Integer programming methods to learn causal structures

Sanjeeb Dash
IBM Research

(joint work with Rui Chen, Tian Gao, Joao Goncalves)

Princeton OLC workshop, June 27-28, 2023
Outline

► 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

\[
P(X_1, X_2, X_3, X_4) = P(X_4|X_1) P(X_3|X_1, X_2) P(X_2|X_1) P(X_1)
\]

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
Bayesian Networks

**Defn:** Let $X = (X_1, ..., X_n)$ be random variables. A Bayesian network is a DAG that specifies a joint distribution over $X$ as a product of local conditional distributions, one per node:

$$
\mathbb{P}(X_1 = x_1, \ldots, X_n = x_n) := \prod_{i=1}^{n} \mathbb{P}(x_i \mid x_{\text{Parents}(i)})
$$
Example

Variables:
Era ∈ \{1930s, 1940s, \cdots\},
Genre ∈ \{Comedy, Drama, Horror, Sci-Fi, Action\},
Rating ∈ \{1, 2, 3, 4, 5\}, Color ∈ \{0, 1\}.

Model 1: \(P(E, G, R, C) = P(E) P(G \mid E) P(R \mid E, G) P(C \mid E)\)

Model 2: \(P(E, G, R, C) = P(E) P(G \mid E) P(R \mid G) P(C \mid E)\)
Sparse models

Data:

<table>
<thead>
<tr>
<th>E</th>
<th>G</th>
<th>R</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>1930</td>
<td>Comedy</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>2010</td>
<td>Drama</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>2000</td>
<td>War</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>2010</td>
<td>Drama</td>
<td>3</td>
<td>1</td>
</tr>
</tbody>
</table>

Distribution:

<table>
<thead>
<tr>
<th>E</th>
<th>G</th>
<th>R</th>
<th>C</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>1930</td>
<td>Comedy</td>
<td>2</td>
<td>0</td>
<td>.01</td>
</tr>
<tr>
<td>2010</td>
<td>Drama</td>
<td>3</td>
<td>1</td>
<td>.05</td>
</tr>
<tr>
<td>2000</td>
<td>War</td>
<td>4</td>
<td>0</td>
<td>.01</td>
</tr>
</tbody>
</table>

Original distribution needs $10 \times 5 \times 5 \times 2 = 500$ values.

Model 2 needs $10 + (10 \times 5) + (5 \times 5) + (10 \times 2) = 105$ values

**Inference:** What is the value of $P(G = \text{War})$ or $P(G = \text{War} \mid R \geq 4)$?
Equivalent structures

The BNs $X \rightarrow Y \rightarrow Z$ and $X \leftarrow Y \leftarrow Z$ are indistinguishable.

More generally, each BN belongs to an equivalence class of BNs that yield the same factorization of the joint probability distribution.
Causal Graphs/Causal BN

- Graphical Models where directed edges represent causal relationships
Structural equation models

Directed Acyclic Graph

\[ \begin{align*}
    x_A &= \epsilon_A \\
x_B &= \epsilon_B \\
x_C &= b_{CA}x_A + b_{CE}x_E + \epsilon_C \\
x_D &= b_{DB}x_B + b_{DE}x_E + \epsilon_D \\
x_E &= \epsilon_E
\end{align*} \]

LISREL (Joreskog, Sorbom), Amos - Software to compute \( b_e \) and distribution of \( \epsilon_v \) etc.
Creating causal graphs

Foster, Ipeirotis 2022
Scoring BNs/causal graphs

BN scores: AIC, BIC, fMNL, MDL, BDeu, UPSM, DPSM, Structural Hamming Distance

SEM Scores: RMR, RMSEA, CFI, GFI, NFI ...
BIC 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
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
    \[(a \rightarrow b \rightarrow \ldots \rightarrow a)\]
  - No almost directed cycles
    \[(a \leftrightarrow b \rightarrow c \rightarrow \ldots \rightarrow a)\]
Continuous Guassian distributions

(Linear) Structural equations

\[
\begin{align*}
x_A &= \epsilon_A \\
x_B &= \epsilon_B \\
x_C &= b_{CA}x_A + \epsilon_C \\
x_D &= b_{DB}x_B + \epsilon_D \\
cov(\epsilon_C, \epsilon_D) &= \Omega_{CD}
\end{align*}
\]

If \( \epsilon_A - \epsilon_D \) are normally distributed random variables, then \( x \) has a multivariate normal distribution with covariance matrix \( \Sigma \) given by

\[
(I - B)^{-1}\Omega(I - B)^{-T}
\]

Possible Goal: Given sample covariance matrix \( \hat{\Sigma} \), solve

\[
\min_{B,\Omega} \| \hat{\Sigma} - (I - B)^{-1}\Omega(I - B)^{-T} \|
\]
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 $G$ is given by

$$BIC_G = 2 \ln(l_G(\Sigma)) - \ln(N)(2|V| + |E|)$$

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

$$\ln(l_G(\Sigma)) = -\frac{N}{2} \sum_{D \in D} \left[ |D| \ln(2\pi) + \log\left(\frac{|\hat{\Sigma}_{gD}|}{\prod_{j \in pa_G(D)} \hat{\sigma}_{Dj}^2}\right) + \frac{N - 1}{N} tr(\hat{\Sigma}_{gD}^{-1}S_D - |pa_G(D) \setminus D|) \right]$$

district = component defined by bidirected edges
c-component = district + in-edges per node in district
We obtain a (BIC) score-maximizing ancestral ADMG for a set of continuous variables that follow a multivariate Gaussian distribution.

- Decomposition into c-components

- Ancestral ADMG

- Districts

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

MIP to find optimal AADMG:

$$\max \sum_{c \in C} s_C z_C$$

$$\sum_{C : i \in D(C)} z_C = 1, \quad \forall i \in V$$

$G(z)$ has no directed and almost directed cycles

$z_C \in \{0, 1\}$
Cutting planes to avoid cycles

Cluster Inequalities:

\[ \sum_{i \in S, P \cap S = \emptyset} z_{i,P} \geq 1, \quad \forall S \subseteq V \]

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

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

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

\[
\text{MIP (parent set variables):}
\]

\[
\begin{align*}
\text{max } & \sum_{i \in V} \sum_{P \in P(i)} c_{i,P} z_{i,P} \\
& \sum_{P \in P(i)} z_{i,P} = 1, \forall i \in V \\
& \sum_{i \in S, P \cap S = \emptyset} z_{i,P} \geq 1, \forall S \subseteq V \\
& z_{i,P} \in \{0, 1\}
\end{align*}
\]
## Sparse instances

<table>
<thead>
<tr>
<th>Metric</th>
<th>Orig.</th>
<th>ABIC</th>
<th>GBAP</th>
<th>AA</th>
<th>BF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avg. BIC</td>
<td>-18059.7</td>
<td>-18228.8</td>
<td>-18200.6</td>
<td>-18071.3</td>
<td>-18058.8</td>
</tr>
<tr>
<td>Δ(BIC)</td>
<td>169.1</td>
<td>140.8</td>
<td>11.6</td>
<td>-1.0</td>
<td></td>
</tr>
<tr>
<td>Std. Δ(BIC)</td>
<td>435.5</td>
<td>224.2</td>
<td>17.7</td>
<td>1.9</td>
<td></td>
</tr>
<tr>
<td>Avg. RMSEA</td>
<td>.010</td>
<td>.027</td>
<td>.014</td>
<td>.021</td>
<td>.008</td>
</tr>
<tr>
<td>Std. RMSEA</td>
<td>.009</td>
<td>.019</td>
<td>.008</td>
<td>.023</td>
<td>.009</td>
</tr>
<tr>
<td>Avg. F-Score</td>
<td>.891</td>
<td>.918</td>
<td>.927</td>
<td>.972</td>
<td></td>
</tr>
<tr>
<td>Avg. Time</td>
<td>123</td>
<td>82</td>
<td>30</td>
<td>29</td>
<td></td>
</tr>
</tbody>
</table>

Scores and running time (minutes) for 10 sparse randomly generated 10-node graphs with small c-components and parent sets.

GBAP: Nowzohour et. al. (2017)
ABIC: Bhattacharya et. al. (2020)
Heuristic to create large c-components

Create an initial list of ‘small’ c-components and repeat steps 1-3.

1) Solve LP relaxation of bow-free MIP with current list of c-components

2) Look at active c-components and augment each one by connecting a node to the c-component via a directed or bidirected edge

3) Add newly generated c-components to initial list and go to step 1 or exit if iteration limit reached.

Solve bow-free MIP.
## Medium density instances

<table>
<thead>
<tr>
<th>Metric</th>
<th>Orig.</th>
<th>ABIC</th>
<th>GBAP</th>
<th>LPH</th>
<th>Orig.</th>
<th>ABIC</th>
<th>GBAP</th>
<th>LPH</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avg. BIC</td>
<td>-20880</td>
<td>-20903</td>
<td>-20910</td>
<td>-20889</td>
<td>-20005</td>
<td>-20082</td>
<td>-20200</td>
<td>-20014</td>
</tr>
<tr>
<td>Avg. Δ(BIC)</td>
<td>32.8</td>
<td>30.2</td>
<td>8.8</td>
<td></td>
<td>76.6</td>
<td>195.3</td>
<td>9.1</td>
<td></td>
</tr>
<tr>
<td>Std. Δ(BIC)</td>
<td>43.6</td>
<td>67.5</td>
<td>6.3</td>
<td></td>
<td>127.0</td>
<td>254.8</td>
<td>7.3</td>
<td></td>
</tr>
<tr>
<td>Avg. RMSEA</td>
<td>.010</td>
<td>.085</td>
<td>.014</td>
<td>.011</td>
<td>.007</td>
<td>.032</td>
<td>.020</td>
<td>.017</td>
</tr>
<tr>
<td>Std. RMSEA</td>
<td>.009</td>
<td>.185</td>
<td>.011</td>
<td>.007</td>
<td>.008</td>
<td>.018</td>
<td>.014</td>
<td>.016</td>
</tr>
<tr>
<td>Avg. F-Score</td>
<td>.854</td>
<td>.920</td>
<td>.875</td>
<td></td>
<td>.866</td>
<td>.845</td>
<td>.841</td>
<td></td>
</tr>
<tr>
<td>Avg. Time</td>
<td>139</td>
<td>91</td>
<td>45</td>
<td></td>
<td>229</td>
<td>81</td>
<td>68</td>
<td></td>
</tr>
</tbody>
</table>

Summary of Results - scores and running times for 10-node graphs with either large parent sets or large c-components.

Results for 20 node instances are not as good.
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
