Supplementary material for: Identifying Counterfactual Queries with the R package cfid

Santtu Tikka

Appendix: counterfactual graphs

We restate here the make-cg algorithm and the associated Lemmas that are used to construct counterfactual graphs from parallel worlds graphs. Lemma 1 is used to characterize conditions where two counterfactual variables in fact represent the same random variable. Lemma 2 shows that under the conditions of Lemma 1 such variables can be merged into a single random variable. For more details, see (Shpitser and Pearl, 2008).

Lemma 1 (Lemma 24 of Shpitser and Pearl (2008)). *Let M be a model inducing G containing variables α, β with the following properties:*

- *α and β have the same domain of values.*
- *There is a bijection f from $Pa(\alpha)$ to $Pa(\beta)$ such that a parent γ and $f(\gamma)$ have the same domain of values.*
- *The functional mechanisms of α and β are the same (except whenever the function for α uses the parent γ, the corresponding function for β uses $f(\gamma)$.*

Assume as observable variable set $\mathbf{Z}$ was observed to attain values $\mathbf{z}$ in $M_{\mathbf{x}}$, the submodel obtained from M by forcing another observable variable set $\mathbf{X}$ to attain values $\mathbf{x}$. Assume further that for each $\gamma \in Pa(\alpha)$, either $f(\gamma) = \gamma$, or γ and $f(\gamma)$ attain the same values (whether by observation or intervention). Then α and β are the same random variable in $M_{\mathbf{x}}$ with observations $\mathbf{z}$.

Lemma 2 (Lemma 25 of Shpitser and Pearl (2008)). *Let $M_{\mathbf{x}}$ be a submodel derived from M with set $\mathbf{Z}$ observed to attain values $\mathbf{z}$, such that Lemma 1 holds for α, β. Let M' be a causal model obtained from M by merging α, β into a new node ω, which inherits all parents and the functional mechanism of α. All children of α, β in M' become children of ω. Then $M_{\mathbf{x}}$, $M'_{\mathbf{x}}$ agree on any distribution consistent with $\mathbf{z}$ being observed.*

The previous two Lemmas are leveraged in the make-cg algorithm as shown in Figure 1. In the implementation of this algorithm, equivalence classes of worlds for each variable are initialized such that each world is the only member of its equivalence class at the beginning. During the iteration of step 3, we update these equivalence classes such that when two instances of the same original variable are found to be the same variable, we combine the equivalence classes of the worlds that the two instances α and β of this variable belong to. When applying Lemma 1, it is not necessary to check the values of the unobserved parents for equality, because unobserved parents are shared between worlds, and thus their values will always be equal when two instances of the same variable are compared.

In the implementation, all graph modifications are carried out at once, because it is not necessary to modify the graph when determining which variable pairs are equivalent. In other words, in the implementation of step 3, we only iterate step 3.3 while dynamically updating a list of variable merges which are all carried out at once before step 4 takes place (i.e., all modifications specified at steps 3.1 and 3.2). This dynamic approach also allows us to skip some redundant comparisons. For example, say that variables A and B have been found to be equivalent, now we only need to compare either A or B to a third variable C.

function **make-cg**(G, γ)
INPUT: G a causal diagram, γ a conjunction of counterfactual events
OUTPUT: A counterfactual graph G_γ, and either a set of events γ' s.t. $P(\gamma') = P(\gamma)$ or **INCONSISTENT**

1. Construct a submodel graph $G_{\mathbf{X}_i}$ for each action $do(\mathbf{x}_i)$ mentioned in γ. Construct the parallel worlds graph G' by having all such submodel graphs share their corresponding U nodes.
2. Let π be a topological ordering of nodes in G', let $\gamma' := \gamma$.
3. Apply Lemmas 1 and 2, in order π, to each observable node pair α, β derived from the same variable in G. For each α, β that are the same do:
   - 3.1 Let G' be modified as specified in Lemma 2.
   - 3.2 Modify γ' by renaming all occurrences of β to α.
   - 3.3 If $\text{val}(\alpha) \neq \text{val}(\beta)$, return ($G'$, **INCONSISTENT**).
4. Return $({G'}_{An(\gamma')}, \gamma')$, where $An(\gamma')$ is the set of nodes in G' ancestral to nodes corresponding to variables mentioned in γ'.

Figure 1: An algorithm for constructing counterfactual graphs.

References

I. Shpitser and J. Pearl. Complete identification methods for the causal hierarchy. *Journal of Machine Learning Research*, 9(64):1941–1979, 2008.