Overview
We propose an alternative paradigm for answering causal queries. The idea is to learn the full causal model from the observational data, and once a full model is available, the query can be answered by applying Probabilistic Graphical Models (PGM) algorithms. We show that when the diagram has a low induced-width this approach can be far more effective than the estimand-based approach.

Contributions:
1. A general scheme, Le4CI, for computing causal queries that utilizes well-known algorithms for learning and inference, and the special case EM4CI that utilizes EM for learning.
2. An analysis of the scheme's theoretical properties, highlighting its challenges and benefits.

Problem
Given a causal diagram, a query \( P(Y \mid do(X = x)) \) and samples from the observed distribution, the task is to determine if the query can be answered (identifiability). If it is, then output the distribution \( P(Y \mid do(X = x)) \).

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.

Limitation: functions in the estimand may be too large to estimate.

Our approach:
1. First learn a full causal model
2. Answer the query using PGM tools.

Motivating Example

Background
Structural Causal Model: \( M = (U, V, P(U)) \)
- \( U = \{U_1, \ldots, U_l\} \) set of unmeasurable latent variables
- \( V = \{V_1, \ldots, V_l\} \) set of observable variables

Causal Diagram: A SCM \( M \) can be associated with a directed graph \( G = (V, U, E) \) called a causal diagram. Each node in the graph uniquely corresponds to a variable in the SCM. There is an arc from node \( X \in (U \cup V) \) to node \( Y \in (U \cup V) \) if \( X \in P(Y) \).

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

Learning for Causal Inference

Identification
- Any two models that agree on the observational distribution and causal diagram will also agree on \( P(Y \mid do(X = x)) \).\[ \]

EM for Causal Inference (EM4CI)

Algorithm 1: EM4CI
Input: SCM \( M = (U, V, P(U)) \); causal query \( Q = P(Y \mid do(X = x)) \)
Output: \( P(Y \mid do(X = x)) \)
Step 1: \( M = (U, V, P(U)) \)
Step 2: \( v = v' \mapsto f(V_{\text{do}}(v)) \)
Result: \( P(Y \mid do(X = x)) \)

Benefits & Challenges

Challenges:
1. In order to learn the full model we need to assume 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. [Jiang 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 form of \( P(Y \mid do(X = x)) \), traditionally a new estimator would need to be derived for each query
2. Utilize the breadth of tools developed for graphical models
3. Expressions can be computationally intensive even for small induced models, where learning is easy

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 the observed and latent variables

Performance Measures:
- To evaluate the error of \( P(Y \mid do(X = x)) \), we use the mean absolute deviation (mad) and mean relative deviation (mrd)
- To evaluate the fitness of the learned model relative to the data we use the average log likelihood (LL)

Empirical Analysis

Baseline Comparison: Plug In Method
- Generates an estimand and the empirical conditional probabilities are computed from observational quantities
- Will converge to the exact result given enough samples

(a) 100 samples \( d = 2, k = 10, k_{\text{hop}} = 16 \)

Competing Scheme: WERM [Jang et al., 2020]

Testing Different Hypothesized Domains

Baseline Comparison: Plug In Method

Table: Performance Comparison between EM and EM4CI with \( d = 2 \)

<table>
<thead>
<tr>
<th>Model</th>
<th>True Value</th>
<th>EM4CI (mad, time(s))</th>
<th>Plug-In (error, time(s))</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( 0.530 ) (0.0574, 0.006)</td>
<td>(0.054, 1.008)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( 0.441 ) (0.0555, 0.025)</td>
<td>(0.0708, 0.029)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( 0.519 ) (0.0235, 0.0264)</td>
<td>(0.0199, 0.016)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( 0.479 ) (0.0186, 0.047)</td>
<td>(0.0049, 0.041)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( 0.449 ) (0.0085, 0.097)</td>
<td>(0.0123, 0.024)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( 0.512 ) (0.013, 0.002)</td>
<td>(0.0195, 0.022)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( 0.512 ) (0.0156, 0.007)</td>
<td>(0.0950, 0.027)</td>
<td></td>
</tr>
</tbody>
</table>

Testing Difficulty Measures

<table>
<thead>
<tr>
<th>Measure</th>
<th>Formula</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAD</td>
<td>( \frac{1}{n} \sum_{i=1}^{n}</td>
<td>P(Y_i \mid do(X_i = x_i)) - P(Y_i)</td>
</tr>
<tr>
<td>MRD</td>
<td>( \frac{1}{n} \sum_{i=1}^{n} \frac{</td>
<td>P(Y_i \mid do(X_i = x_i)) - P(Y_i)</td>
</tr>
<tr>
<td>LL</td>
<td>( \frac{1}{n} \sum_{i=1}^{n} \ln P(Y_i \mid do(X_i = x_i)) )</td>
<td>Average Log Likelihood</td>
</tr>
</tbody>
</table>

Conclusion
- \( EM4CI \) was fast compared to other methods
- \( MAD \) and \( mrd \) were small on most benchmarks
- \( EM4CI \) is another tool for causal inference, not meant to replace the estimand based approach but used as an alternative when beneficient (low induced models) are used

Open Questions
- What is the best hypothesized domain to use?