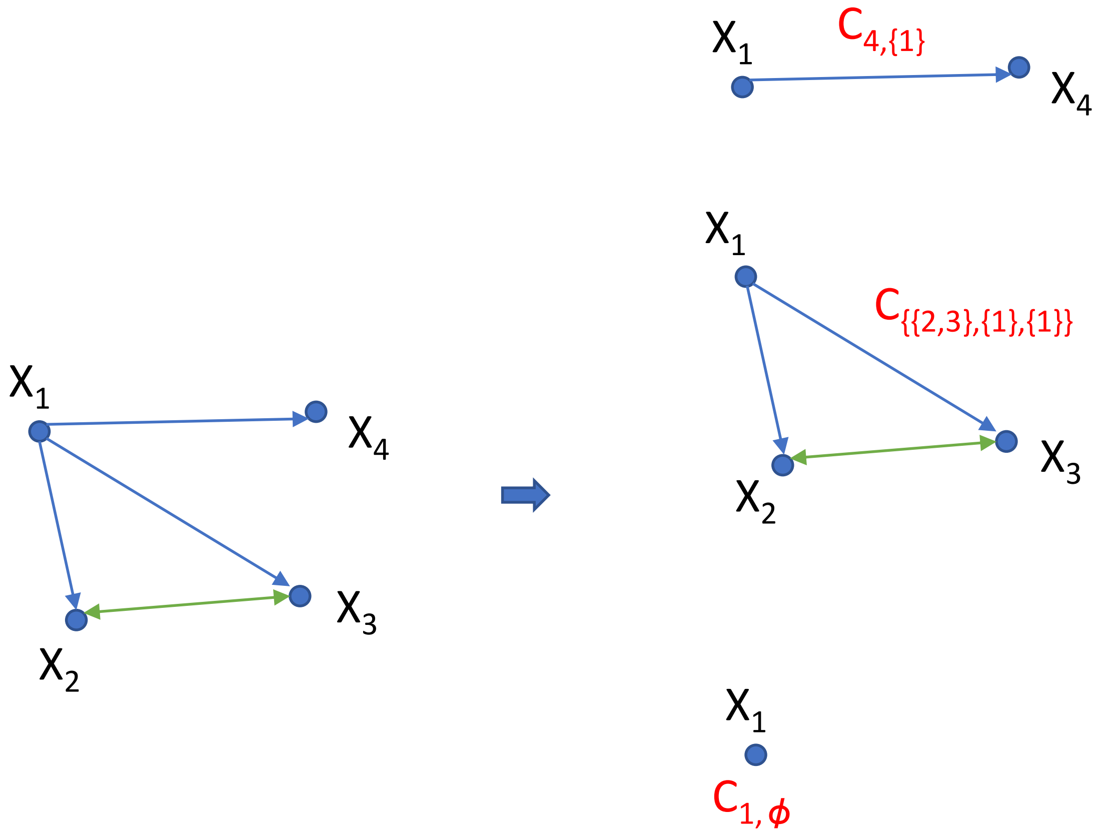


c-components

► We obtain a (BIC) score-maximizing ancestral ADMG for a set of continuous variables that follow a multivariate Gaussian distribution.

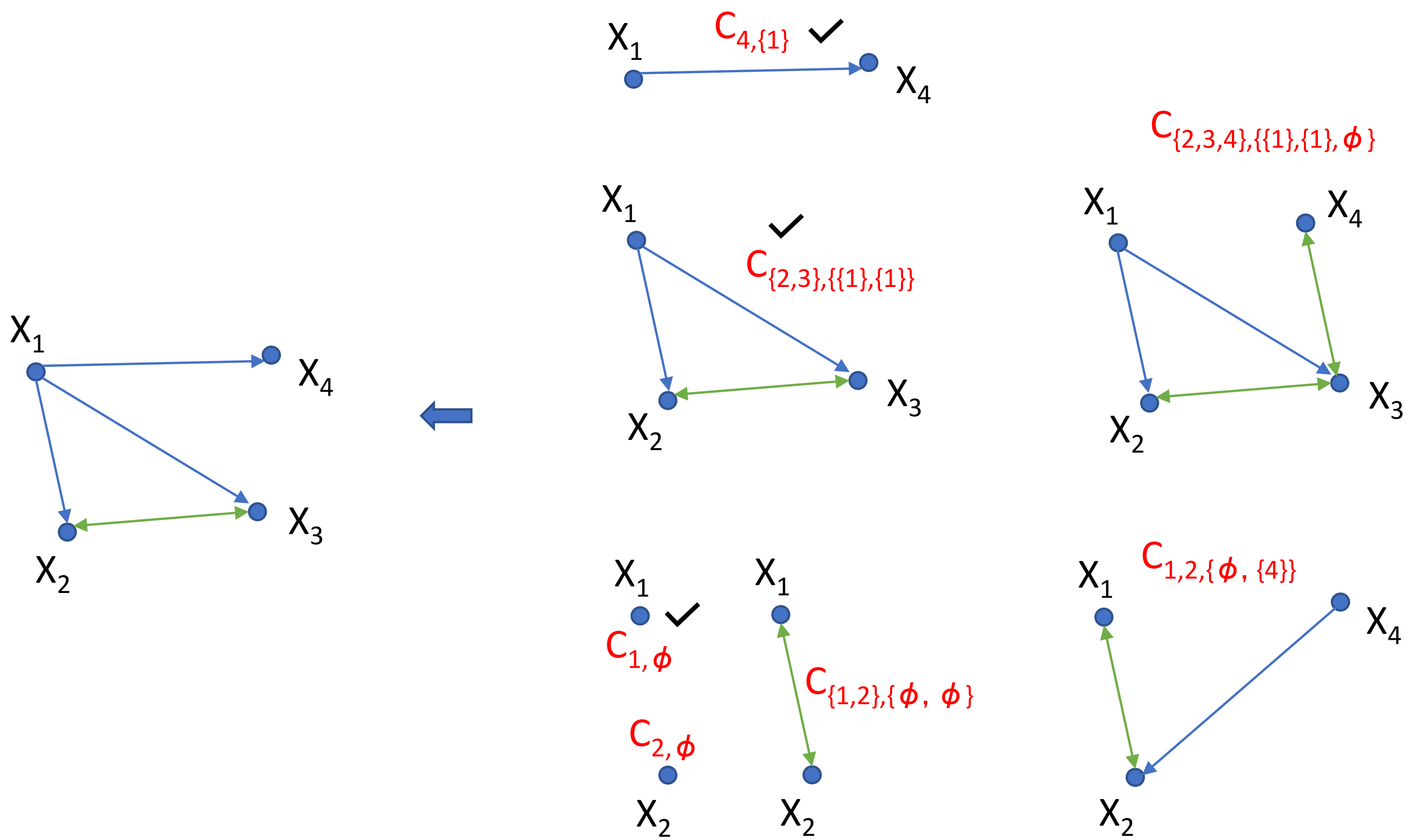
Score decomposition for AADMG

Score of AADMG is sum of scores of c-components



Approach

Our work: Learn an AADMG with maximum score from c-components



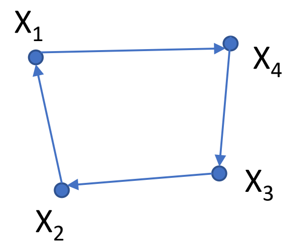
MIP formulation

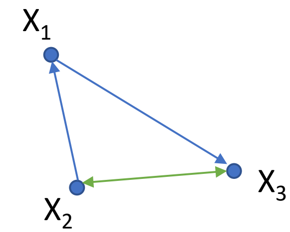
Let $\mathcal{C}$ be set of all c-components, and let $D(C)$ be the district of a c-component C.

MIP to find optimal AADMG:

$$
\begin{aligned}
\max \quad & \sum_{c \in \mathcal{C}} s_C z_C \\
& \sum_{C : i \in D(C)} z_C = 1, \ \forall i \in V \\
& G(z) \text{ has no directed and almost directed cycles} \\
& z_C \in \{0, 1\}
\end{aligned}
$$

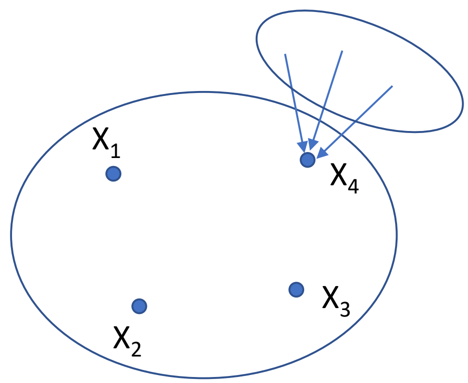
Cutting planes to avoid cycles

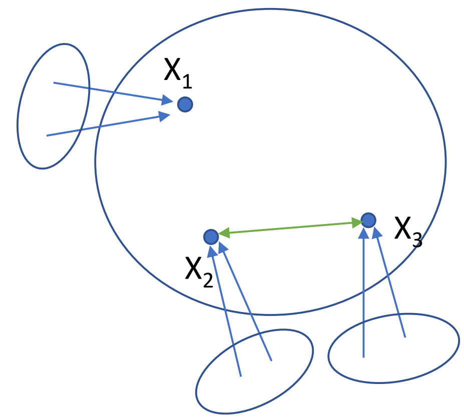




Cluster Inequalities:

$$\sum_{i \in S, P \cap S = \emptyset} z_{i,P} \geq 1, \;\; \forall S \subseteq V$$





Biclusterinequalities: $(w_{i,j} = \sum_{C: i \leftrightarrow j \in D(C)} z_C)$

$$\sum_{v \in S \setminus \{i,j\}} \sum_{P: P \cap S = \emptyset} z_{v,P} + \sum_{P^1: P^1 \cap S = \emptyset} \sum_{P^2: P^2 \cap S = \emptyset} z_{i,j,P^1,P^2} \geq w_{i,j}$$

Cutting planes generation

► Karger's ('93) random contraction algorithm for min-cut problems: Randomly contract edge ij with probability $\propto$ edge weight

► *Separation heuristic* for cluster inequalities:
- Let $\mu^k(S)$ denote the LHS of the cluster inequality at iteration k and

$$w_{ij}^k = \mu^k(\{i\}) + \mu^k(\{j\}) - \mu^k(\{i,j\}),\ \forall i,j$$

- At iteration k, randomly contract edge ij with probability $\propto w_{ij}^k$
- Remove nodes i and j, create a pseudo-node i' and replace all occurrences of i and j in the original graph by the pseudo-node
- Repeat until $\mu^k(\{i\}) < 1$ for some $i \Rightarrow$ a violated cluster inequality

► Similar separation heuristic for bi-cluster inequalities

Numerical Experiments

- Test set 1:
  1. Randomly generated DAGs with 20 nodes
  2. $l =$ 2,4,6 variables set to be latent
  3. $d =$ remaining observed variables
  4. A sample of $N = 1000/10,000$ realizations of observed variables per instance

- Candidate c-components:
  1. Single-node districts with up to three parents
  2. Two-node districts with up to one parent each node

- Compared methods:
  1. AGIP: our IP model
  2. DAGIP: our IP model with only single-node districts
  3. M^{3}HC: a greedy hybrid method by Tsirlis et al. (2018)
  4. FCI: an exact constraint-based method by Sprites et al. (2000)
  5. cFCI: an exact constraint-based method by Ramsey et al. (2012)

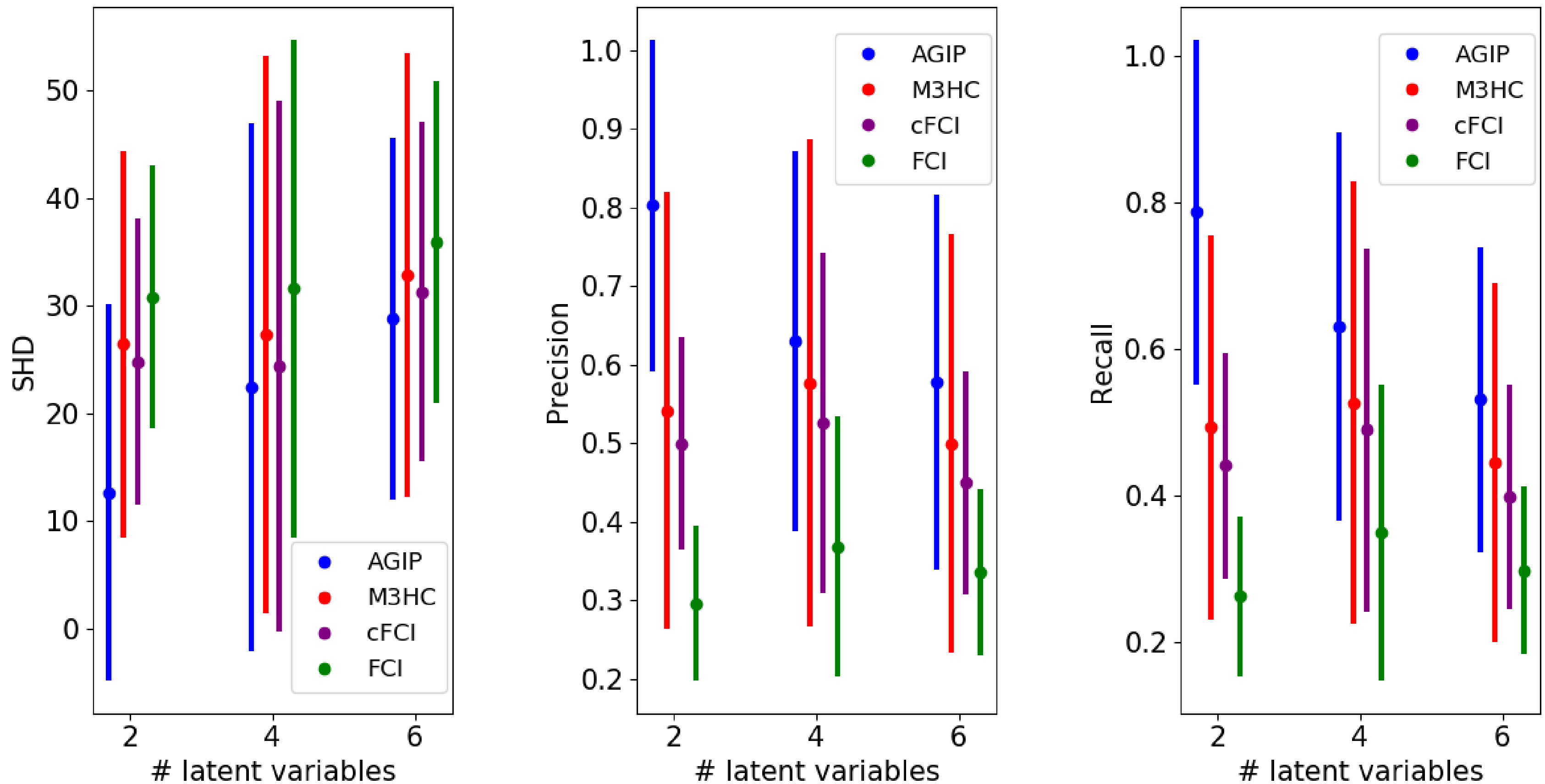
Quality of formulation

20-node graphs; d = number of observed nodes, l = number of latent variables (removed from graph), N = number of samples.

(d, l, N)	Avg # bin vars before pruning	Avg # bin vars after pruning	Avg pruning time (s)	Avg root gap (%)	Avg soln. time (s)
$(18, 2, 1000)$	59229	4116	19.1	0.65	60.4
$(16, 4, 1000)$	39816	3590	13.6	0.43	41.0
$(14, 6, 1000)$	20671	1788	3.9	0.54	8.9
$(18, 2, 10000)$	59229	9038	33.0	0.67	323.2
$(16, 4, 10000)$	39816	7378	21.4	0.53	215.4
$(14, 6, 10000)$	20671	3786	6.4	0.56	47.2

Results for varying number of latent vars.

$d = 18$, $l = 2, 4, 6$, $N = 10,000$,



Current work

- Find optimal bow-free/arid graphs (supersets of AADMGs) using MIP

Use BSNL formulation, but extra variables for c-components with > 1 node districts and no bows

MIP (parent set variables):

$$\max \sum_{i \in V} \sum_{P \in P(i)} c_{i,P} z_{i,P}$$

$$\sum_{P \in P(i)} z_{i,P} = 1, \ \forall i \in V$$

$$\sum_{i \in S, P \cap S = \emptyset} z_{i,P} \geq 1, \ \forall S \subseteq V \ *$$

$$z_{i,P} \in \{0, 1\}$$

Sparse instances

Dataset	Ground Truth	AADMG	Bow-free	Bhattacharya
0	-17741.6	**-17741.6**	**-17741.6**	-17765.1
1	-17508.5	**-17508.5**	**-17508.5**	-17511.9
2	-17872.5	*-17871.2*	*-17871.2*	**-17872.5**
3	-19055.6	-19093.6	**-19055.6**	-19123.7
4	-17888.1	-17884.1	-17881.6	-17908.4
5	-18584.9	-18595.5	**-18584.9**	-18625.4
6	-17791.2	-17790.1	-17789.5	-17795.6
7	-18964.8	-19010.8	-18964.8	-20438.8
8	-17562.1	-17562.1	-17562.1	-17565.6
9	-17627.9	-17655.9	-17627.9	-17681.6

Scores for sparse randomly generated datasets

Method	Precision			Recall		
	skeleton	dir.	bidir.	skeleton	dir	bdir
AADMG	0.906	0.711	0.450	0.950	0.818	0.283
Bow-free	0.969	0.812	0.633	0.975	0.873	0.517
Bhattacharya	0.830	0.749	0.179	0.949	0.774	0.383

Average results

Medium density instances

Dataset	Ground Truth	AADMG	Bow-free	Bhattacharya	LP-heuristic
0	-19057.4	-19169.2	-19117.4	-19071.4	-19061.3
1	-19802.3	-20082.1	-19916.3	-19830.9	-19825.3
2	-20606.4	-21074.8	-20857.5	-20613.9	-20623.2
3	-21178.7	-21332.9	-21267.9	-21207.7	-21190.7
4	-20865.8	-20993.5	-20962.1	-20876.5	-20870.1
5	-18846.5	-19031.6	-18936.4	-18848.3	-18855.4
6	-21268.7	-21405.1	-21347.0	-21716.6	-21288.2
7	-18906.2	-18924.9	-18921.7	-18927.6	-18908.4
8	-22152.7	-22517.5	-22320.3	-22226.1	-22189.1
9	-21059.0	-21118.6	-21100.4	-21110.3	-21070.5

Method	Precision			Recall		
	skeleton	dir.	bidir.	skeleton	dir	bdir
AADMG	0.840	0.442	0.100	0.693	0.488	0.050
Bow-free	0.837	0.336	0.083	0.732	0.383	0.034
Bhattacharya	0.799	0.641	0.388	0.946	0.783	0.398
LP-heuristic	0.812	0.424	0.367	0.858	0.589	0.074

Open questions

- How does one deal with the exponentially many variables
- Find valid inequalities for bounded indegree acyclic graphs

Cussens, Jarvisalo, Korhonen, Bartlett '17: detailed study of associated polytopes

References

1. R. Chen, S. Dash, T. Gao, Integer programming for causal structure learning in the presence of latent variables. ICML 2021, PMLR 139:1550-1560.
2. J. Cussens, M. Jarvisalo, J. H. Korhonen, M. Bartlett, Bayesian Network Structure Learning with Integer Programming: Polytopes, Facets and Complexity, JAIR **58** (2017).