DISENTANGLING MIXTURES OF UNKNOWN CAUSAL INTERVENTIONS

Abhinav Kumar
Gaurav Sinha
Paper ID - 781

CONTENTS

MOTIVATION

- Causal Bayesian Networks (CBN) have become the popular choice to model causal relationships in data.
- CBN is defined as a DAG with edges denoting immediate causal relationships.
- They simulate effects of external interventions via the do() operator
- Real-world interventions are not always precise and end up affecting unintended targets.
- For different units, the unintended targets might be different which leads to the data collected becoming a mixture of causal interventions.
- Finding all possible unintended interventions present in the data can be of high value in applications such as Gene editing.

PROBLEM DEFINITION

Given a CBN G on $V = (V_1, \dots, V_N)$ and its distribution $\mathbb{P}(V)$ along with a mixture distribution $\mathbb{P}_{mix}(V)$ defined as

$$\mathbb{P}_{\boldsymbol{mix}}(\boldsymbol{V}) = \sum_{\boldsymbol{i} \in [\boldsymbol{m}]} \boldsymbol{\pi_i} \mathbb{P}_{\boldsymbol{t_i}}(\boldsymbol{V})$$

-$\pi_i \geq 0$ are scalars, named mixing coefficients
-$\sum_{i \in [m]} \pi_i = 1$
-$\mathbb{P}_{t_i}(V)$ is the interventional distribution $\mathbb{P}(V | do(T_i = t_i))$

Ques: Can we identify the set of intervention tuples $T = \{(\pi_i, t_i), i \in [m]\}$ using $G, \mathbb{P}(V), \mathbb{P}_{mix}(V)$?

$$\mathbb{P}_{mix}(V) = \sum_{i\in[m]} \pi_i \mathbb{P}_{t_i}(V)$$

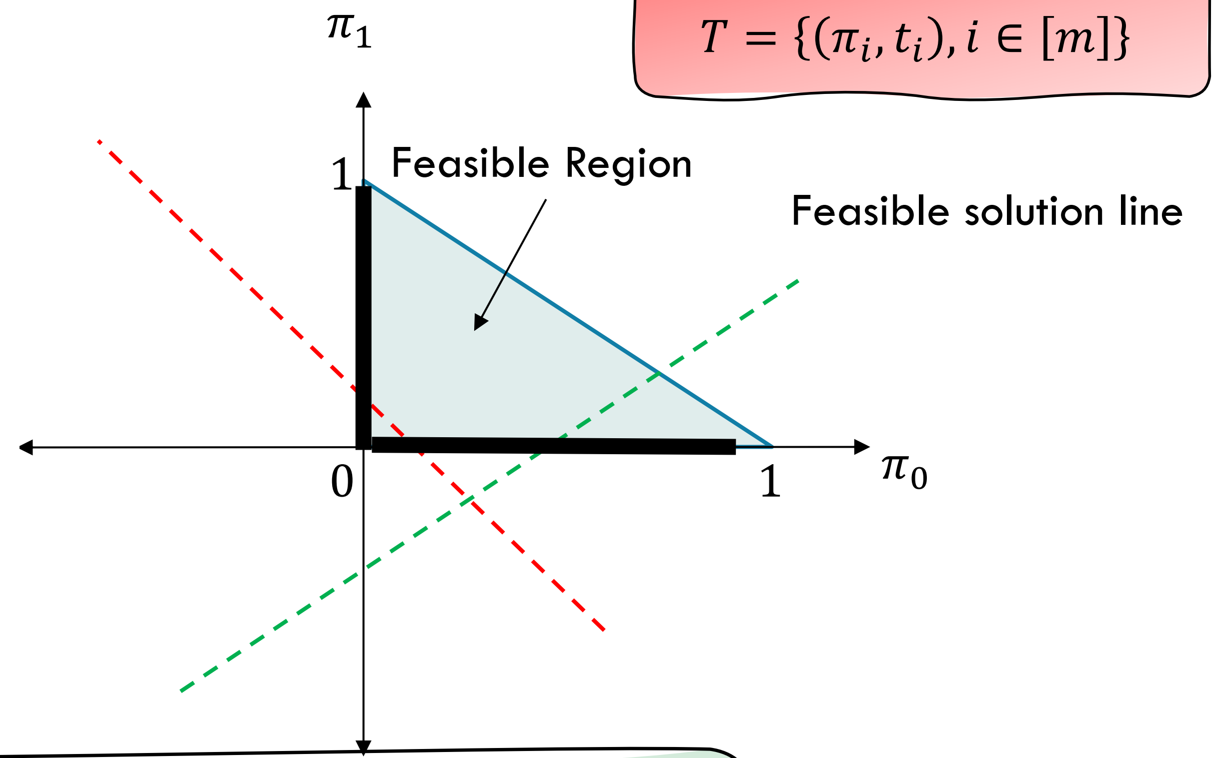
$$T = \{(\pi_i, t_i), i \in [m]\}$$

ASSUMPTIONS

$$\mathbb{P}_{mix}(V_1) = \pi_0\mathbb{P}_0(V_1) + \pi_1\mathbb{P}_1(V_1) + (1 - \pi_0 - \pi_1)\mathbb{P}(V_1)$$

On setting $V_1 = 0$ and then $V_1 = 1$ in the above equation, we obtain

$$\begin{bmatrix} 1 - p_0 & -p_0 \\ p_0 - 1 & p_0 \end{bmatrix} \begin{bmatrix} \pi_0 \\ \pi_1 \end{bmatrix} = \begin{bmatrix} \mathbb{P}_{mix}(V_1 = 0) - p_0 \\ \mathbb{P}_{mix}(V_1 = 1) - p_1 \end{bmatrix}$$



Assumption 1 [Exclusion] :: For every V_i, there is some v_i that v_i that does not appear in any target of the mixture.

$$\mathbb{P}_{mix}(V_1, V_2) = 0.2\ \mathbb{P}_{0,1}(V_1, V_2) + 0.8\ \mathbb{P}\ (V_1, V_2)$$

Assumption 2 [Positivity] :: $\mathbb{P}(v) > 0$ for all settings v of V.

MAIN CONTRIBUTIONS

$$\mathbb{P}_{mix}(V) = \sum_{i \in [m]} \pi_i \mathbb{P}_{t_i}(V)$$

$$T = \{(\pi_i, t_i), i \in [m]\}$$

1. **Theorem:** For any $\mathbb{P}_{mix}(V)$ generated by a T that satisfies exclusion:
   a) T is the unique set satisfying exclusion that generates it.
   b) Given access to $\mathrm{G}, \mathbb{P}(V), \mathbb{P}_{mix}(V)$, there exists an algorithm that runs in time $n * (m * k)^{O(1)}$ and returns T.

2. **Algorithm:** When the access is provided via samples, the above algorithm is modified to run with finite samples.

3. **Simulation:** Benchmark performance of the finite sample algorithm.

PROOF BY INDUCTION

Base Case: (|V|=1)

Inductive Hypothesis: Assume the result for all CBNs and mixtures (satisfying exclusion) on N-nodes.

Induction Step: Use the inductive hypothesis to show the result for all CBNs and the mixture (satisfying exclusion) on N+1 nodes

Algorithm 1: DISENTANGLE

input : Variables $\boldsymbol{V} = (V_1, \ldots, V_{N+1})$, CBN $\mathcal{G}$, Distributions $\mathbb{P}(\boldsymbol{V}), \mathbb{P}_{mix}(\boldsymbol{V})$

output : $\mathcal{T}$: Set of intervention tuples

1. When $|\boldsymbol{V}| = 1$, setup the linear system in Equation 3 and solve it using technique described in Lemma 4.1 to obtain a set $\mathcal{T}$ of intervention tuples. **return** $\mathcal{T}$.
2. Let $V_1 \prec \ldots \prec V_{N+1}$ denote a topological order in $\mathcal{G}$. Marginalize on V_{N+1} to create access to $\mathbb{P}_{mix}(\boldsymbol{V}_N)$ and $\mathbb{P}(\boldsymbol{V}_N)$ where $\boldsymbol{V}_N = (V_1, \ldots, V_N)$. Construct $\mathcal{G}_N = \mathcal{G} \setminus \{V_{N+1}\}$. Recursively call this algorithm with inputs $\mathcal{G}_N$, $\mathbb{P}(\boldsymbol{V}_N)$, $\mathbb{P}_{mix}(\boldsymbol{V}_N)$, to compute the unique set of intervention tuples $\mathcal{S} = \{(\boldsymbol{s}_1, \mu_1), \ldots, (\boldsymbol{s}_q, \mu_q)\}$ that satisfies Assumption 3.1 and generates $\mathbb{P}_{mix}(\boldsymbol{V}_N)$. Let $\boldsymbol{s}_1, \ldots, \boldsymbol{s}_q$ be ordered such that $i \leq j$ implies that $\boldsymbol{s}_j \not\subseteq \boldsymbol{s}_i$.
3. By inspecting $\boldsymbol{s}_j, j \in [q]$, identify $\bar{v}_i \in C_{V_i}$ such that $\bar{v}_i \notin \boldsymbol{s}_j$ for any $j \in [q]$. Define $\boldsymbol{s}_{-j} = \{\bar{v}_i : V_i \notin S_j\}$. Let $C_{V_{N+1}} = \{v^1, \ldots, v^k\}$. For each $i \in [q]$ and $l \in [k]$, create setting $\boldsymbol{v}_{i,l} = \boldsymbol{s}_i \cup \boldsymbol{s}_{-i} \cup \{v^l\}$.
4. For each $i \in [q]$, evaluate distributions at $\boldsymbol{v}_{i,l}$, to setup the system of equations described in Equation 9. Solve the system using the technique outlined in proof of Lemma 4.7. At the end of this process collect all the intervention tuples thus obtained, in the set $\mathcal{T}$. **return** $\mathcal{T}$.

PROOF IDEA ($|V| = 1$)

$$\mathbb{P}_{mix}(V) = \sum_{i\in[m]} \pi_i \mathbb{P}_{t_i}(V)$$

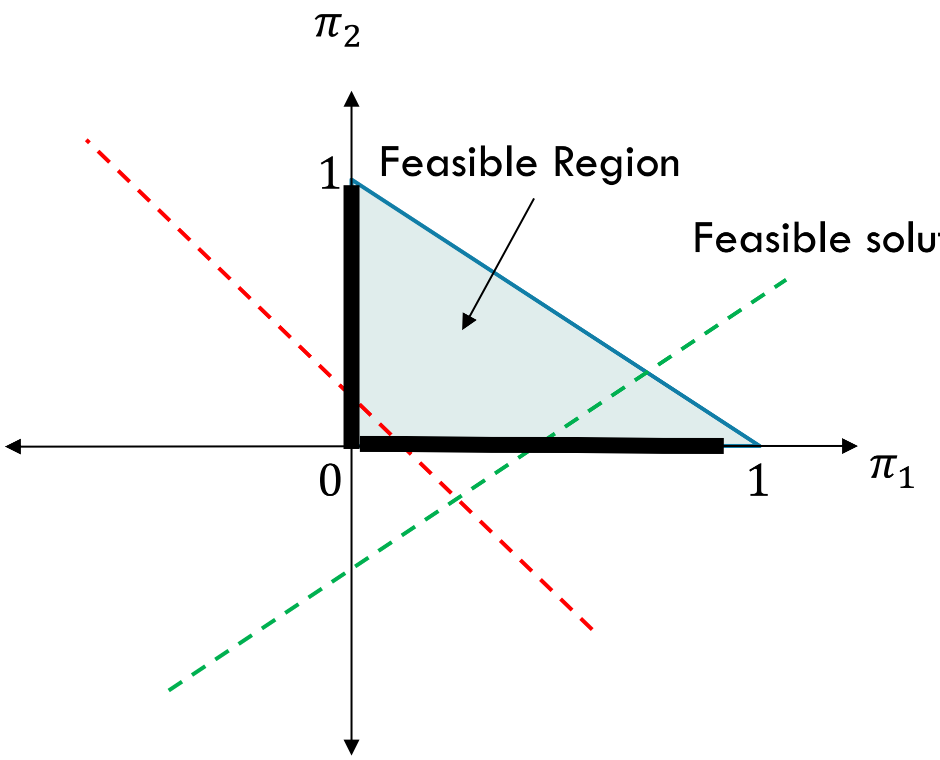
$$T = \{(\pi_i, t_i), i \in [m]\}$$

$C_V = \{v^1, \dots, v^k\}$ is the set of values that V takes

$$\mathbb{P}_{mix}(V) - \mathbb{P}(V) = \sum_{i=1}^{k} \pi_i (\mathbb{P}_{v^i}(V) - \mathbb{P}(V)).$$

$$\begin{bmatrix} 1-a_1 & -a_1 & . & . & -a_1 \\ -a_2 & 1-a_2 & . & . & -a_2 \\ . & . & . & . & . \\ -a_k & -a_k & . & . & 1-a_k \end{bmatrix} \begin{bmatrix} \pi_1 \\ \pi_2 \\ . \\ \pi_k \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ . \\ b_k \end{bmatrix}$$

where $b_i = \mathbb{P}_{mix}(v^i) - \mathbb{P}(v^i)$ and $a_i = \mathbb{P}(v^i) > 0$

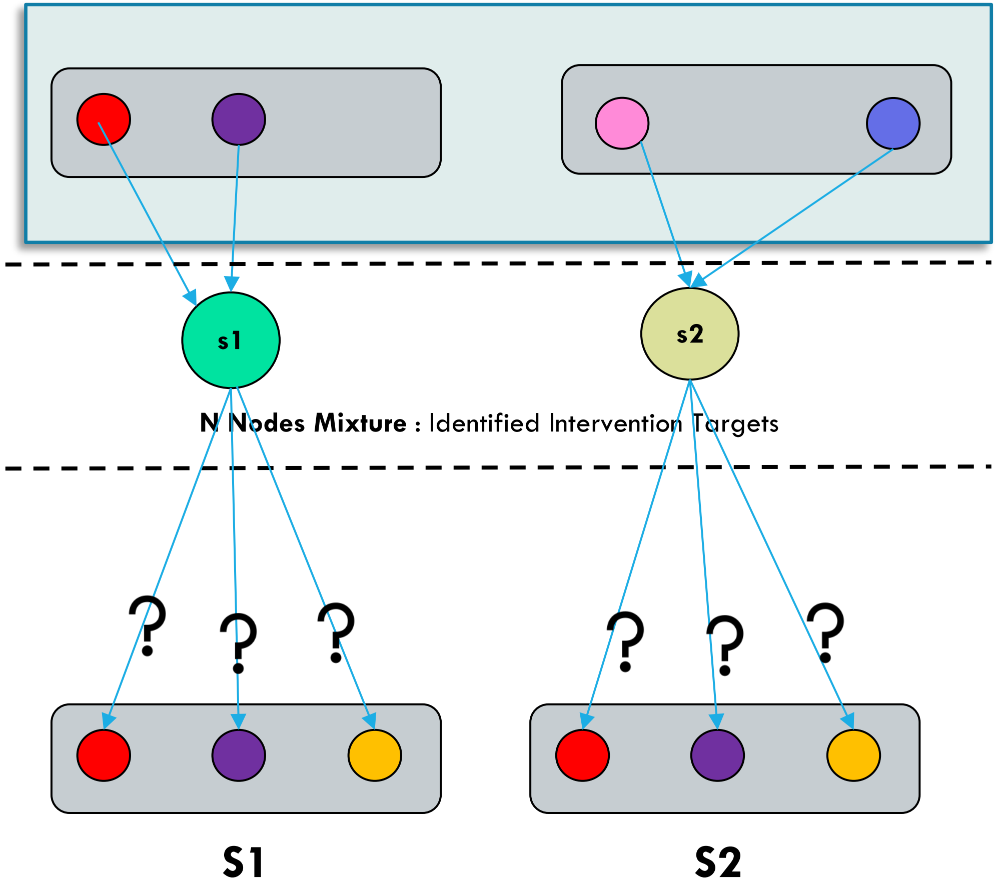


PROOF IDEA ($|V| > 1$) - INDUCTION

$$\mathbb{P}_{mix}(V) = \sum_{i \in [m]} \pi_i \mathbb{P}_{t_i}(V)$$

$$T = \{(\pi_i, t_i), i \in [m]\}$$

N+1 Nodes Mixture : Unknown actual targets



N+1 Nodes Mixture : Identify unknown targets from estimated intervention targets

Marginalize

- Assume $V_1 < \cdots < V_{N+1}$
- Marginalize on V_{N+1} to get mixture on $(V_1, \ldots, V_N)$.
- Recursively compute targets $s_1, \ldots, s_q$ that generates this new mixture.

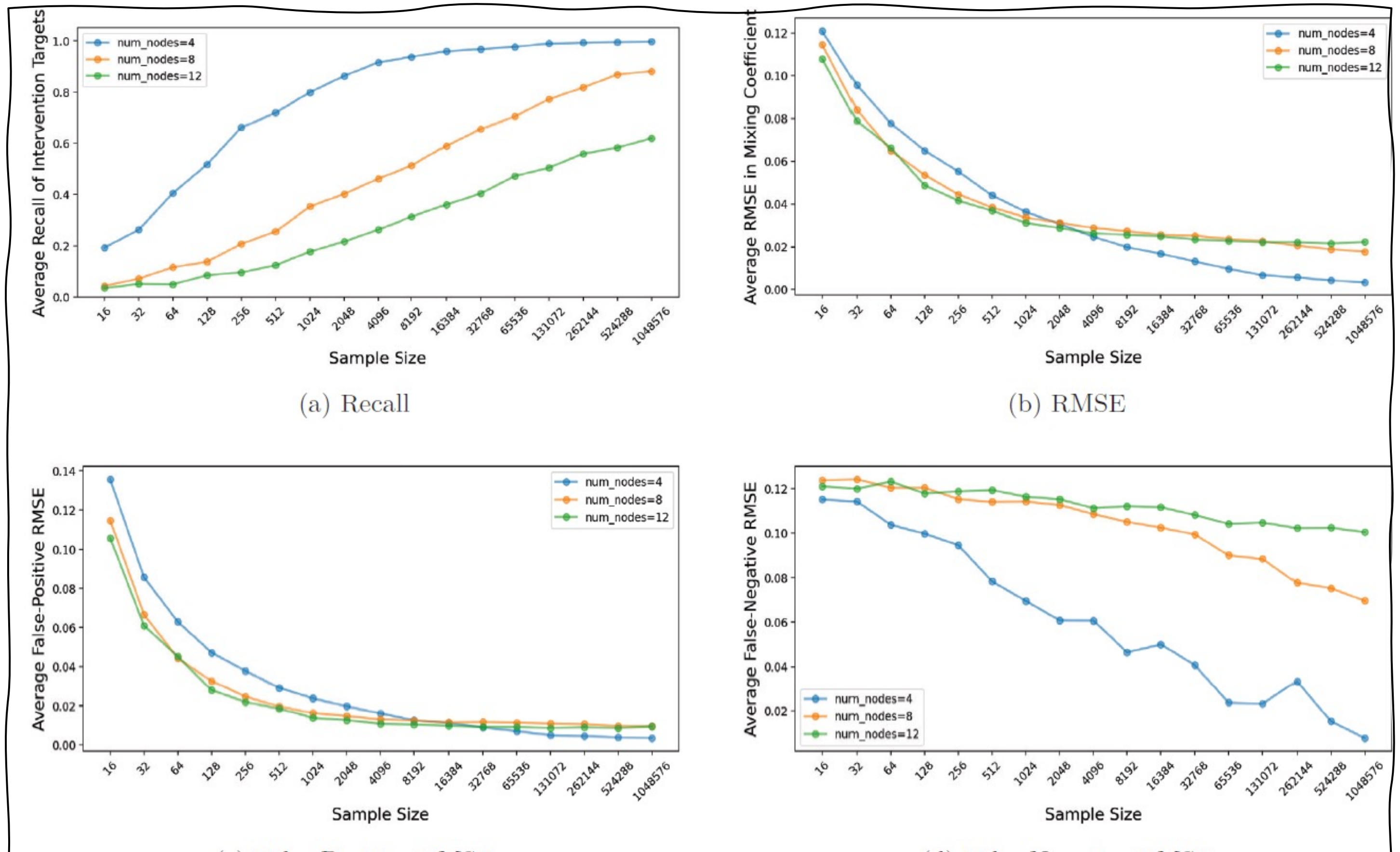
Lift:

- Define sets $S_i = \{s_i, s_i \cup \{v^1\}, \ldots, s_i \cup \{v^l\}\}$.
  - $\{t_1, \ldots, t_m\} \subset S_1 \cup \cdots \cup S_q$
- Gives equations $\mathbb{P}_{mix}(V) = \sum_{j \in [q]} \sum_{s \in S_j} \pi_s \mathbb{P}_s(V)$
- This system has a unique solution under Assumptions 1 and 2.
- Sort Targets: $s_i < s_j \; if \; s_j \not\subset s_i$
- Evaluate at setting:

$$\boldsymbol{v}_{j,l} = \boldsymbol{s}_j \cup \boldsymbol{s}_{-j} \cup \{v^l\}$$

$$\boldsymbol{s}_{-j} = \{\bar{v}_i : V_i \notin S_j\}$$

EXPERIMENT RESULTS



(a) Recall

(b) RMSE

(c) False-Positive RMSE

(d) False-Negative RMSE

Figure 1: Performance of Algorithm 2 for varying sample sizes and number of nodes