Estimating Causal Effects from Learned Causal Networks

Anna K Raichev(araichev@uci.edu), Jin Tian(jtian@iastate.edu),
Alex Ihler(ihler@ics.uci.edu), and Rina Dechter (dechter@ics.uci.edu)

Overview

We propose an alternative to the estimand based paradigm for answering causal queries. The idea is to learn the full causal model from the observational data and causal diagram, and then answer the query by applying Probabilistic Graphical Models (PGM) algorithms. We show that this model completion learning approach can be far more effective than estimated approaches, particularly in large models when the estimand computation is complex and the induced width of the diagram is small.

Contributions:

1. Provide a first of its kind, extensive empirical evaluation on causal effect algorithms on varied and large synthetic and real networks.
2. Show empirically that our approach has more accurate estimates than estimated based schemes.

Problem

Given a causal diagram, an identifiable query $P(\boldsymbol{Y} \mid do(\boldsymbol{X} = x))$ and samples from the observed distribution, the task is to output the distribution of $P(\boldsymbol{Y} \mid do(\boldsymbol{X} = x))$.

Current Practice

1. Apply state of the art algorithms for identifiability. These are polynomial algorithms involving the graph and the query only. [Tian, 2002]
2. Generate an estimand, namely an algebraic expression for the query involving only probabilistic expressions over the visible variables.
3. Estimate the estimand from the observational data.

Limitations

1. More sophisticated statistical estimation techniques don't scale when functions in the estimand are too large.
2. We can use the *Plug-In method***,** in which each term is estimated only on the configurations seen in the observed data. However, this approach also limits the quality of our estimates.

Motivating Example

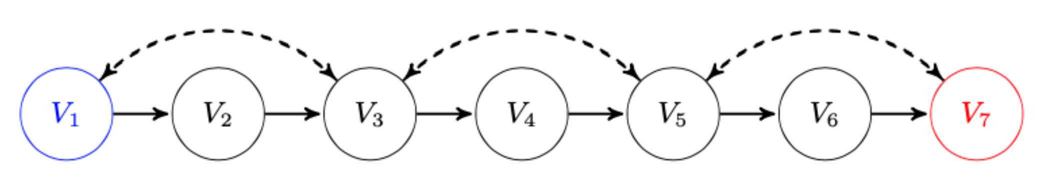


Figure 1: Chain Model with 7 observable variables and 3 latent variables

Using the estimated based approach we get the expression:

$$P(V_7 \mid do(V_1)) = \sum_{V_2,V_3,V_4,V_5,V_6} P(V_6 \mid V_1, V_2, V_3, V_4, V_5) P(V_4 \mid V_1, V_2, V_3) P(V_2 \mid V_1)$$
$$\times \sum_{V_1'} P(V_7 \mid V_1', V_2, V_3, V_4, V_5, V_6) P(V_5 \mid V_1', V_2, V_3, V_4) P(V_3 \mid V_1', V_2) P(V_1')$$

- As model size increases, we have scalability issues.
- However, the induced width of this model is only 3.

Background

Structural Causal Model: $\boldsymbol{M} = \langle \boldsymbol{U}, \boldsymbol{V}, \boldsymbol{F}, P(\boldsymbol{U}) \rangle$

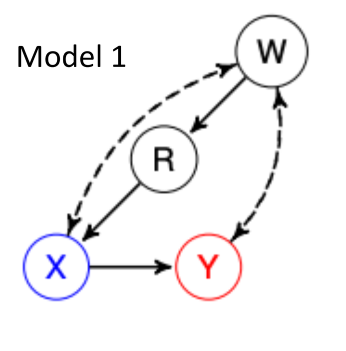
- $\boldsymbol{U} = \{U_1, \ldots, U_k\}$ set of unmeasurable latent variables
- $\boldsymbol{V} = \{V_1, V_2, \ldots, V_n\}$ set of observable variables
- $\boldsymbol{F} = \{f_i : V_i \in \boldsymbol{V}\}$ is a set of functional mechanisms f_i that each determine the value v_i of their corresponding V_i as a function of V_i's causal parents $PA_i \subseteq \boldsymbol{U} \cup \boldsymbol{V} \setminus V_i$
- $P(\boldsymbol{U})$ is a probability distribution over the exogenous variables

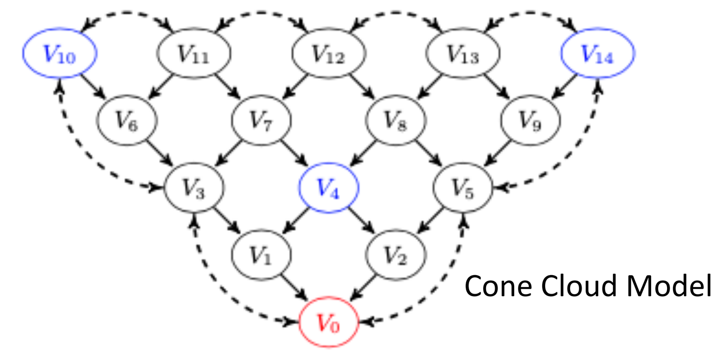
Causal Diagram: A SCM $\boldsymbol{M}$ can be associated with a directed graph $G = \langle \boldsymbol{V} \cup \boldsymbol{U}, E \rangle$ called a causal diagram. Each node in the graph uniquely corresponds to a variable in the SCM. There is an arc from node $X \in (\boldsymbol{U} \cup \boldsymbol{V})$ to node $V_i \in \boldsymbol{V}$ iff $X \in PA_i$

Causal effect and the truncation formula: We use $P(Y \mid do(X))$ to denote the distributions resulting from an intervention which fixes the value of X, and is called the causal effect of $do(X)$ on Y

$$P(\boldsymbol{V}, \boldsymbol{U} \mid do(\boldsymbol{X})) = \prod_{V_j \notin \mathbf{X}} P(V_j \mid PA_j) \cdot P(\boldsymbol{U})$$

Causal Diagrams:





Blue variables are intervened on and red variables are the outcome variables corresponding to the query $P(Y \mid do(X))$

Learning for Causal Inference

Identifiability

- Any two models that agree on the observational distribution and causal diagram will also agree on $P(\boldsymbol{Y} \mid do(\boldsymbol{X} = x))$.

EM for Causal Inference (EM4CI)

Algorithm 1: EM4CI

input : A causal diagram $\mathcal{G} = \langle \boldsymbol{U} \cup \boldsymbol{V}, E \rangle$, $\boldsymbol{U}$ latent and $\boldsymbol{V}$ observables; $\mathcal{D}$ samples from $P(\boldsymbol{V})$;
output : Estimated $P(\boldsymbol{Y} \mid do(\boldsymbol{X} = \boldsymbol{x}))$

// k= latent domain size, $BIC_{\mathcal{B}} = BIC$ score of $\mathcal{B}, \mathcal{D}$,
// $LL_{\mathcal{B}}$ is the log-likelihood of $\mathcal{B}, \mathcal{D}$
Initialize: $BIC_{\mathcal{B}} \leftarrow \inf$,
If $\neg$identifiable($\mathcal{G}, Q$), terminate.
for $k = 2, \ldots,$ to upper bound, **do**
  $(\mathcal{B}', LL_{\mathcal{B}'}) \leftarrow \max_{LL} \{\text{EM}(\mathcal{G}, \mathcal{D}, k) \mid$**for** $i = 1$ to $10\}$
  Calculate $BIC_{\mathcal{B}'}$ from $LL_{\mathcal{B}'}$
  if $BIC_{\mathcal{B}'} \leq BIC_{\mathcal{B}}$,
    $\mathcal{B} \leftarrow \mathcal{B}'$, $BIC_{\mathcal{B}} \leftarrow BIC_{\mathcal{B}'}$
  else, break.
endfor
$\mathcal{B}_{\boldsymbol{X}=\boldsymbol{x}} \leftarrow$ generate truncated CBN from $\mathcal{B}$.
return $\leftarrow$ evaluate $P_{\mathcal{B}_{\boldsymbol{X}=\boldsymbol{x}}}(\boldsymbol{Y})$

1. Check if query is identifiable.
2. Using samples from the observed distribution $P(\boldsymbol{V})$ to learn a full causal Bayesian network B with domain size *k* consistent with $(\mathcal{G}, P(\boldsymbol{V}))$ using the the resulting model with maximum log likelihood from running the EM algorithm ran 10 times.
3. Compute the BIC score and increase *k*.
4. Stop when we find the minimum BIC score.
5. truncate $\boldsymbol{M}$ into the causal model $\boldsymbol{M}_X$ by removing the function associated with X and assigning $X = x$ in all functions where X appears.
6. Apply a PGM algorithm to answer apply a PGM algorithm to answer the associated query $P(\boldsymbol{Y} \mid \boldsymbol{X} = x)$.
7. Return $P(\boldsymbol{Y} \mid do(\boldsymbol{X} = x))$.

Complexity

- Time and memory are exponential in the induced width.

Benefits & Challenges

Challenges

1. In order to learn the full model we need to learn a domain size for the latent variables.
2. There exists theoretical bounds on sufficient domain sizes. However the bounds are very conservative & can be very large to be practical [J. Zhang et al, 2022].
3. EM algorithm can be slow and converge to incorrect local optima in high dimensional space.

Benefits

1. Learning phase only needs to be performed once to answer *any* identifiable query of form $P(\boldsymbol{Y} \mid do(\boldsymbol{X} = x))$; traditionally a new estimand would need to be derived for each query.
2. EM4CI consistently yields extremely accurate results.

Experimental Setup

Benchmarks

- Each benchmark includes a causal diagram, a query, and observational data synthetically generated from the full model.
- Used a range of domain sizes of for the variables.
- Test on bayesian networks from real world domains, and created latent confounders from the source vertices.

Performance Measures

- To evaluate the accuracy of $P(\boldsymbol{Y} \mid do(\boldsymbol{X} = x))$, we use the mean absolute deviation (*mad*): averaging the absolute error over all single-value queries over all instantiations of the intervened and queried variables.
- BIC score is used to evaluate fitness of the learned model and impose some regularization over the domain sizes.

Notation

- Capital letters (X) represent variables, & small letters (x) represent their values. Boldfaced capital letters ($\mathbf{X}$) denote a collection of variables.
- $n = |\mathbf{V}|$, $d = |D(V)|$, $k = |D(U)|$ in the true model, and $k_{lrn} = |D'(U)|$ the latent domain of the learned model

Empirical Analysis

Baseline Comparison: Plug In Method

Table 3: Results of EM4CI & Plug-In on $P(Y \mid do(\mathbf{X}))$ $(d, k) = (2, 10)$

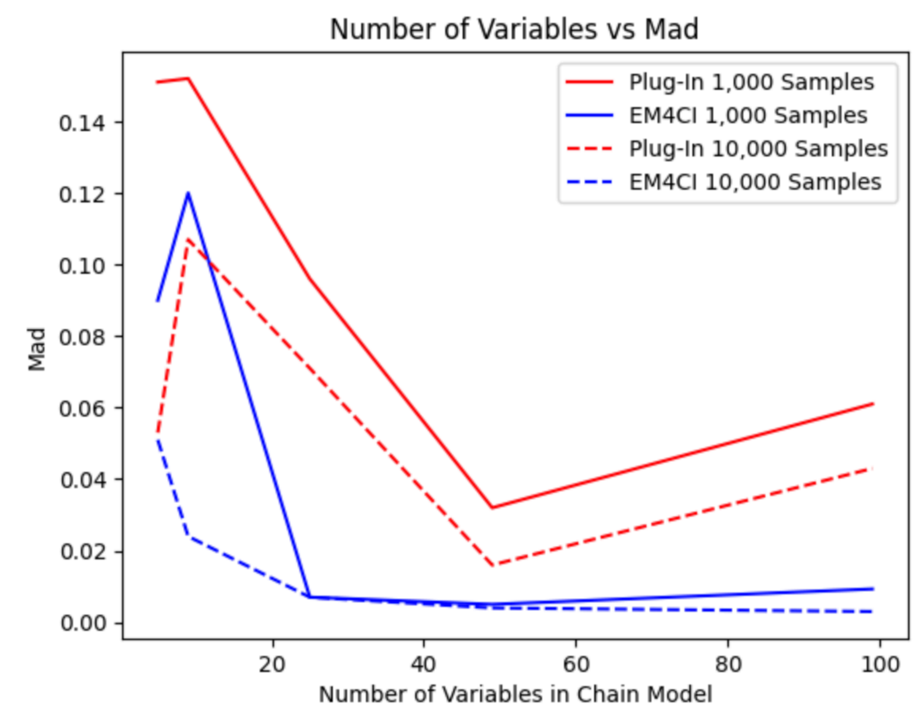
	100 Samples					1,000 Samples				
		EM4CI		Plug-In			EM4CI		Plug-In	
Model	k_{lrn}	mad	time(s)	mad	time(s)	k_{lrn}	mad	time(s)	mad	times(s)
1	2	0.0037	0.4759	0.0104	1.9	2	0.0032	3.1	0.0025	2.3
2	2	0.1832	1.8643	0.1436	2.3	2	0.0490	8.4	0.0867	2.0
3	2	0.1288	0.9288	0.0569	1.1	2	0.0040	3.6	0.0039	0.7
4	2	0.1819	1.8169	0.1469	2.3	2	0.1438	12.0	0.0704	2.1
5	2	0.4910	1.6539	0.5000	2.0	2	0.0044	17.3	0.0058	2.2
6	2	0.2663	0.3004	0.3930	2.0	2	0.1263	0.5	0.1319	2.1
7	2	0.2520	0.7757	0.2509	1.9	2	0.0891	7.1	0.0238	2.0
8	2	0.1372	0.6348	0.1579	2.0	2	0.2340	4.7	0.1303	1.9

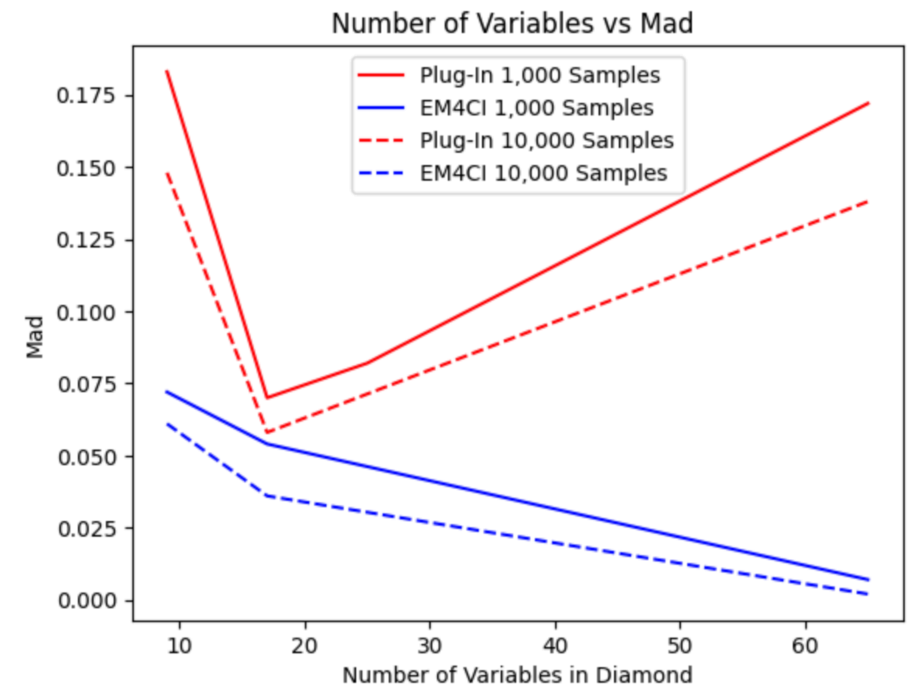
Competing Scheme: WERM [Y. Jung et al., 2020]

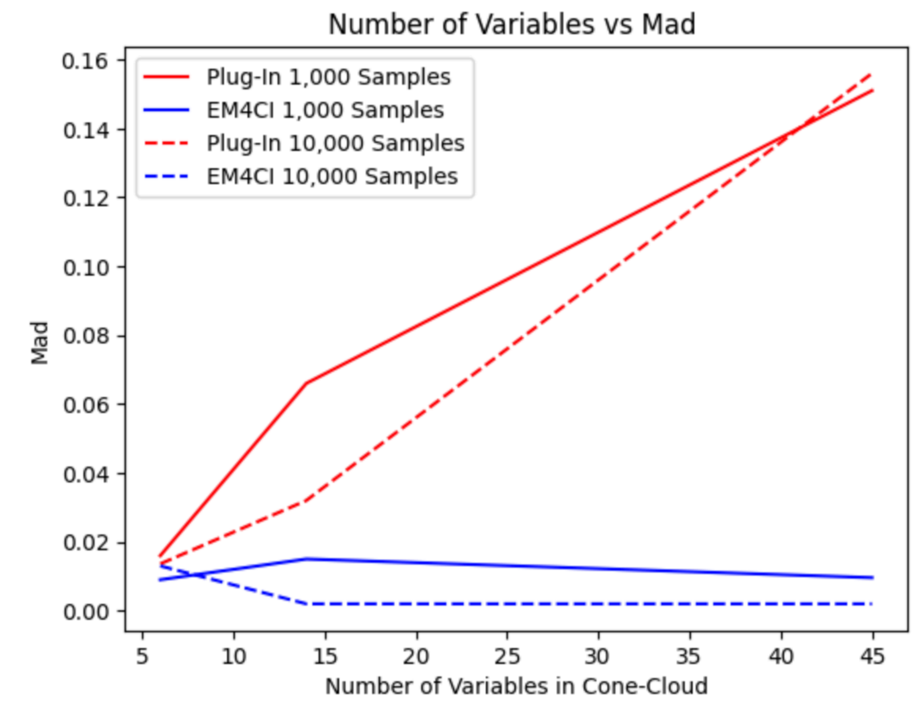
	1,000 Samples					10,000 Samples				
	WERM		EM4CI			WERM		EM4CI		
Model	error	time(s)	error	time(s)	k_{lrn}	error	time(s)	error	time(s)	k_{lrn}
1	0.0071	18.7	0.0059	8.8	2	0.0031	32.6	0.0046	63.5	2
8	0.1082	25.8	0.1566	7.6	2	0.11	47.7	0.0001	81.4	2
3'	0.027	27.2	0.0004	3.5	2	0.001	44.1	0.0009	53.1	2

- Learns causal effects by weighted empirical risk minimization.
- State of the art method that focuses on estimating the quantities in the estimand using statistical methods.

Synthetic Network Results



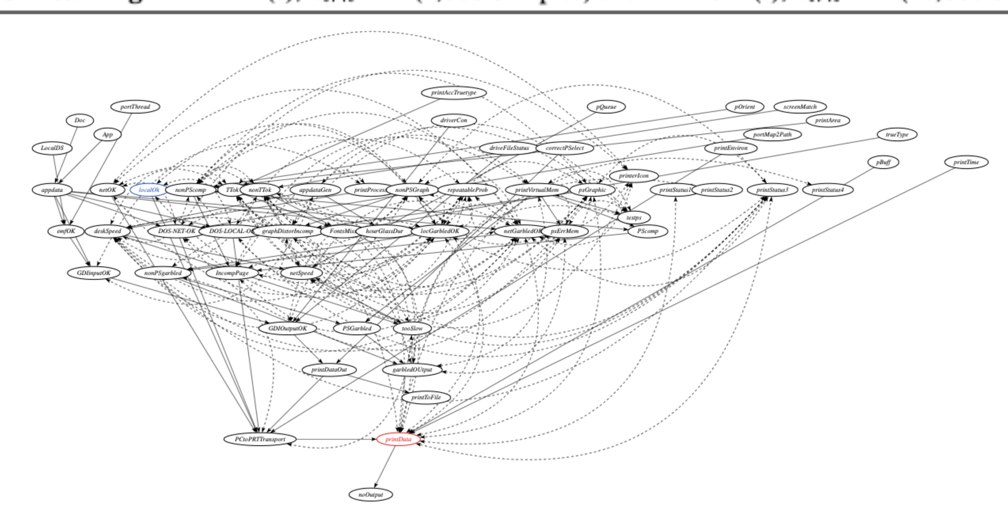




UAI Benchmark Results

Table 1: Plug-In & EM4CI results on the **A Network**
$|\boldsymbol{V}| = 46;\ |\boldsymbol{U}| = 8;\ d = 2;\ k = 2$ treewidth ≈ 16

	Plug-In				*EM4CI*			
	1,000 Samples		*10,000 Samples*		*1,000 Samples*		*10,000 Samples*	
Query	mad	time(s)	mad	time(s)	mad	time(s)	mad	time(s)
$P(V_{51} \mid \text{do}(V_{10}))$	0.0584	8.0	0.0114	55.7	0.0139	0.0012	0.0083	0.0012
$P(V_{51} \mid \text{do}(V_{14}))$	0.0319	8.3	0.0056	51.3	0.0143	0.0047	0.0086	0.0046
$P(V_{51} \mid \text{do}(V_{41}))$	0.0255	13.9	0.0092	48.3	0.0147	0.0042	0.0079	0.0041
$P(V_{51} \mid \text{do}(V_{45}))$	0.0496	9.8	0.0206	49.1	0.0140	0.0031	0.0082	0.0030
EM4CI Learning	time=71(s), $k_{lrn} = 4$ (1,000 Samples) and time=541(s), $k_{lrn} = 4$ (10,000 Samples)							



Conclusion

- EM4CI was extremely accurate on all benchmarks we tried.
- Inference on multiple queries was very fast after learning.
- EM4CI is another tool for causal inference, not meant to replace the estimand based approach but used as an alternative when beneficial.