Causal Inference Despite Limited Global Confounding via Mixture Models

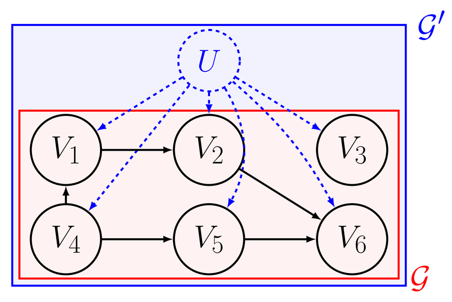
Spencer Gordon [1] **Bijan Mazaheri** [1] Yuval Rabani [2] Leonard Schulman [1]

[1]California Institute of Technology [2]The Hebrew University of Jerusalem

PAPER

Problem

Setup



- A *Bayesian Network* is a directed acyclic graph (DAG) $\mathcal{G} = (\mathbf{V}, \mathbf{E})$.
- A *Bayesian Network Distribution* on n random variables $\mathbf{V}$ is Markovian on Bayesian Network $\mathcal{G}$.
- A k-*mixture* of such distributions (k-MixBND) is represented using one additional vertex U with $\mathsf{CH}(U) = \mathbf{V}$.

Task

Knowns:

- The *marginal* probability distribution on $\mathbf{V}$:

$$\Pr(\mathbf{V}) = \sum_{u \in U} \Pr(u) \Pr(\mathbf{V} \mid u)$$

Unknowns:

- The probability distribution on U, i.e. $\Pr(u)$ for $u \in \{1, \ldots, k\}$.
- The *within source* probability distribution $\Pr(\mathbf{V} \mid u) = \mathcal{P}_u(\mathbf{V})$ for $u \in \{1, \ldots, k\}$.
  - $\mathcal{P}_u(\mathbf{V})$ is a Bayesian network distribution, so it suffices to find $\mathcal{P}_u(V \mid \mathsf{pa}(V))$ for all $V \in \mathbf{V}$, and assignments $\mathsf{pa}(V)$ to $\mathsf{PA}(V)$.

Motivation

- Setting emerges when combining multiple...
  - -Populations
  - -Environments
  - -Datasets

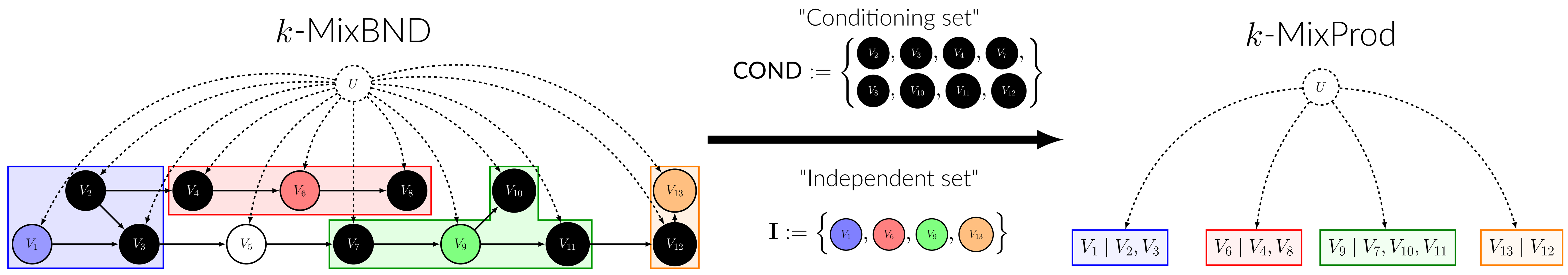
  $k <$ observed support!
- Graphically, the causal relationships are considered unidentifiable.
- Interventional probabilities *can* be calculated using the *within source* distributions, $\Pr(\mathbf{V} \mid u)$.
- Solving the mixture model allows for a degree of **deconfounding**.

Previous Work

Using 3-independent variabls and larger alphabets (linear in k):

- E. S. Allman, 2009 use algebraic methods to exploit within-source independence.
- Anandkumar, Hsu, and Kakade, 2012 follows a similar strategy using tensor decomposition.
- Wang and Blei, 2019 introduced deconfounders using multiple causes.
  - Criticized in Ogburn, Shpitser, and Tchetgen, 2019 and D'Amour, 2019
  - Criticism is linked to the difference between **learning** parameters that generate similar statistics and **identification** of the true parameters.

Strategy



Reduction to k-MixProd

- The *conditional* probability distribution $\mathbf{I} \mid$ COND is an instance of k-MixProd on the **independent set $\mathbf{I}$**.
- We will create many such instances (called "runs") and stitch together the results. These runs will need to:
  1. **Cover** all of the variables and assignments to their parents .
  2. Be **alignable** with eachother so their results can be synthesized.
- **Runtime of k-MixBND**: $n2^{\Delta^2}$ executions of k-MixProd where Δ is a bound on the degree of $\mathcal{G}$.
- **Runtime of k-MixProd**: $2^{\mathcal{O}(k^2)}n^{\mathcal{O}(k)}$ in Gordon et al., 2021 and $2^{\mathcal{O}(k\log(k))}n^{\mathcal{O}(k)}$ in upcoming work.

Difficulties

- k-MixProd is symmetric to permutations in the labels of U.
  Solution: Alignment variables
- Conditioning (via post-selection) limits the variables whose information we have access to. How can we ensure we have obtained all of the necessary information?
  Solution: "Good sets of runs" and alignment spanning trees
- The parameters returned from k-MixProd instances are of the form $\mathcal{P}_u(V \mid \mathsf{mb}(V))$ -- we want them of the form $\mathcal{P}_u(V \mid \mathsf{pa}(V))$ (which is standard for Bayesian networks).
  Solution: Bayesian unzipping

Assumptions

1. We have access to a k-MixProd oracle requiring $O(k)$ variables that are independent within each source.
2. The observable variables in our BND are binary and discrete.
   - Extensions exist for larger alphabets.
3. The mixture is supported on $\leq k$ sources.
4. The underlying Bayesian DAG is sufficiently sparse.
   - The algorithm works if $n \geq (\Delta+1)^4 N_{\text{mp}}$
   - Milder requirements exist for specific graphs.
5. The resulting product mixtures are non-degenerate.
6. The DAG structure $\mathcal{G}$ is known.
   - Upcoming work on how to do causal discovery to learn $\mathcal{G}$.

Alignment of Source Labels

Alignment Variables

Run a:

$U^{(a)}$	$\mathcal{P}_{u^{(a)}}(X_1)$	$\mathcal{P}_{u^{(a)}}(X_2)$
0	.7	.4
1	.3	.6

Run b:

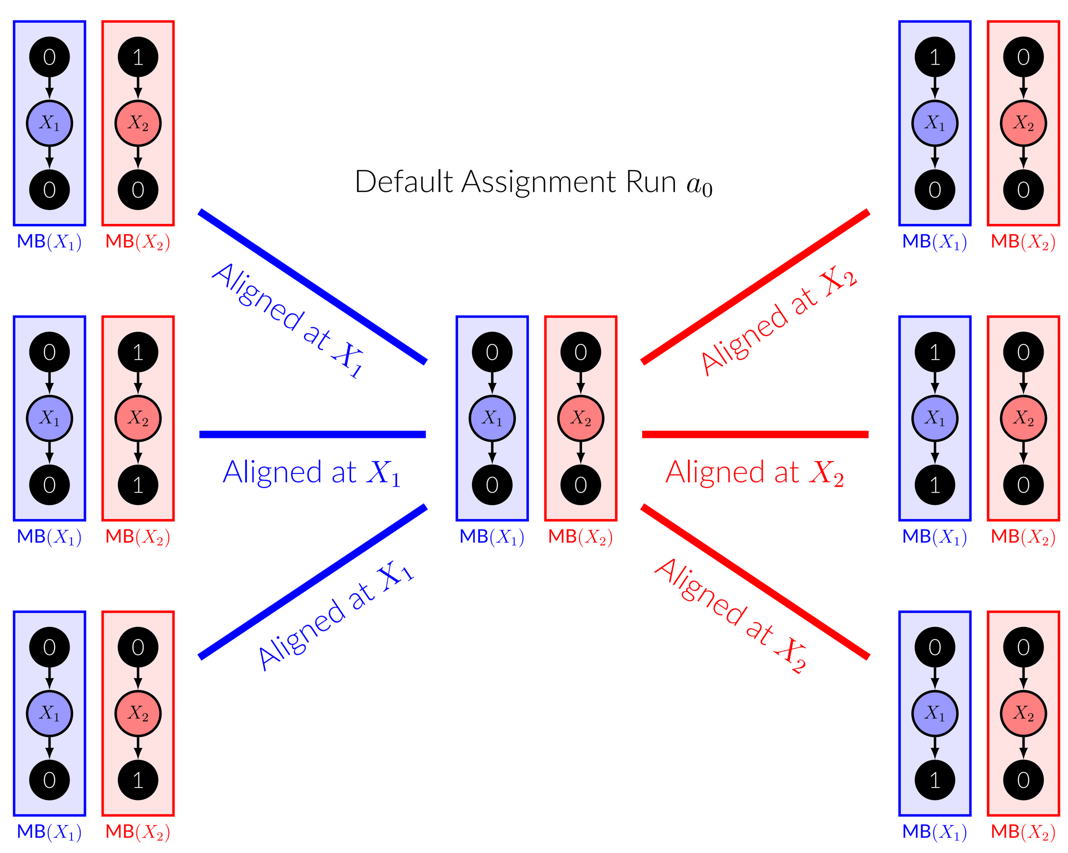
$U^{(b)}$	$\mathcal{P}_{u^{(b)}}(X_1)$	$\mathcal{P}_{u^{(b)}}(X_3)$
0	.3	.2
1	.7	.8

Combined:

U	$\mathcal{P}_u(X_1)$	$\mathcal{P}_u(X_2)$	$\mathcal{P}_u(X_3)$
0	.7	.4	.8
1	.3	.6	.2

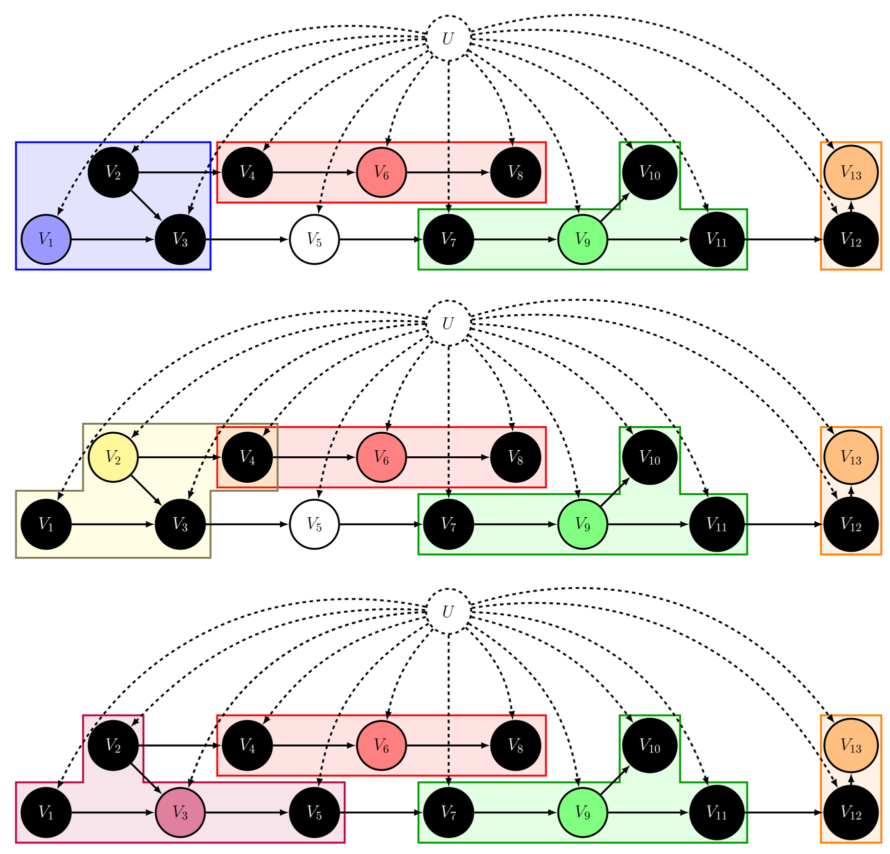
- Runs a and b are aligned at X_1, allowing $U^{(a)}$ and $U^{(b)}$ to be aligned.
- To ensure we have a "good collection of runs," any two runs must be alignable via a chain of alignment variables - i.e. we must have an **alignment spanning tree**.

Changing the Conditioned Values



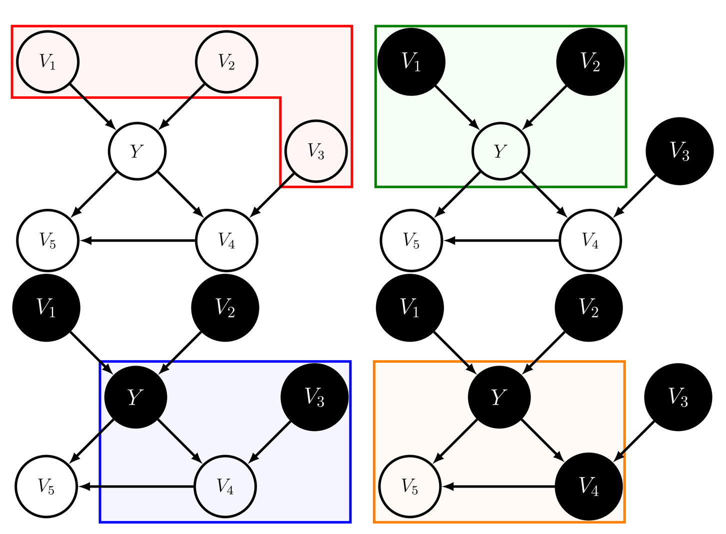
Change the assignments to Markov boundaries while leaving at least one assignment the same.

Varying the Independent Set



Disjoint Markov boundaries ensure that a single variable can be swapped without requiring others in the independent set to be conditioned on.

Bayesian Unzipping: $\Pr(V \mid \mathsf{MB}(V)) \to \Pr(V \mid \mathsf{PA}(V))$



Let y^0 denote ``$y = 0$'' and $y^1$ denote ``$y = 1$''.

$$\mathcal{P}_u(y^1 \mid \mathsf{mb}(Y)) = \frac{\mathcal{P}_u(y^1, \mathsf{mb}(Y))}{\mathcal{P}_u(y^1, \mathsf{mb}(Y)) + \mathcal{P}_u(y^0, \mathsf{mb}(Y))}$$

We apply the standard factoring,

$$\mathcal{P}_u(y, \mathsf{mb}(Y)) = \mathcal{P}_u(v_1, v_2, v_3)\mathcal{P}_u(y \mid v_1, v_2)\mathcal{P}_u(v_4 \mid y, v_3)\mathcal{P}_u(v_5 \mid y, v_4),$$

to all three terms.

$$\mathcal{P}_u(y^1 \mid \mathsf{mb}(Y)) = \frac{\mathcal{P}_u(y^1 \mid v_1, v_2)\mathcal{P}_u(v_4 \mid y^1, v_3)\mathcal{P}_u(v_5 \mid y^1, v_4)}{\mathcal{P}_u(y^1 \mid v_1, v_2)\mathcal{P}_u(v_4 \mid y^1, v_3)\mathcal{P}_u(v_5 \mid y^1, v_4)) + \mathcal{P}_u(y^0 \mid v_1, v_2)\mathcal{P}_u(v_4 \mid y^0, v_3)\mathcal{P}_u(v_5 \mid y^0, v_4)}$$

- $\mathcal{P}_u(v_1, v_2, v_3)$ appears in both numerator and denominator, so it cancels out.
- If we traverse in **reverse topological order**, then $\mathcal{P}_u(v_4 \mid y, v_3)$ and $\mathcal{P}_u(v_5 \mid y, v_4)$ terms are previously calculated for both $y \in \{y^0, y^1\}$.
- $\mathcal{P}_u(y^0 \mid v_1, v_2) + \mathcal{P}_u(y^1 \mid v_1, v_2) = 1$, so we can solve for green terms.
- Iterating this process incurs stability costs proportional to the depth of the graph.
  - This can be avoided by not conditioning on the children of the deepest variables in the independent set.

References

Anandkumar, A., D. J. Hsu, and S. M. Kakade (2012). ``A Method of Moments for Mixture Models and Hidden Markov Models''. In: *Proc. 25th Ann. Conf. on Learning Theory - COLT*. Vol. 23. JMLR Proceedings, pp. 33.1–33.34. URL: `http://proceedings.mlr.press/v23/anandkumar12/anandkumar12.pdf`.

D'Amour, Alexander (2019). ``Comment: Reflections on the deconfounder''. In: *Journal of the American Statistical Association* 114.528, pp. 1597–1601.

E. S. Allman C. Matias, J. A. Rhodes (2009). ``Identifiability of parameters in latent structure models with many observed variables''. In: *Ann. Statist.* 37.6A, pp. 3099–3132. DOI: `10.1214/09-AOS689`.

Gordon, S. L. et al. (2021). ``Source Identification for Mixtures of Product Distributions''. In: *Proc. 34th Ann. Conf. on Learning Theory - COLT*. Vol. 134. Proc. Machine Learning Research. PMLR, pp. 2193–2216. URL: `http://proceedings.mlr.press/v134/gordon21a.html`.

Ogburn, Elizabeth L, Ilya Shpitser, and Eric J Tchetgen Tchetgen (2019). ``Comment on "blessings of multiple causes"''. In: *Journal of the American Statistical Association* 114.528, pp. 1611–1615.

Wang, Y. and D. M. Blei (2019). ``The Blessings of Multiple Causes''. In: *Journal of the American Statistical Association* 114.528, pp. 1574–1596. DOI: `10.1080/01621459.2019.1686987`.