Adding Local Constraints to Bayesian Networks

Mark Crowley[1] Brent Boerlage[2] David Poole[1]

[1]Department of Computer Science
University of British Columbia

[2]Netica Division
Norsys Software Corp.

Canadian Conference of Artificial Intelligence - May 30, 2007

Outline

1. Definition of Problem
   - Constraints Using Conditioning
   - Side Effects
   - Undirected Models
   - Types of Conditioning
2. Proposed Methods for Removing Side Effects
   - Shielding Side Effects
   - Antifactors
   - Antinetworks
   - Antinetwork Solution
3. Comparison to Chain Graphs and Future Work
   - Analysis of Complexity
   - Conclusion
   - Future Work

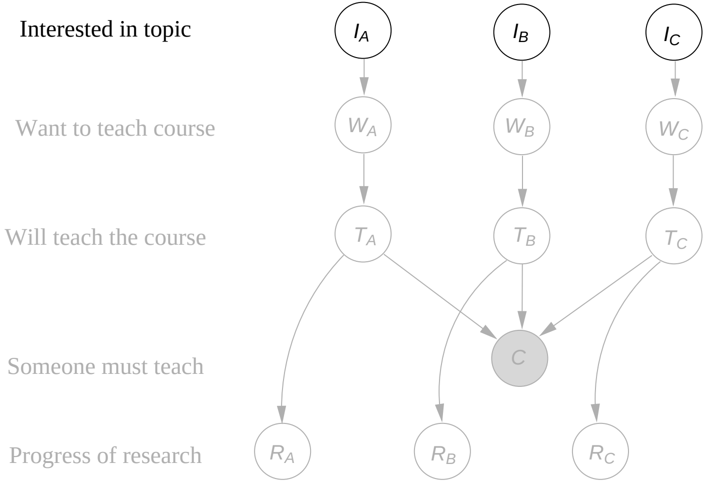
Outline

1. Definition of Problem
   - Constraints Using Conditioning
   - Side Effects
   - Undirected Models
   - Types of Conditioning
2. Proposed Methods for Removing Side Effects
   - Shielding Side Effects
   - Antifactors
   - Antinetworks
   - Antinetwork Solution
3. Comparison to Chain Graphs and Future Work
   - Analysis of Complexity
   - Conclusion
   - Future Work

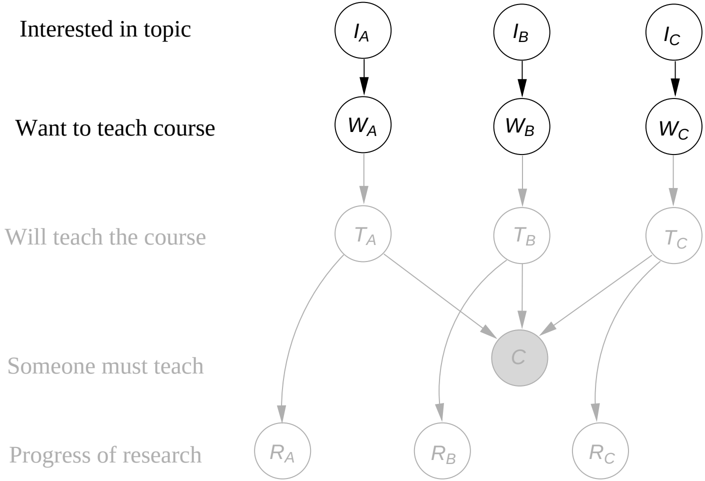
Outline

1. Definition of Problem
   - Constraints Using Conditioning
   - Side Effects
   - Undirected Models
   - Types of Conditioning
2. Proposed Methods for Removing Side Effects
   - Shielding Side Effects
   - Antifactors
   - Antinetworks
   - Antinetwork Solution
3. Comparison to Chain Graphs and Future Work
   - Analysis of Complexity
   - Conclusion
   - Future Work

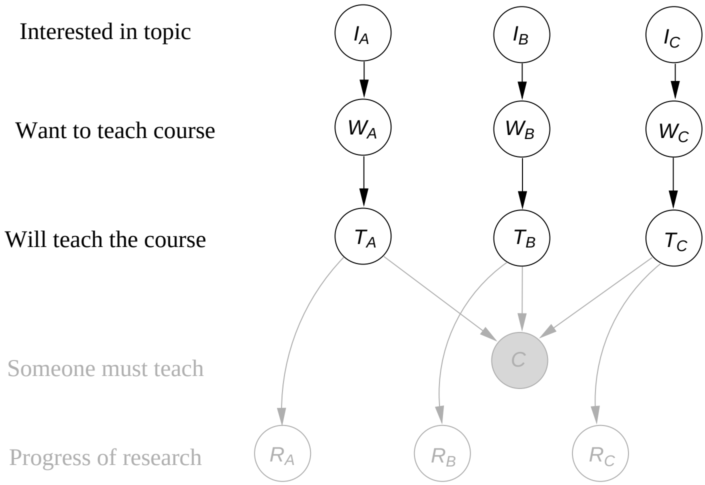
Constraints are modelled using conditioned nodes.



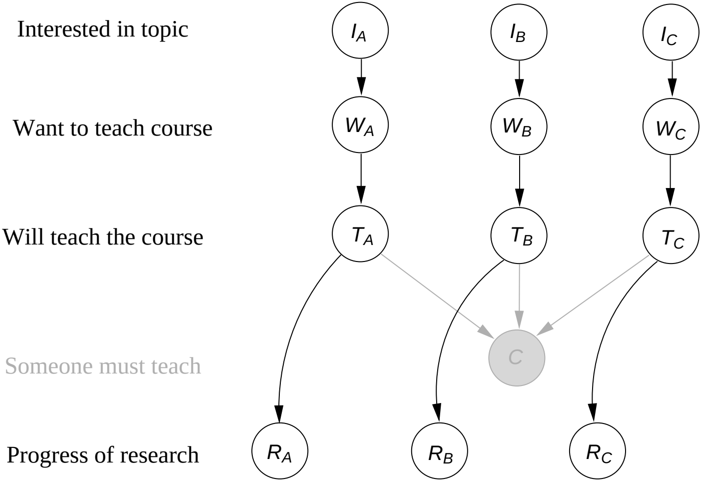
Constraints are modelled using conditioned nodes.



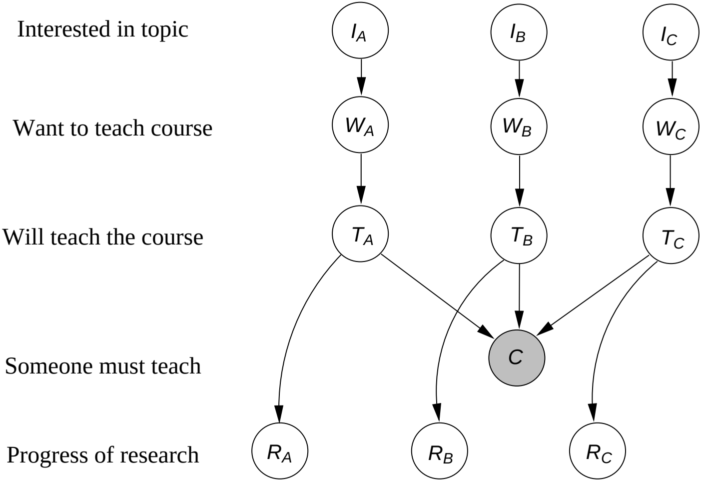
Constraints are modelled using conditioned nodes.



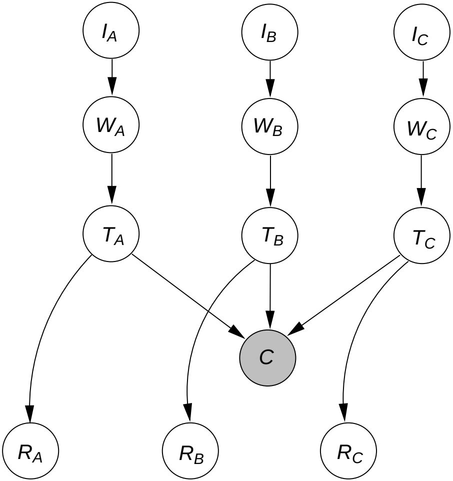
Constraints are modelled using conditioned nodes.



Constraints are modelled using conditioned nodes.



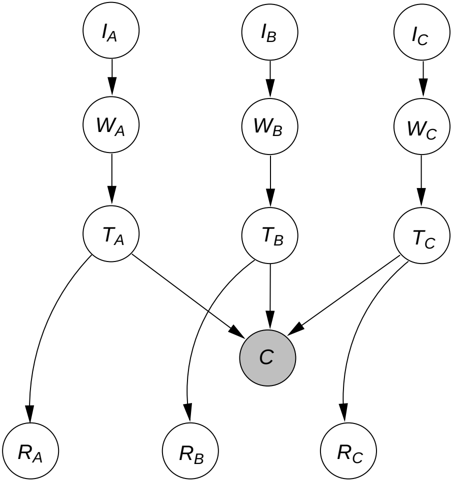
Conditioning has Side Effects



Conditioning has Side Effects

$p(I_A = \text{true}) = .7$

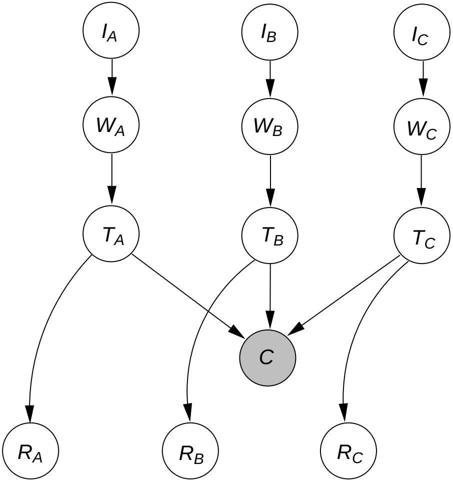
$p(I_A = \text{true}|C) \neq .7$



Conditioning has Side Effects

$p(I_A = \text{true}) = .7$

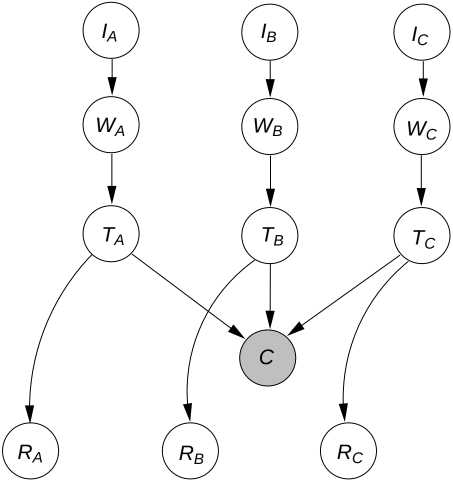
$p(I_A = \text{true}|C) \neq .7$



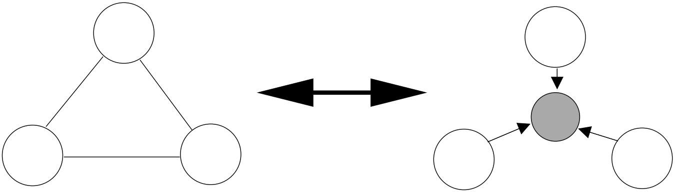
Conditioning has Side Effects

$p(I_A = \text{true}) = .7$

$p(I_A = \text{true}|C) \neq .7$



Another Approach



Another Approach

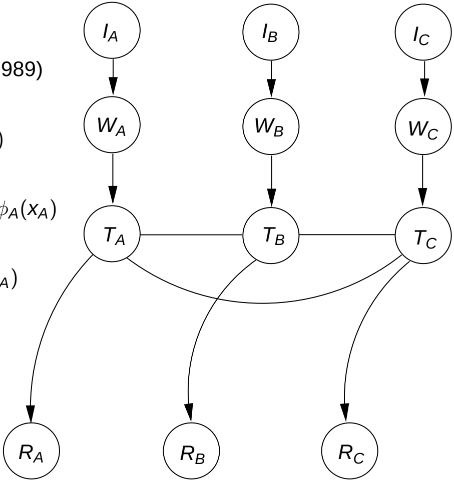
Chain Graphs

(Lauritzen and Wermuth, 1989)

$$p(x_{\mathcal{V}}) = \prod_{\tau \in \mathcal{T}} p(x_\tau | x_{pa(\tau)})$$

$$p(x_\tau | x_{pa(\tau)}) = \frac{1}{Z(x_\tau)} \prod_{A \in \mathcal{A}(\tau)} \phi_A(x_A)$$

$$Z(x_\tau) = \sum_{x_\tau} \prod_{A \in \mathcal{A}(\tau)} \phi_A(x_A)$$



There are three types of conditioning in Bayes nets

Observation Conditioning Conditioned value is new information that rules out incompatible possible worlds. The most common type.

Intervention Conditioning Value is set arbitrarily from outside the model, cannot be used for inference about ancestors. Used for decision variables.

Constraint Conditioning A node's value is set as part of the model definition to induce some distribution amongst its parents. This type is the focus of our research.

There are three types of conditioning in Bayes nets

Observation Conditioning Conditioned value is new information that rules out incompatible possible worlds. The most common type.

Intervention Conditioning Value is set arbitrarily from outside the model, cannot be used for inference about ancestors. Used for decision variables.

Constraint Conditioning A node's value is set as part of the model definition to induce some distribution amongst its parents. This type is the focus of our research.

There are three types of conditioning in Bayes nets

Observation Conditioning Conditioned value is new information that rules out incompatible possible worlds. The most common type.

Intervention Conditioning Value is set arbitrarily from outside the model, cannot be used for inference about ancestors. Used for decision variables.

Constraint Conditioning A node's value is set as part of the model definition to induce some distribution amongst its parents. This type is the focus of our research.

We Use a Conditioned Node to Block Side Effects

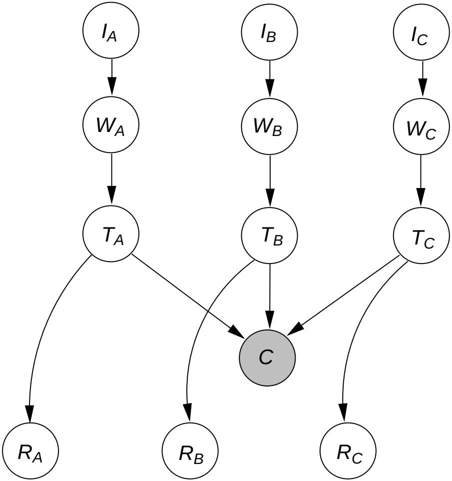
Shielding

$$p(W_A, W_B, W_C | c, \hat{c}) = p(W_A, W_B, W_C)$$

$$p(W_A | I_A, c, \hat{c}) = p(W_A | I_A)$$

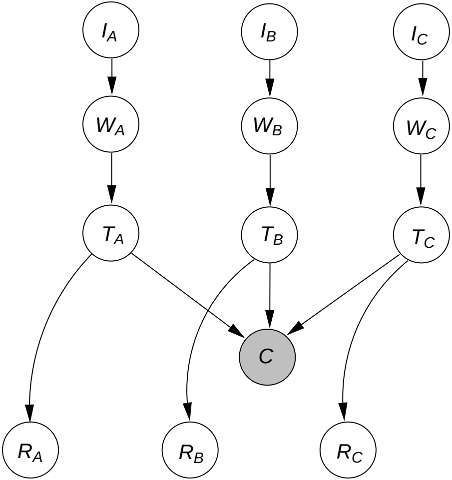
$$p(W_B | I_B, c, \hat{c}) = p(W_B | I_B)$$

$$p(W_C | I_C, c, \hat{c}) = p(W_C | I_C)$$



Shielding with Antifactors

So what is an antifactor?

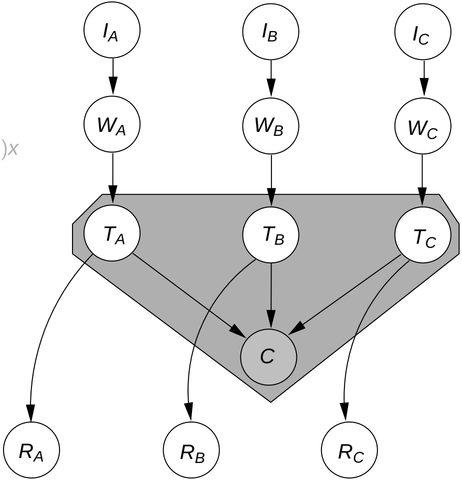


Shielding with Antifactors

So what is an antifactor?

$$p(\hat{c}|W_A, W_B, W_C) = \frac{1}{Z}\frac{1}{f_{T_{ABC}}(W_A, W_B, W_C)}$$

$$f_{T_{ABC}}(W_A, W_B, W_C) = \sum_{\mathbf{T}} p(c|T_A, T_B, T_C) x$$
$$p(T_A|W_A)p(T_B|W_B)p(T_C|W_C)$$

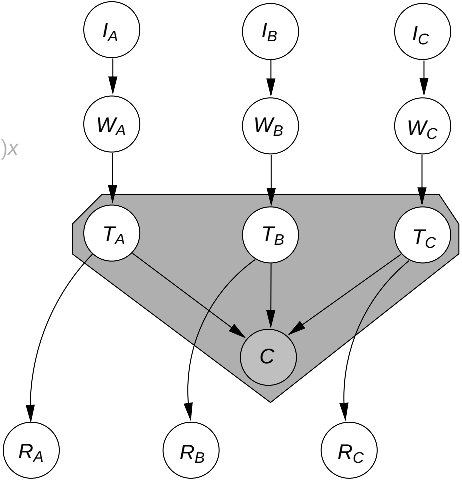


Shielding with Antifactors

So what is an antifactor?

$$p(\hat{c}|W_A, W_B, W_C) = \frac{1}{Z}\frac{1}{f_{T_{ABC}}(W_A,W_B,W_C)}$$

$$f_{T_{ABC}}(W_A, W_B, W_C) = \sum_{\mathbf{T}} p(c|T_A, T_B, T_C)x$$
$$p(T_A|W_A)p(T_B|W_B)p(T_C|W_C)$$

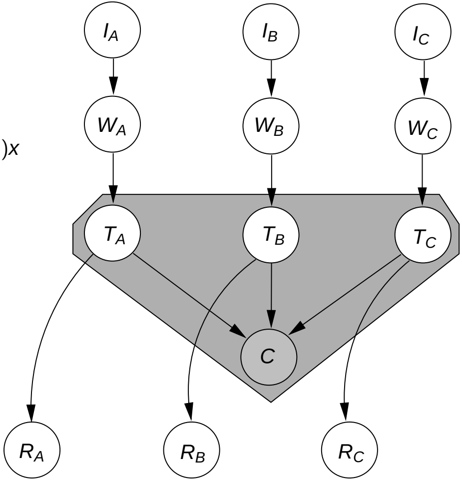


Shielding with Antifactors

So what is an antifactor?

$$p(\hat{c}|W_A, W_B, W_C) = \frac{1}{Z}\frac{1}{f_{T_{ABC}}(W_A, W_B, W_C)}$$

$$f_{T_{ABC}}(W_A, W_B, W_C) = \sum_{\mathbf{T}} p(c|T_A, T_B, T_C)x$$
$$p(T_A|W_A)p(T_B|W_B)p(T_C|W_C)$$

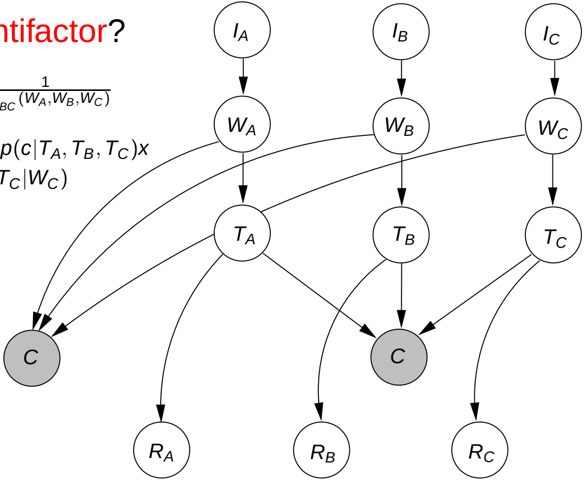


Shielding with Antifactors

So what is an antifactor?

$$p(\hat{c}|W_A, W_B, W_C) = \frac{1}{Z}\frac{1}{f_{T_{ABC}}(W_A,W_B,W_C)}$$

$$f_{T_{ABC}}(W_A, W_B, W_C) = \sum_{\mathbf{T}} p(c|T_A, T_B, T_C)x$$
$$p(T_A|W_A)p(T_B|W_B)p(T_C|W_C)$$

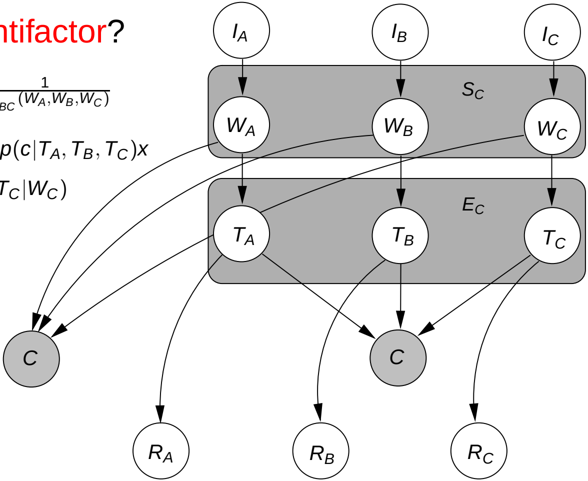


Shielding with Antifactors

So what is an antifactor?

$$p(\hat{c}|W_A, W_B, W_C) = \frac{1}{Z}\frac{1}{f_{T_{ABC}}(W_A,W_B,W_C)}$$

$$f_{T_{ABC}}(W_A, W_B, W_C) = \sum_{\mathbf{T}} p(c|T_A, T_B, T_C)x$$
$$p(T_A|W_A)p(T_B|W_B)p(T_C|W_C)$$



Pro's and Con's of Antifactors

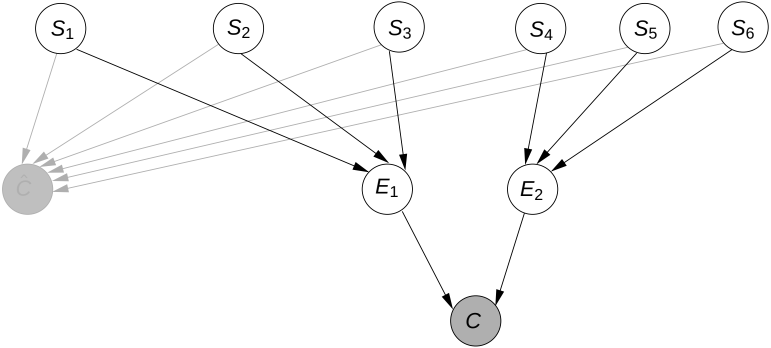
General Antifactors can also be defined that allow multiple c-nodes to have overlapping sets of parents.

PRO An antifactor solution always exists and is easy to compute.

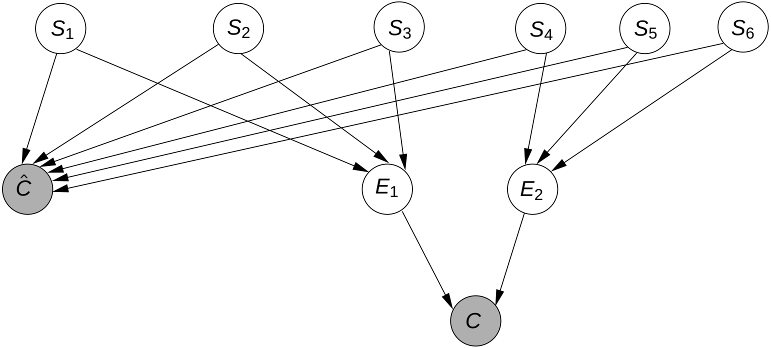
- exception: when one instance of nodes in S_C has a zero probability given the summed out affected network

CON Network with antifactor added to it could be much less efficient during inference.

Antifactors can lead to complexity blowup

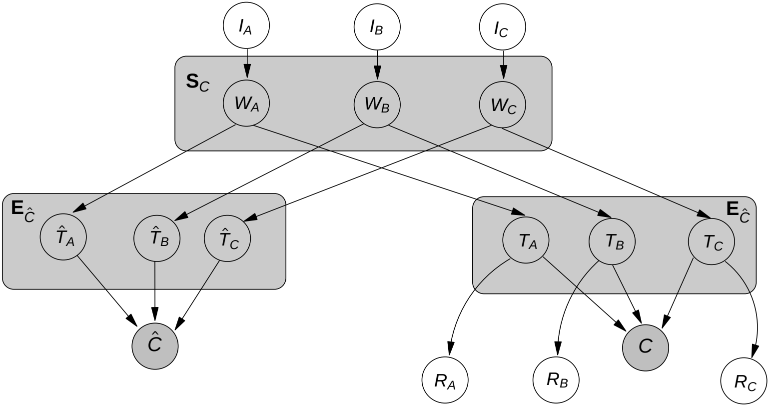


Antifactors can lead to complexity blowup



Shielding with Antinetworks

An antinetwork is a set of nodes, with a conditioned node, that sums out to the antifactor but maintains a conditional structure.



Definition of Antinetwork Solution

A system of nonlinear equations defines the solution for an antinetwork that implements shielding:

$$
\begin{aligned}
\mathit{Let}\ \pi &= f_{\mathbf{E}_{\hat{c}}}(\mathbf{S}_c) = \frac{1}{f_{\mathbf{E}_c}(\mathbf{S}_c)} \\
\pi &= \sum_{x \in \mathbf{E}_{\hat{c}}} p(\hat{c}|\mathbf{E}_{\hat{c}}) \prod_{\hat{e} \in x} p(\hat{e}|\mathbf{S}_{\hat{c}}) \\
0 &= \sum_{x \in \mathbf{E}_{\hat{c}}} \gamma_{\hat{c}} \prod_{\hat{e} \in x} (\psi_{\hat{e}})^{(\hat{e}=\mathbf{t})} (1 - \psi_{\hat{e}})^{(\hat{e}=\mathbf{f})} - \pi
\end{aligned}
$$

We call this system $g_s(X)$ for each instance $s \in \mathbf{S}_C$ and x being an assignment to all variables in $\mathbf{E}_{\hat{c}}$.

Existence and Discovery

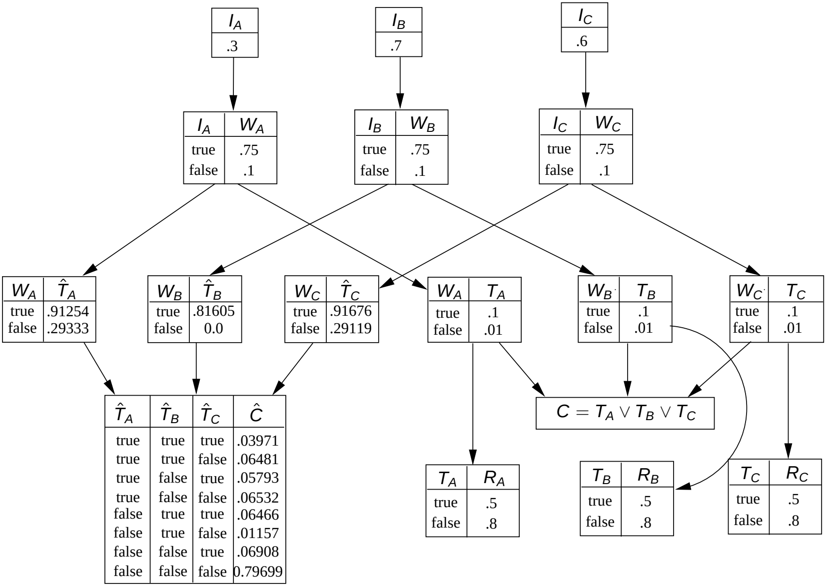
Existence

- $X_0 : x = \text{false for all } x \in X_0 \qquad X_1 : x = \text{true for all } x \in X_1$
- $g_s(X_0) = -\pi \qquad g_s(X_1) = 1 - \pi$
- $g_s(X_0) \leq 0 \leq g_s(X_1) \qquad \text{for all } s \in \mathbf{S}_C$
- for all $s \in \mathbf{S}_C$ there is an X_s such that $g_s(X_s) = 0$
- simultaneous solution not yet proven to always exist

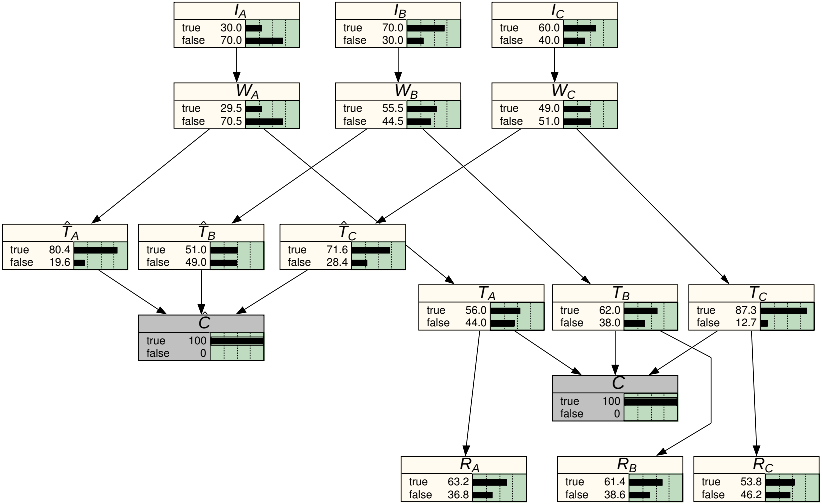
Discovery Use constrained optimization:

- objective function $\sum_{s \in \mathbf{S}_C} g_s(X)$
- constraints $g_s(X) = 0$ for all $s \in \mathbf{S}_C$
- start pointing X_1

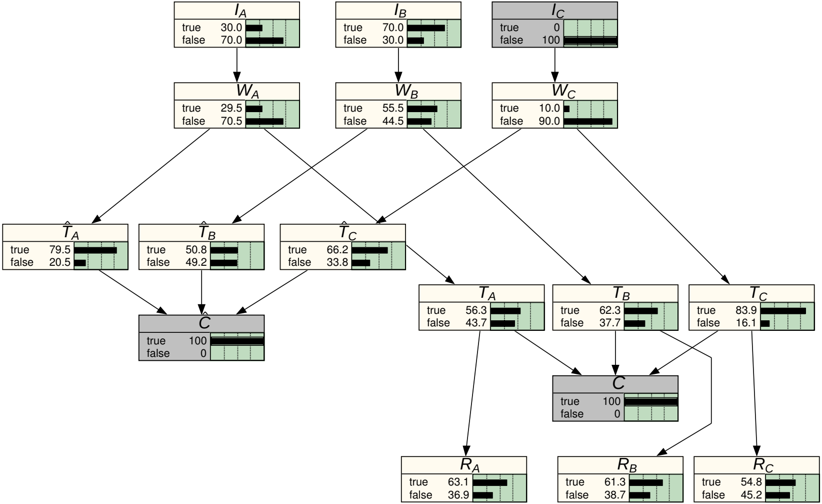
Example of a Solved Antinetwork



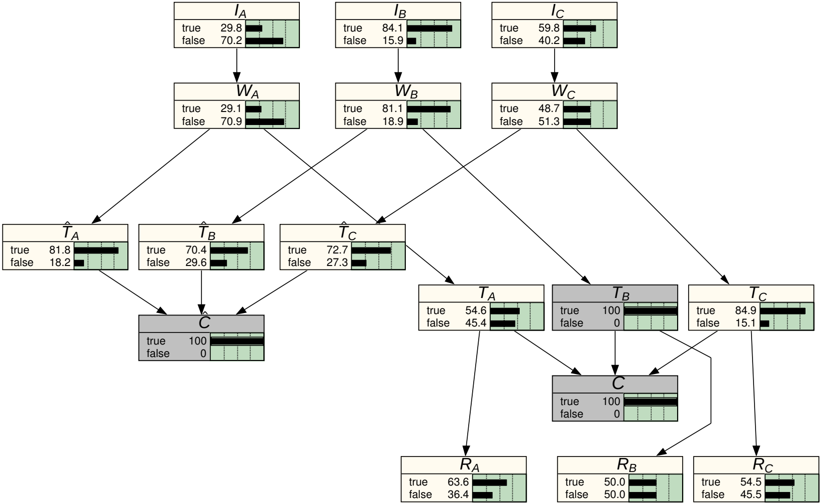
Example of a Solved Antinetwork



Example of a Solved Antinetwork



Example of a Solved Antinetwork



SBNs can be more efficient than CGs

Each node has size D and each *e-node* has m parents.

$$\begin{aligned} CG &= D^{|E_C|+|S_C|} \\ &= D^{|E_C|+m|E_C|} \qquad \text{since } |S_C| = m \times |E_C| \\ &= D^{|E_C|(m+1)} \end{aligned}$$

$$\begin{aligned} SBN_{\text{antifactor}} &= D^{|E_C|+|S_C|} + D^{|S_C|} + D^{|E_C|} \\ SBN_{\text{antinetwork}} &= D^{2|E_C|} + |E_C|D^{2+m} + 2D^{|E_C|} \end{aligned}$$

$$SBN_{\text{antinetwork}} < CG \quad \text{when both } |E_C| \geq 2 \text{ and } m \geq 2.$$

Summary

- SBNs express an interesting set of distributions and can do so efficiently for some networks.
- Shielding allows constraints/joint distributions to be expressed simply without side effects.
- Conditioning is a useful concept for unifying discussion of directed, undirected and mixed modelling tools.

Possibilities for Future Work

- Find solutions for more complex networks with overlapping affected sets.
- Find sets of networks that always have solutions if any.
- Compare complexity of larger SBNs to CGs. Perform quantitative comparisons.
- Create precompiler to convert a BN to an SBN using antifactors, antinetworks or both in combination to be used with standard BN software.
- Look for specific distributions that can be shielded efficiently without an antinetwork.

Thank You

Questions?