General Purpose SAT-Solvers for Causal Discovery

Frederick Eberhardt
Caltech

[joint work with Antti Hyttinen and Matti Jarvisalo]
Causal Discovery

data
sample

$w \ x \ y \ z$

samples
Causal Discovery

Data sample

\[
\begin{array}{cccc}
  w & x & y & z \\
\end{array}
\]

Assumptions, e.g.
- causal Markov
- causal faithfulness
- functional form
- etc.

Inference algorithm
Causal Discovery

assumptions, e.g.
• causal Markov
• causal faithfulness
• functional form
• etc.

equivalence classes

model specifications

direct edges

confounders

samples

inference algorithm
General Model Space

assumption / algorithm

Markov
faithfulness
causal sufficiency
acyclicity
parametric assumption
### General Model Space

<table>
<thead>
<tr>
<th>assumption / algorithm</th>
<th>PC / GES</th>
<th>FCI</th>
<th>CCD</th>
<th>LiNGaM</th>
<th>non-linear additive noise</th>
</tr>
</thead>
<tbody>
<tr>
<td>Markov faithfulness</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>causal sufficiency</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>acyclicity</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>parametric assumption</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>linear non-Gaussian noise</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>non-linear additive noise</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>
## General Model Space

<table>
<thead>
<tr>
<th>assumption / algorithm</th>
<th>PC / GES</th>
<th>FCI</th>
<th>CCD</th>
<th>LiNGaM</th>
<th>non-linear additive noise</th>
</tr>
</thead>
<tbody>
<tr>
<td>Markov faithfulness</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>causal sufficiency</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>acyclicity</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>parametric assumption</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>
## General Model Space

<table>
<thead>
<tr>
<th>Assumption / Algorithm</th>
<th>PC / GES</th>
<th>FCI</th>
<th>CCD</th>
<th>LiNGaM</th>
<th>Non-linear Additive Noise</th>
</tr>
</thead>
<tbody>
<tr>
<td>Markov faithfulness</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Causal sufficiency</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Acyclicity</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Parametric assumption</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

- **w/ latents**
- **w/ cycles**
- **w/ both**
Combining Experiment and Observation

experiment

observational study
Combining Experiment and Observation

Experiment

Observational Study
Combining Experiment and Observation

true (unknown) model

experiment

observational study

samples
Causal Discovery

assumptions, e.g.
- causal Markov
- causal faithfulness
- functional form
- etc.

causal structures consistent with data

inference algorithm

experimental samples

observational samples

data sample

w x y z

w x y
Causal Discovery

assumptions, e.g.
• causal Markov
• causal faithfulness
• functional form
• etc.

causal structures consistent with data

samples

inference algorithm

etc.
Causal Discovery

**background knowledge**

samples

assumptions, e.g.
- causal Markov
- causal faithfulness
- functional form
- etc.

inference algorithm

causal structures consistent with data

e.g.

\[
\begin{align*}
\text{Sample} &: w, x, y, z \\
\text{Inference} &: (w, x, y) \\
\text{Causal Structure} &: (x, y, z, w)
\end{align*}
\]
Causal Discovery

**background knowledge**

- edge presences/absences

**assumptions, e.g.**
- causal Markov
- causal faithfulness
- functional form
- etc.

**causal structures consistent with data**

samples

inference algorithm

background knowledge

• edge presences/absences

assumptions, e.g.

- causal Markov
- causal faithfulness
- functional form
- etc.
Causal Discovery

background knowledge

• edge presences/absences
• pathways

assumptions, e.g.
• causal Markov
• causal faithfulness
• functional form
• etc.

causal structures consistent with data

inference algorithm

etc.
Causal Discovery

**background knowledge**

\[ x \quad y \quad w \]

- edge presences/absences
- pathways
- tier orderings

**assumptions, e.g.**
- causal Markov
- causal faithfulness
- functional form
- etc.

**causal structures consistent with data**

**inference algorithm**

\[ x \quad y \quad z \quad w \]

\[ \text{samples} \]

\[ w \quad x \quad y \quad z \]

etc.
Causal Discovery

**background knowledge**

- edge presences/absences
- pathways
- tier orderings
- “priors”

**assumptions, e.g.**
- causal Markov
- causal faithfulness
- functional form
- etc.

**causal structures consistent with data**

**inference algorithm**

```
| x | z |
| y | w |
```

```
| w | x | y | z |
```

```
| x | y |
| z | w |
```
Settings: examples

subsampled time series

(time)

(cf. work by Plis & Danks)
Settings: examples

subsampling time series

(cf. work by Plis & Danks)
Settings: examples

subsampled time series

(cf. work by Plis & Danks)

inference algorithm
Settings: examples

subsampled time series

(cf. work by Plis & Danks)

biological settings

(cf. work by Murray-Watters & Glymour)

inference algorithm
Causal Effects

inference algorithm
Causal Effects

inference algorithm    equivalence class
Causal Effects

inference algorithm ➟ equivalence class ➟ $P(y|\text{do}(w))$

causal effect
Causal Effects

How to apply the do-calculus in settings when the causal structure is underdetermined?
High-Level
High-Level

data
sample

samples

$w \ x \ y \ z$

samples

$w \ x \ y$
High-Level

assumptions, e.g.
• causal Markov
• causal faithfulness
• etc.

samples

w  x  y  z

samples

w  x  y
High-Level

Data sample

Assumptions, e.g.
- causal Markov
- causal faithfulness
- etc.

Background knowledge, e.g.
- pathways
- tier ordering
- “priors”
- etc.
High-Level

data sample

assumptions, e.g.
• causal Markov
• causal faithfulness
• etc.

background knowledge, e.g.
• pathways
• tier ordering
• “priors”
• etc.

setting
• subsampled time series
• tier structure

samples

w x y z

samples

w x y
High-Level

data sample

assumptions, e.g.
- causal Markov
- causal faithfulness
- etc.

background knowledge, e.g.
- pathways
- tier ordering
- “priors”
- etc.

setting
- subsampled time series
- tier structure

samples

\[ w \ x \ y \ z \]

samples

\[ w \ x \ y \]

(in)dependence constraints

\[ x \not\perp y | C | J \]
High-Level

- Data sample
  - Assumptions, e.g.
    - Causal Markov
    - Causal faithfulness
    - Etc.
  - Background knowledge, e.g.
    - Pathways
    - Tier ordering
    - "Priors"
    - Etc.
  - Setting
    - Subsampled time series
    - Tier structure

Encode these as logical constraints on the underlying graph structure.
High-Level

- assumptions, e.g. causal Markov, causal faithfulness, etc.
- background knowledge, e.g. pathways, tier ordering, "priors", etc.
- setting: subsampled time series, tier structure

Encode these as logical constraints on the underlying graph structure

(in)dependence constraints

$\mathbf{x} \not\perp \mathbf{y} | \mathbf{C} \parallel \mathbf{J}$

(data sample)

(max) SAT-solver
d-separation and independence

• Under the assumption of causal Markov and causal Faithfulness:

\[ x \not\perp y \mid C \parallel J \iff x \not\perp y \mid C \parallel J \]
d-separation and independence

- Under the assumption of causal Markov and causal Faithfulness:

\[ x \not\perp y \mid C \parallel J \iff x \not\perp y \mid C \parallel J \]

- \( x \) and \( y \) are d-connected given \( C \) when variables in \( J \) are subject to intervention
- \( x \) and \( y \) are dependent given \( C \) when variables in \( J \) are subject to intervention
Example with 3 variables: x, y, z

\[ x \perp y \]
Example with 3 variables: $x, y, z$

$x \perp y$

PC-algorithm
Example with 3 variables: $x, y, z$

$x \perp y$

PC-algorithm
Example with 3 variables: $x$, $y$, $z$

$\begin{array}{c}
x \perp y\\
\end{array}$

PC-algorithm

SAT-algorithm

define atoms

\begin{align*}
A & := "x \rightarrow y \in G" \\
B & := "y \rightarrow x \in G" \\
C & := "z \rightarrow x \in G" \\
D & := "z \rightarrow y \in G"
\end{align*}
Example with 3 variables: $x, y, z$

\[ x \perp y \]
SAT-based Causal Discovery

• Formulate the independence constraints in propositional logic

\[ x \perp y \iff \neg A \land \neg B \ldots \]

\[ A = \text{‘}x \rightarrow y \text{ is present’} \]
SAT-based Causal Discovery

- Formulate the independence constraints in propositional logic
  
- Encode the constraints into one formula.

\[ x \perp y \iff \neg A \land \neg B \ldots \]

\[ A = 'x \to y \text{ is present}' \]

\[ \neg A \land \neg B \land \neg (C \land D) \land \neg \ldots \]
SAT-based Causal Discovery

• Formulate the independence constraints in propositional logic
  \[ x \perp y \iff \neg A \land \neg B \ldots \]
  \[ A = 'x \rightarrow y \text{ is present}' \]

• Encode the constraints into one formula.
  \[ \neg A \land \neg B \land \neg (C \land D) \land \neg \ldots \]

• Find satisfying assignments using a SAT-solver
  \[ A = \text{false} \]
  \[ B = \text{false} \iff \]
  ...
SAT-based Causal Discovery

• Formulate the independence constraints in propositional logic

\[ x \perp y \iff \neg A \land \neg B \ldots \]

\[ A = 'x \rightarrow y \text{ is present}' \]

• Encode the constraints into one formula.

\[ \neg A \land \neg B \land \neg(C \land D) \land \neg\ldots \]

• Find satisfying assignments using a SAT-solver

\[ A = false \]

\[ B = false \]

\[ \ldots \]

→ very general setting (allows for cycles and latents) and trivially complete
SAT-based Causal Discovery

• Formulate the independence constraints in propositional logic

\[ x \perp y \iff \neg A \land \neg B \ldots \]

\[ A = ‘x \rightarrow y \text{ is present}’ \]

• Encode the constraints into one formula.

\[ \neg A \land \neg B \land \neg (C \land D) \land \neg \ldots \]

• Find satisfying assignments using a SAT-solver

\[ A = false \]

\[ B = false \iff \]

\[ \ldots \]

➡ very general setting (allows for cycles and latents) and trivially complete

➡ **BUT**: erroneous test results induce conflicting constraints: UNSatisfiable
Conflicts and Errors

- Statistical independence tests produce errors

\[
\begin{align*}
\text{constraint} \\
x \not\perp z \\
y \not\perp z \\
x \perp y \\
x \perp y|z
\end{align*}
\]
Conflicts and Errors

- Statistical independence tests produce errors
  
  ➡️ **Conflict**: no graph can produce the set of constraints

\[
\begin{align*}
\text{constraint} \\
\mathbf{x} \not\perp \mathbf{z} \\
\mathbf{y} \not\perp \mathbf{z} \\
\mathbf{x} \perp \mathbf{y} \\
\mathbf{x} \perp \mathbf{y} | \mathbf{z}
\end{align*}
\]
Conflicts and Errors

- Statistical independence tests produce errors

Conflict: no graph can produce the set of constraints

```
constraint
x \perp z
y \perp z
x \perp y
x \perp y \mid z
```
Conflicts and Errors

- Statistical independence tests produce errors

**Conflict**: no graph can produce the set of constraints

\[
\begin{align*}
\text{constraint} & \\
& x \not\perp z \\
& y \not\perp z \\
& x \not\perp y \\
& x \perp y | z \\
\end{align*}
\]
Conflicts and Errors

- Statistical independence tests produce errors

**Conflict**: no graph can produce the set of constraints

<table>
<thead>
<tr>
<th>Constraint</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x \not\perp z$</td>
<td>3000</td>
</tr>
<tr>
<td>$y \not\perp z$</td>
<td>2500</td>
</tr>
<tr>
<td>$x \perp y$</td>
<td>500</td>
</tr>
<tr>
<td>$x \perp y \mid z$</td>
<td>250</td>
</tr>
</tbody>
</table>

Diagram:

- Nodes: $x$, $y$, $z$
- Edges: $x \rightarrow z$, $y \rightarrow z$
Constraint Satisfaction Approach

• **INPUT:** (in)dependence constraints weighted according to reliability

\[
\min_G \sum_{k : \text{constraint } k \text{ is not satisfied by } G} w(k)
\]

• **OUTPUT:** a graph \( G \) that minimizes the cost
Constraint Satisfaction Approach

• INPUT: (in)dependence constraints weighted according to reliability

\[
\min_G \sum_{k : \text{constraint } k \text{ is not satisfied by } G} w(k)
\]

• OUTPUT: a graph \( G \) that minimizes the cost

What are suitable weights?
Weighting Schemes

- **Constant weights**
  - unit weights for all constraint
Weighting Schemes

• **Constant weights**
  - unit weights for all constraint

• **Hard dependencies**
  - only treat rejections of the null-hypothesis as hard constraints, in line with classical statistics
  - give dependences infinite weight, maximize the independences (unit weight) in light of these dependences
Weighting Schemes

• **Constant weights**
  - unit weights for all constraints

• **Hard dependencies**
  - only treat rejections of the null-hypothesis as hard constraints, in line with classical statistics
  - give dependences infinite weight, maximize the independences (unit weight) in light of these dependences

• **Log weights**
  - obtain the probability of an (in)dependence and weigh it according to the log of the probability
  - Model selection with Bayes rule:

    $\frac{x \not\perp y | C}{P(x|C)P(y|x, C)} \quad \text{vs.} \quad \frac{x \perp y | C}{P(x|C)P(y|C)}$
Weighting Schemes

- **Constant weights**
  - unit weights for all constraints

- **Hard dependencies**
  - only treat rejections of the null-hypothesis as hard constraints, in line with classical statistics
  - give dependences infinite weight, maximize the independences (unit weight) in light of these dependences

- **Log weights**
  - obtain the probability of an (in)dependence and weigh it according to the log of the probability
  - Model selection with Bayes rule:
    \[
    x \not\perp y | C \quad \text{vs.} \quad x \perp y | C
    \]
    \[
    \frac{P(x|C)P(y|x, C)}{P(x|C)P(y|C)}
    \]
  - probabilistic classifier: find G such that if it were true, test results would be optimal in the sense of a proper score
Optimization

- Answer Set Programming (ASP) is a modern declarative programming paradigm
  - solver used: Clingo
  - SAT-solver and branch and bound algorithm
  - finds globally optimal weighted maxSAT solution
Simulation 1: cycles and latents

- ROC of dependences, passive observational data set, 6 observed variables, average degree 2; 500 samples, 200 models, linear Gaussian parameterization
Simulation 2: no cycles, no latents

- cPC returns a fully determined output only 58/200 times at its optimum
Simulation 3: no cycles, but latents

• cFCl only returns unambiguous results 61/200 times at its optimum
Simulation 4: Scalability

- up to 7 variables and only a few data sets for now (9\times10^{18} graphs)
Background Knowledge

\[ x \xrightarrow{w} y \xrightarrow{} z \]

“priors”
Background Knowledge

\[
x \not\perp w \mid | x
\]

"priors"
Background Knowledge

$x \not\perp w \mid \mid x \perp z$

$x \perp z | y > w$

“priors”
Background Knowledge

$x \not\perp w \mid x \perp z$  weight $= 0.8$

"priors"
Background Knowledge

\[ x \mathbin{\approx} \mathbf{w} \mathrel{|} x \succeq z \quad \text{weight} = 0.8 \]

\[ (x > z) \land (x > w) \land (y > z) \land (y > w) \]

“priors”
Background Knowledge

$x \nparallel w \mid x \parallel z \quad \text{weight} = 0.8$

$\begin{array}{c}
x > z \quad \land 
\quad (x > w) \\
\land (y > z) \land (y > w)
\end{array}$

“priors”
Background Knowledge

\[ x \not\perp w \mid x \geq z \quad \text{weight} = 0.8 \]

\[ (x > z) \land (x > w) \]
\[ \land (y > z) \land (y > w) \]

- specific probabilities for each graph
- soft sparsity constraint
- ...
Settings
range(1..5).

1 { u(U): urange(U) } 1.

{ edgel(X,Y) } :- node(X), node(Y).

path(X,Y,1) :- edgel(X,Y).
path(X,Y,L) :- path(X,Z,L-1), edgel(Z,Y), L <= U, u(U).

edgelu(X,Y) :- path(X,Y,L), u(L).

conflu(X,Y) :- path(Z,X,L), path(Z,Y,L), node(X), node(Y), node(Z), X < Y, L < U, u(U).

:- edgeu(X,Y), edgelu(X,Y).
:- no_edgeu(X,Y), edgelu(X,Y).
:- confu(X,Y), edgelu(X,Y).
:- no_confu(X,Y), conflu(X,Y).
Settings

\begin{align*}
\text{range for rate of subsampling} & \quad \text{subsampling rate is unique} \\
\text{def. of edge in graph} & \quad \text{recursive def. of path} \\
\text{def. of edge in subsampled graph} & \quad \text{def. of how confounders arise due to subsampling} \\
\text{constraints on how edges in subsampled graph relate to edges in true graph}
\end{align*}

\begin{align*}
\text{urange}(1..5). \\
1 \{ u(U): \text{urange}(U) \} 1. \\
\{ \text{edge1}(X,Y) \} :- \text{node}(X), \text{node}(Y). \\
\text{path}(X,Y,1) :- \text{edge1}(X,Y). \\
\text{path}(X,Y,L) :- \text{path}(X,Z,L-1), \text{edge1}(Z,Y), L <= U, u(U). \\
\text{edge1u}(X,Y) :- \text{path}(X,Y,L), u(L). \\
\text{conf1u}(X,Y) :- \text{path}(Z,X,L), \text{path}(Z,Y,L), \text{node}(X), \text{node}(Y), \text{node}(Z), X < Y, L < U, u(U). \\
\text{:- edgeu}(X,Y), \text{not edge1u}(X,Y). \\
\text{:- no_edgeu}(X,Y), \text{edge1u}(X,Y). \\
\text{:- confu}(X,Y), \text{not conf1u}(X,Y). \\
\text{:- no_confu}(X,Y), \text{conf1u}(X,Y).
\end{align*}
Runtime comparison

For a graph determined at subsampling rate 2, infer the equivalence class of graphs at the system time scale (1-step)
Output of Causal Search Algorithms

(max) SAT-solver

e etc.
Output of Causal Search Algorithms

(max) SAT-solver

\{ equivalence class? \}

\( x \rightarrow y \rightarrow z \rightarrow w \)

\( x \rightarrow y \rightarrow z \rightarrow w \)

\( x \rightarrow y \rightarrow z \rightarrow w \)

\( x \rightarrow y \rightarrow z \rightarrow w \)

e tc.
Output of Causal Search Algorithms

Query:

(max) SAT-solver
Output of Causal Search Algorithms

Query:
- list the structures in the equivalence class
Output of Causal Search Algorithms

Query:
- list the structures in the equivalence class
- what structural features are determined?
  - edges, confounders
  - ancestral relations
  - pathways
Output of Causal Search Algorithms

Query:
• list the structures in the equivalence class
• what structural features are determined?
  - edges, confounders
  - ancestral relations
  - pathways
• what are the highest scoring equivalence classes?
Output of Causal Search Algorithms

Query:
• list the structures in the equivalence class
• what structural features are determined?
  - edges, confounders
  - ancestral relations
  - pathways
• what are the highest scoring equivalence classes?

Response:
• enumeration of solutions
• “backbone” of the SAT-instance
• …
Computing Causal Effects

\(P(y \mid do(x))\)?
equivalence class? \( P(y \mid do(x)) \) ?
• enumerate each graph in the equivalence class and run the Tian-Shpitser algorithm to determine the causal effect?
equivalence class? \( P(y \mid do(x)) \) ?

- enumerate each graph in the equivalence class and run the Tian-Shpitser algorithm to determine the causal effect?

- Alternative:
equivalence class? \( P(y|do(x)) \) ?

- enumerate each graph in the equivalence class and run the Tian-Shpitser algorithm to determine the causal effect?

- Alternative:

**do-calculus**

Rule 1 (insertion/deletion of observations)

\[ P(y|do(x), z, w) = P(y|do(x), w) \text{ if } Y \perp Z|X, W||X \]

Rule 2 (action/observation exchange)

\[ P(y|do(x), do(z), w) = P(y|do(x), z, w) \text{ if } Y \perp I_Z|X, Z, W||X \]

Rule 3 (insertion/deletion of actions)

\[ P(y|do(x), do(z), w) = P(y|do(x), w) \text{ if } Y \perp I_Z|X, W||X \]
equivalence class? $P(y|do(x))$?

- enumerate each graph in the equivalence class and run the Tian-Shpitser algorithm to determine the causal effect?

- Alternative:

**do-calculus**

Rule 1 (insertion/deletion of observations)

$$P(y|do(x), z, w) = P(y|do(x), w) \text{ if } Y \perp Z|X, W||X$$

Rule 2 (action/observation exchange)

$$P(y|do(x), do(z), w) = P(y|do(x), z, w) \text{ if } Y \perp I_Z|X, Z, W||X$$

Rule 3 (insertion/deletion of actions)

$$P(y|do(x), do(z), w) = P(y|do(x), w) \text{ if } Y \perp I_Z|X, W||X$$
equivlance class? \[ P(y|do(x)) \]?

• enumerate each graph in the equivalence class and run the Tian-Shpitser algorithm to determine the causal effect?

• Alternative:

\textit{do-calculus}

Rule 1 (insertion/deletion of observations)
\[ P(y|do(x), z, w) = P(y|do(x), w) \text{ if } Y \perp Z|X, W||X \]

Rule 2 (action/observation exchange)
\[ P(y|do(x), do(z), w) = P(y|do(x), z, w) \text{ if } Y \perp I_{Z|X, Z, W||X} \]

Rule 3 (insertion/deletion of actions)
\[ P(y|do(x), do(z), w) = P(y|do(x), w) \text{ if } Y \perp I_{Z|X, W||X} \]

⇒ search in the equivalence class over the possible applications of the \textit{do-calculus} rules by \textit{querying} the satisfaction of their conditions
Algorithm for the do-calculus when the graph is unknown

- Determine the equivalence class implicitly using a SAT-solver
- Query one solution graph G
- Run the Tian-Shpitser-algorithm on G to determine whether the causal effect \( P(y|do(w)) \) is determined for G
- If it is, determine which do-calculus rules were applied and record the constraints \( C_1, \ldots, C_n \) that were used
  - Add \( \neg C_1 \lor \ldots \lor \neg C_n \) as a constraint to refine the current equivalence class
- If not, determine the “hedge” \( H \) and add \( \neg H \) to refine the current equivalence class
- Repeat until the equivalence is exhausted
- Return the set of estimates of the causal effect and NA if it cannot be determined in one member of the equivalence class
Comparison of our approach to enumeration
In sum: *do*-calculus using a SAT-solver

- enables computation of the causal effect when the graph structure is underdetermined
In sum: \textit{do}-calculus using a SAT-solver

- enables computation of the causal effect when the graph structure is underdetermined

\rightarrow how should one estimate a causal effect when the equivalence class of causal structures was determined on the basis of a set of conflicted constraints?
In sum: *do*-calculus using a SAT-solver

- enables computation of the causal effect when the graph structure is underdetermined

➡ how should one estimate a causal effect when the equivalence class of causal structures was determined on the basis of a set of conflicted constraints?

- some avenues one can explore with the query-based approach:
  - explore more closely the conditions involved in determining the causal effect
  - find multiple different estimators
  - even though the overall graph structure may not be determinable without resolving conflicts, some causal effects may be
Conclusion

• the use of general purpose SAT-solvers provides an extraordinarily versatile tool for causal discovery

• it opens new avenues for handling background knowledge and the computation of causal effects when the causal structure is underdetermined

• it provides a query based approach in contrast to a representation of an equivalence class of causal structures

• it suggests that current general purpose constraint solvers outperform domain specific approaches
References

• {Hyttinen, Plis, Danks, Eberhardt & Järvisalo} (work in progress). *Causal Discovery from Subsampled Time Series Data by Constraint Optimization.*

Other relevant work that is closely related:


Thank you!