Controlling Selection Bias in Causal Inference (Supplementary Material)

Elias Bareinboim
Cognitive Systems Laboratory
Department of Computer Science
University of California, Los Angeles
Los Angeles, CA. 90095
eb@cs.ucla.edu

Judea Pearl
Cognitive Systems Laboratory
Department of Computer Science
University of California, Los Angeles
Los Angeles, CA. 90095
judea@cs.ucla.edu

Supplementary Material – Proofs

Theorem 1

(if part) Our target quantity is $OR(X, Y \mid \mathbf{C})$ and given that $\mathbf{Z}$ is OR-admissible relative to $(X, Y, \mathbf{C})$, Corollary 2 permits us to add $\mathbf{Z}$ and rewrite it as $OR(X, Y \mid \mathbf{C}, \mathbf{Z})$. Given that the first condition of the theorem holds, Corollary 1 implies $OR(X, Y \mid \mathbf{C}, \mathbf{Z}) = OR(X, Y \mid \mathbf{C}, \mathbf{Z}, S = 1)$. This establishes G-recoverability since the r.h.s. is estimable from the available s-biased data.

(only if part) If the conditions of the theorem cannot be satisfied, then $OR(X, Y \mid \mathbf{C})$ is not G-recoverable, that is, there exist two distributions P_1, P_2 compatible with G such that they agree in the probability under selection, $P_1(\mathbf{V} \setminus \{S\} \mid S = 1) = P_2(\mathbf{V} \setminus \{S\} \mid S = 1)$, and disagree in the odds ratio, $OR_1(X, Y \mid \mathbf{C}) \neq OR_2(X, Y \mid \mathbf{C})$. We first consider the case when $\mathbf{C} = \{\}$, and we will construct two such distributions. Let P_1 be compatible with the graph $G_1 = G$, and P_2 with the subgraph G_2 where all edges pointing to S are removed. Both are compatible with G, since compatibility with a subgraph assures compatibility with the graph itself (Pearl, 1988). Notice that P_2 harbors an additional independence $(\mathbf{V} \setminus \{S\} \perp\!\!\!\perp S)_{P_2}$. By construction $P_1(X, Y \mid S = 1) = P_2(X, Y \mid S = 1)$, but since

$$P_2(X, Y|S = 1) = P_2(X, Y),$$

we have:

$$P_1(X, Y|S = 1) = P_2(X, Y)$$

We can then simplify OR_2 rewriting it as follows

$$OR_2 = \frac{P_1(X, Y, S = 1)P_1(\overline{X}, \overline{Y}, S = 1)}{P_1(\overline{X}, Y, S = 1)P_1(X, \overline{Y}, S = 1)}, \tag{1}$$

and similarly for OR_1,

$$OR_1 = \frac{P_1(X, Y)P_1(\overline{X}, \overline{Y})}{P_1(\overline{X}, Y)P_1(X, \overline{Y})} \tag{2}$$

We want to show that it is possible to produce a parametrization of P_1 in such way that $OR_1(X, Y) \neq OR_2(X, Y)$. First, let us consider the class of Markovian models. Accordingly, P_1 can be parametrized through its factors in the Markov decomposition $P_1(S = 1 \mid \mathbf{PA_s}), P_1(X \mid \mathbf{PA_x}), \ldots$, or more generally, $P_1(V_i \mid \mathbf{PA_i})$ for each family in the graph. This choice of parameters induces a valid parameterization for P_2 as well. Firstly, let us consider the case in which condition 1 of the theorem fails, i.e., $\{X, Y\}$ are not separable from S. Thus, eq. (1) can be rewritten using the identity $P_1(X, Y, S = 1) = P_1(S = 1 \mid X, Y)P_1(X, Y)$, yielding:

$$OR_2 = OR_1 \left(\frac{P_1(S = 1|X, Y)P_1(S = 1 \mid \overline{X}, \overline{Y})}{P_1(S = 1 \mid \overline{X}, Y)P_1(S = 1 \mid X, \overline{Y})} \right) \tag{3}$$

Note that making the multiplier of OR_1 in eq. (3) different than 1 entails $OR_2 \neq OR_1$, which will happen for *almost all* parametrizations of $P_1(S = 1 \mid .)$ independently of the one chosen for $P_1(X, Y)$. In case there are additional nodes pointing to S, we can just make them independent of S in this new parametrization given that compatibility with the subgraph is enough to ensure compatibility with G.

Now, let us consider the case in which condition 2 of the theorem fails, i.e., there is no OR-admissible sequence in relation to $(X, Y, \{\})$. Let $\mathbf{Z} = \mathbf{V} \setminus \{X, Y, S\}$, and expand $P_1(X, Y, S = 1)$ in the following way[1]:

$$\begin{aligned} P_1(X, Y, S = 1) &= \sum_Z P_1(X, Y, S = 1, \mathbf{Z}) \\ &= \sum_Z P_1(X \mid \mathbf{PA_x}) \ldots P_1(S = 1 \mid \mathbf{PA_s}) \end{aligned}$$

[1] It clear that we should consider in the expression above (in respect to $\mathbf{Z}$) just the nodes that are somehow related to S, i.e., its ancestors, otherwise we could just sum these vertices out because they do not offer any additional constraint over the distribution of interest related to OR, and then in its respective parameterization.

$$= \sum_{Z} \prod_{\mathbf{V} \cap S = 1} P_1(V_i \mid \mathbf{PA_i}) \tag{4}$$

Notice that each term in eq. (1) can be rearranged for each assignment of S' parents (i.e., $\mathbf{PA_s} = \mathbf{pa_s^{(j)}}$), for instance, we can write based on eq. (4):

$$P_1(X, Y, S = 1) =$$
$$P_1(S = 1 \mid \mathbf{PA_s} = \mathbf{pa_s^{(1)}}, \lambda)\Big(\sum_{\mathbf{Z}, \mathbf{PA_s} = \mathbf{pa_s^{(1)}}} \prod_{\mathbf{V} \setminus S} P_1(V_i \mid \mathbf{PA_i})\Big) +$$
$$P_1(S = 1 \mid \mathbf{PA_s} = \mathbf{pa_s^{(2)}}, \lambda)\Big(\sum_{\mathbf{Z}, \mathbf{PA_s} = \mathbf{pa_s^{(2)}}} \prod_{\mathbf{V} \setminus S} P_1(V_i \mid \mathbf{PA_i})\Big) +$$
$$\ldots$$
$$P_1(S = 1 \mid \mathbf{PA_s} = \mathbf{pa_s^{(k)}}, \lambda)\Big(\sum_{\mathbf{Z}, \mathbf{PA_s} = \mathbf{pa_s^{(k)}}} \prod_{\mathbf{V} \setminus S} P_1(V_i \mid \mathbf{PA_i})\Big) \tag{5}$$

where k is the number of configurations of S' parents, and λ indexes configurations of X or Y whenever one of them is a parent of S. Given eq. (5), let us call $P_1(S = 1 \mid \mathbf{PA_s} = \mathbf{pa_s^{(1)}}, \lambda) = \alpha_1^{\lambda}$, $P_1(S = 1 \mid \mathbf{PA_s} = \mathbf{pa_s^{(2)}}, \lambda) = \alpha_2^{\lambda}, \ldots$, and also call $\sum_{\mathbf{Z}, \mathbf{PA_s} = \mathbf{pa_s^{(j)}}} \prod_{\mathbf{V} \setminus S} P_1(V_i \mid \mathbf{PA_i}) = f_j(x, y)$ for each configuration $X = x, Y = y, \mathbf{PA_s} = \mathbf{pa_s^{(j)}}$. Then, we can write eq. (5) in the following simplified manner:

$$P_1(X, Y, S = 1) = \alpha_1^{\lambda} f_1(x, y) + \alpha_2^{\lambda} f_2(x, y) + \ldots \tag{6}$$

for all values of X and Y. We can then rewrite OR_2 based on eq. (6) as

$$OR_2 = \frac{(\alpha_1^{\lambda} f_1(x, y) + \alpha_2^{\lambda} f_2(x, y) + \ldots)}{(\alpha_1^{\overline{\lambda}} f_1(\overline{x}, y) + \alpha_2^{\overline{\lambda}} f_2(\overline{x}, y) + \ldots)} \times \frac{(\alpha_1^{\overline{\lambda}} f_1(\overline{x}, \overline{y}) + \alpha_2^{\overline{\lambda}} f_2(\overline{x}, \overline{y}) +)}{(\alpha_1^{\lambda} f_1(x, \overline{y}) + \alpha_2^{\lambda} f_2(x, \overline{y}) + \ldots)} \tag{7}$$

and similarly for OR_1:

$$OR_1 = \frac{(f_1(x, y)) + f_2(x, y) + \ldots)(f_1(\overline{x}, \overline{y}) + f_2(\overline{x}, \overline{y}) + \ldots)}{(f_1(\overline{x}, y)) + f_2(\overline{x}, y) + \ldots)(f_1(x, \overline{y}) + f_2(x, \overline{y}) + \ldots)} \tag{8}$$

There is an important observation here. Given that there is no admissible sequence relative to $(X, Y, \{\})$, there exists a set $\mathbf{W}$ such that $\mathbf{W}$ is needed to separate S from X or Y, but also $(\mathbf{W} \not\perp\!\!\!\perp \{X, Y\} \mid \mathbf{Z'})$, for $\mathbf{Z'}$ non-descendents of $\mathbf{W}$ and in $Anc(S)$, otherwise there will exist an admissible sequence. If $\mathbf{W}$ is different than $\{S\}$, it is the case that, by construction, $\mathbf{W}$ is contained in the factor $f_i(x, y)$. Thus, we have an asymmetry given that $\mathbf{W}$, and so $f_i()$, change depending *simultaneously* on the specific instantiation of X and Y, and consequently eq. (7) cannot be simplified in the general case. I.e., the linear combinations encoded in $f_i()$'s at eq. (7) do not deteriorate, factoring out independently of the given parametrization given that there is a different element in each one of them.

Now let us consider the following parametrization for P_1: set $P_1(V_i \mid \mathbf{PA_i}) = 1/2$ for all families except for the family of the S node (i.e., $P(S = 1 \mid \mathbf{PA_s})$) and the exclusive families included in the factor $f_i(x, y)$ (i.e., for when $X = x, Y = y$). Thus, rewrite OR_2 based on eq. (7):

$$OR_2 \quad = \quad \frac{(\alpha_1^{\lambda} f_1(x, y) + \alpha_2^{\lambda} f_2(x, y) + \ldots)}{(1/2)^l (\alpha_1^{\lambda} + \alpha_2^{\lambda} + \ldots)} \tag{9}$$

where l is equal to k minus the number of summands in the respective expression (eq. (5)). Let us also rewrite eq. (8) accordingly with this given parametrization, which yields:

$$OR_1 = \frac{(f_1(x, y) + f_2(x, y) + \ldots)}{k(1/2)^l} \tag{10}$$

After applying some simplifications on eqs. (9) and (10), we obtain, respectively,

$$OR_2 = \frac{(\alpha_1^{\lambda} f_1(x, y) + \alpha_2^{\lambda} f_2(x, y) + \ldots)}{(\alpha_1^{\lambda} + \alpha_2^{\lambda} + \ldots)} \tag{11}$$

and

$$OR_1 = \frac{(f_1(x, y) + f_2(x, y) + \ldots)}{k} \tag{12}$$

Notice that OR_2 in eq. (11) is the weighted arithmetic mean of $f_i(.)$'s averaged by α_i^{λ}'s, and OR_1 in eq. (12) is the arithmetic mean of $f_i(.)$'s. After simplifications, the remaining parameters lie in the space $[0, 1]^{m+k}$, where m is the number of free parameters in $f_i(.)$'s. Note that $OR_1 - OR_2 = 0$ adds a constraint in this space, and in order to satisfy it we should choose any point in a surface in $[0, 1]^{m+k-1}$ inside $[0, 1]^{m+k}$, i.e., which has Lebesgue measure zero. Consequently, if we randomly choose parameters the equality will *almost never* hold (and the inequality $OR_1 \neq OR_2$ *almost always*), and then just randomly draw the parameters from $[0, 1]^{m+k}$ until this is the case, which finishes this part of the proof. The case of the conditional OR is similar, and we basically have to write appropriately eqs. (1) and (2) considering $\mathbf{C}$, and exactly the same reasoning applies.

For the case when the graph contains unobservable variables, the proof is essentially the same except that an appropriate parametrization of the underlying generating model should be used – for such, consider the factorization given in (Evans and Richardson, 2011).

Theorem 2

For the necessity of the condition, we need to show that the failure of any ancestor A_i of S that is also a descendant of X (including S itself) to be separated (from either X or Y) prevents recoverability of $OR(Y, X \mid \mathbf{C})$. Indeed, A_i cannot be part of admissible sequence nor can any of its children be part of an admissible sequence, because in order to separate any such child from either X or Y we would need to condition on the father A_i, and then, the sequence will become non-admissible. Proceeding by induction, we eventually reach S itself, whose failure to enter an admissible sequence renders the existence of such sequence impossible. By Theorem 1, the inexistence of admissible sequence implies the not G-recoverability of $OR(X, Y, \mathbf{C})$. □

Theorem 3

We use along the proof some graphoid axioms and other DAG properties as shown in (Pearl, 1988). Let us first consider the correctness of the algorithm. The main idea of the reduction sequence is to use each conditional independence (CI) in step 2 of the sink-procedure to substantiate an OR reduction, creating a mapping starting from the s-biased data $OR(X, Y \mid \mathbf{C}, Z_1, ..., Z_k, S = 1)$ and reaching the target (unbiased) expression $OR(X, Y \mid \mathbf{C})$. If nodes are not added in step 3 of the algorithm, it is obvious that the sequence induces a valid step-OR reduction, which witnesses the OR G-recoverability. So, let us consider the case when nodes have to be added to $\mathbf{T}$ along the execution of the algorithm. At each step i, we reduce $OR(X, Y \mid C, \mathbf{T}, Z_1, ..., Z_i)$ to $OR(X, Y \mid \mathbf{C}, \mathbf{T}, Z_1, ..., Z_{i-1})$ allowed by the CI in step 2. But given that $\mathbf{T_i}$ can be added to $\mathbf{T}$ along the execution of the algorithm, we need to show that this operation is allowed, i.e., it does not invalidate the construction of the desired mapping between the unbiased OR and the s-biased one. Towards contradiction, consider an arbitrary node Z_j such that

$$(Z_j \perp\!\!\!\perp X \mid \mathbf{C}, \mathbf{T}, Y, Z_1, ..., Z_{j-1}) \text{ or } (Z_j \perp\!\!\!\perp Y \mid \mathbf{C}, \mathbf{T}, X, Z_1, ..., Z_{j-1}) \quad (13)$$

Now, consider the first Z_k such that $k < j$ and, in order to satisfy step 2 in the sink-procedure, $\mathbf{W}$ has to be added to the conditioning set, then

$$(Z_k \perp\!\!\!\perp X \mid \mathbf{C}, \mathbf{T}, Y, Z_1, ..., Z_{k-1}, W) \text{ or } (Z_k \perp\!\!\!\perp Y \mid \mathbf{C}, \mathbf{T}, X, Z_1, ..., Z_{k-1}, W) \quad (14)$$

but also

$$(Z_j \perp\!\!\!\perp X \mid \mathbf{C}, \mathbf{T}, Y, Z_1, ..., Z_{j-1}, W) \text{ or } (Z_j \perp\!\!\!\perp Y \mid \mathbf{C}, \mathbf{T}, X, Z_1, ..., Z_{j-1}, W) \quad (15)$$

is false. If the sink-procedure ends, it is also true that

$$(\mathbf{T} \perp\!\!\!\perp Y \mid \mathbf{C}, X) \quad (16)$$

From eq. (13), all paths from Z_j to X or Y (including the ones passing through $\mathbf{W}$) are closed after conditioning on $\{\mathbf{C}, \mathbf{T}, Y, Z_1, ..., Z_{j-1}\}$. From eq. (14) and the minimal choice of T_i in step 3, it must be the case that there is a path p from Z_k to X or Y such that p is blocked by some $W \in \mathbf{W}$. From eq. (15), there exists a path p' that has to be open after condition on $\mathbf{W}$, and therefore there exists a collider U such that $U = W$ or $W \in Desc(U)$. Let us consider two possible scenarios for p', the first when it goes from Z_j to Y, and the second when it goes from Z_j to X. In the former case, there is an open path from W to Y, which is a contradiction with eq. (16) given that $\mathbf{W} \subseteq \mathbf{T}$. Then it must be the case that $\mathbf{W}$ only blocks paths ending in X, so let us assume the case in which the end node in p' is X. From (14), p is such that $Z_k \longleftrightarrow\!\!\leftarrow ... - W - ... \rightarrow\!\!\longleftrightarrow X$, where we are condition on all intermediate converging arrows and W must be a chain or a common cause (i.e., $\rightarrow W \rightarrow$ or $\leftarrow W \rightarrow$). Split p into $p_1 : Z_k \ldots W$, and $p_2 : W \ldots X$. From eq. (15), p' is such that $\mathbf{W}$ opens a collider U, then the path from Z_j to X. Split p' into $p'_1 : Z_j ... \rightarrow U$ and $p'_2 : U \leftarrow ...X$. Now we have two possibilities. If p_2 is such that $W \rightarrow \ldots X$, we can concatenate $Z_k \overset{p'_1}{\rightarrow} U \rightarrow W \overset{p_2}{\rightarrow} X$, which shows an open path from Z_k to X even before conditioning on W, contradiction. If p_2 is such that $W \leftarrow \ldots X$, p_1 must be $W \rightarrow \ldots Z_k$, and we have two possibilities: (a) Z_k can be a descendent of W, and in this case the collider in U is already open even without conditioning on W, contradiction; (b) W is connected to Z_k through some collider, for instance, p_1 could be $W \rightarrow \ldots \rightarrow C \leftarrow \ldots Z_k$, but similarly as before, given that we condition on C, which is a descendent of W, and so of U, the collider was already conditioned as well as the path from Z_k to X open, contradiction. Therefore, it cannot be the case that after adding $\mathbf{T_k} \subseteq NonDesc(X)$ to block paths from Z_k to X or Y, there is a node Z_j such that $k < j$, and which previously had its paths to X or Y blocked, turned to have them open after conditioning on T_k. Thus, we are allowed to modify each CI obtained in step 2 before Z_k in the sequence adding $\mathbf{T_k}$, and then based on the admissible sequence starting from $OR(X, Y \mid \mathbf{C}, \mathbf{T}, Z_1, ..., Z_n)$, we can reduce it through this new augmented CIs of step 2 until reaching the desired expression $OR(X, Y \mid \mathbf{C})$.

Now we consider the complexity of the algorithm, and we show that it runs in polynomial time. Notice that only the step 3 of the algorithm could imply some backtracking – i.e., when it chooses a (minimal) set $\mathbf{T_i}$ of non-descendants of X that renders the equality in step 2 to be true. The choice of separating set *per se* is polynomial, see footnote 5.

Consider that the choice of $\mathbf{T_i}$ implies failure in step 5 when it tests the validity of $(\mathbf{T} \perp\!\!\!\perp Y \mid X, \mathbf{C})$. Assume that it exists a sequence $\mathbf{Q}$ of ancestors of S and not ancestors of X, $(Z_1, ..., Z_k, ..., Z_n)$ such that for each Z_i there is a separating set $\mathbf{T_i}$ which makes the independence test valid. Let $\mathbf{T} = \bigcup \mathbf{T_i}$, and assume that $(\mathbf{T} \perp\!\!\!\perp Y \mid X, \mathbf{C})$ holds. Assume now that in round k, the sink procedure chooses a different (minimal) separating set than $\mathbf{T_k}$, and call this new set $\mathbf{T'_k}$, and subsequently $(\mathbf{T'_{k+1}}, ..., \mathbf{T'_n})$. We have the new sequence $\mathbf{Q'}$ with additional separators $(\mathbf{T_1}, ..., \mathbf{T_{k-1}}, \mathbf{T'_k}, ..., \mathbf{T'_n})$. Call $\mathbf{T'} = \bigcup \mathbf{T'_i}$, and $\mathbf{\Delta} = \mathbf{T'} \setminus (\mathbf{T} \cap \mathbf{T'})$.

We have that $(\mathbf{T'} \not\perp\!\!\!\perp Y \mid X, \mathbf{C})$ holds, or just $(\mathbf{\Delta} \not\perp\!\!\!\perp Y \mid X, \mathbf{C})$. (This follows from $(\mathbf{\Delta} \perp\!\!\!\perp Y \mid X, \mathbf{C})$, which by composition yields $(\mathbf{T'} \perp\!\!\!\perp Y \mid X, \mathbf{C})$, contradiction. See also (Pearl and Paz, 2010).) Let $\delta \in \mathbf{\Delta}$ be the first node such that that $\mathbf{Q}$ and $\mathbf{Q'}$ disagree and which make step 5 to fail. δ blocks at least one path from Z_k to X (after condition on $\{\mathbf{C}, Y, \mathbf{T}, Z_1, ..., Z_{k-1}, \mathbf{T_i} \setminus \delta\}$) or from Z_k to Y (after condition on $\{\mathbf{C}, X, \mathbf{T}, Z_1, ..., Z_{k-1}, \mathbf{T_i} \setminus \delta\}$), otherwise the sequence will not be admissible (pass in the test of step 2). By construction, it must be the case that there is an open path from Z_k to Y passing through δ (after cond. on $\{C, X, Q, Z_1, ..., Z_{k-1}, \mathbf{T_i} \setminus \delta\}$).

Let p be part of this path from δ to Y (or, $\delta - ... - Y$). There must exist in $\mathbf{Q}$ a vertex v which blocks this same path from Z_k to $\{X, Y\}$ or $\{Y\}$ in the test of step 2. But v is in p or connected through an open path p' to δ (i.e., $p : \delta - ... - v - ... - Y$ or $v - ... - p' - ... - \delta - ... - p - ... - Y$), otherwise we would not need δ in the first place, contradicting minimality. In both cases, there is an open path from v to Y, which contradicts the assumption about $\mathbf{Q}$ validating $(T \perp\!\!\!\perp Y \mid X, C)$ as true, and therefore it cannot exist such δ. Applying the same reasoning for the whole sequence $\mathbf{Q'}$ inductively, we conclude that it cannot exist such sequence. Therefore, step 5 does not imply any backtracking.

Similarly, let us consider the case when the choice of $\mathbf{T_j}$ implies failure in a subsequent step 2. In the sequence $\mathbf{Q'}$, it is true that when the algorithm chooses $\mathbf{T_j}$ to satisfy the admissibility of Z_j, it blocks some paths from Z_j to X. Now, assume that for Z_k, $k < j$, there is an open path through $\mathbf{T_j}$, i.e., $Z_k \longleftrightarrow U \longleftrightarrow X$, where $U = T_j$ or $T_j \in \mathit{Desc}(U)$. But if you do not choose $\mathbf{T_j}$ (or any other node that blocks this path), we would have an open path from Z_k to X through $\mathbf{T_j}$, contradiction.

We now argue about the completeness of the procedure. Let us first consider the case in which there is not X-independent variable in the admissible sequence, the sink-procedure will return an admissible sequence whenever one exists. Notice that the sink-procedure performs a search for an admissible sequence in reverse topological order, and this only makes the conditional independence's tests easier than in any other order. This is so because in each step, we are adding all non-descendents of Z_k (are non-colliders for Z_k), which completely disconnects Z_k from X or Y except for paths passing through non-descendents of X. (Also, non step-wise reductions can be converted to step-wise one through the graphoids decomposition and weak union.)

Assume that there is a sequence $(A_1, ..., A_m)$ called $\mathbf{A}$ that does not follow the order given by the sink-procedure and it is admissible. Now, let us call $\mathbf{Q}$ the sequence $(Z_1, ..., Z_n)$ given by the sink-procedure, and further assume that $\mathbf{Q}$ is not admissible. It is true that the last element of both sequences is S, and in $\mathbf{Q}$ we would have the blocking set $\{Z_1, ..., Z_{n-1}\}$ while in $\mathbf{A}$ we would have $\{A_1, ..., A_{m-1}\}$. It is true that $\{A_1, ..., A_{m-1}\} \subseteq \{Z_1, ..., Z_{n-1}\}$, and this is an invariant along the algorithm for all nodes in $\mathbf{A}$. Recall two facts: (a) for now, we are assuming that there are not disagreements between $\mathbf{T_Q}$ and $\mathbf{T_A}$; (b) adding descendents of Z_k in each step can only open some paths and spoil separation. It must be the case for the sink-procedure to fail, there exists $Z_k \in \mathbf{Q}$ such that $(Z_k \perp\!\!\!\perp X \mid Y, \mathbf{C}, Z_1, ..., Z_{k-1})$ and $(Z_k \perp\!\!\!\perp Y \mid X, \mathbf{C}, Z_1, ..., Z_{k-1})$ are both false. Thus, there is at least one path from Z_k to X and from Z_k to Y that are not blocked by $\{Z_1, ..., Z_{k-1}\} \cup \{C\}$ (and respectively, $\{Y\}$ and $\{X\}$); call the set of these paths P_1 and P_2, respectively.

Assume that $\mathbf{A}$ also chooses Z_k at some point along its execution, and Z_k is labeled there A_m. It must be the case that all paths from A_m to X or all paths from A_m to Y are blocked by $\{A_1, ..., A_{m-1}\} \cup \{C\}$ (and respectively, $\{Y\}$ and $\{X\}$). But if $\{A_1, ..., A_{m-1}\} \subseteq \{Z_1, ..., Z_{k-1}\}$, this is a contradiction. Now assume that $\mathbf{A}$ does not choose Z_k along its execution. There are ancestors of S which have to block P_1 from S to X or P_2 from S to Y, and we consider without loss of generality the subset $\{A_1, ..., A_l\}$ that renders this separation to hold. Consider A_j the first descendant of Z_k in G^* that is in $\{A_1, ..., A_l\}$. If such node is S, we reach a contradiction. Assume that A_j is not S but some of its ancestors. To separate A_j from X or Y, we need to block the paths from it to X or Y, but there are unblockable paths P_1 and P_2 passing through Z_k ($A_j \leftarrow ... - Z_k - P_1 - X$ or $A_j \leftarrow ... - Z_k - P_2 - Y$), and therefore A_j cannot be part of an admissible sequence, contradiction. Then, it is the case that if both algorithms do not disagree in the choice of the non-

M	1	2	3	4	5	6	7	8	9	10	11	12
1	$(c_1-1)b_1$	c_1b_2	c_1b_3	c_1b_4								
2	c_2b_1	$(c_2-1)b_2$	c_2b_3	c_2b_4								
3	c_3b_1	c_3b_2	$(c_3-1)b_3$	c_3b_4								
4					$(c_4-1)b_1$	c_4b_2	c_4b_3	c_4b_4				
5					c_5b_1	$(c_5-1)b_2$	c_5b_3	c_5b_4				
6					c_6b_1	c_6b_2	$(c_6-1)b_3$	c_6b_4				
7									$(c_7-1)b_1$	c_7b_2	c_7b_3	c_7b_4
8									c_8b_1	$(c_8-1)b_2$	c_8b_3	c_8b_4
9									c_9b_1	c_9b_2	$(c_9-1)b_3$	c_9b_4
10	$(1-c_{10})b_1$	$-c_{10}b_2$	$-c_{10}b_3$	$-c_{10}b_4$	$(1-c_{10})b_1$	$-c_{10}b_2$	$-c_{10}b_3$	$-c_{10}b_4$	$(1-c_{10})b_1$	$-c_{10}b_2$	$-c_{10}b_3$	$-c_{10}b_4$
11	$-c_{11}b_1$	$(1-c_{11})b_2$	$-c_{11}b_3$	$-c_{11}b_4$	$-c_{11}b_1$	$(1-c_{11})b_2$	$-c_{11}b_3$	$-c_{11}b_4$	$-c_{11}b_1$	$(1-c_{11})b_2$	$-c_{11}b_3$	$-c_{11}b_4$
12	$-c_{12}b_1$	$-c_{12}b_2$	$(1-c_{12})b_3$	$-c_{12}b_4$	$-c_{12}b_1$	$-c_{12}b_2$	$(1-c_{12})b_3$	$-c_{12}b_4$	$-c_{12}b_1$	$-c_{12}b_2$	$(1-c_{12})b_3$	$-c_{12}b_4$

descendents of X, there is indeed not admissible sequence. For the case when we add X-independent variables along the sequence, the result also follows, and this is so based on the fact shown previously that there is no backtracking in the choice of $\mathbf{T_i}$, and any algorithm that chooses T_i consistently obtains the same outcome in terms of separation. Each time that the sink-procedure does not return any sequence, we can produce a counter-example for the G-recoverability of the triplet $(X, Y, \mathbf{C})$ based on the construction of Theorem 1. $\square$

Theorem 4

Let us first show the result for the binary case. To match the dimensionality requirement, we assume that $\mathbf{Z} = Z_1 \cup Z_2$ and both Z_1 and Z_2 are binary satisfying:

$$P(Z_1, Z_2 \mid X, Y, S) \quad = \quad P(Z_1, Z_2 \mid X, Y) \quad (17)$$

To simplify the notation, let us write:

- $P(X = x, Y = y \mid Z_1 = z_1, Z_2 = z_2) = \alpha_{xy,z_1z_2}$
- $P(Z_1 = z_1, Z_2 = z_2) = \beta_{z_1z_2}$
- $P(Z_1 = z_1, Z_2 = z_2 \mid X = x, Y = y) = \gamma_{z_1z_2,xy}$

Note that the parameters $\gamma_{z_1z_2,xy}$ and $\beta_{z_1z_2}$ impose constraints on the distribution α_{xy,z_1z_2}, which can be made explicit by the following equation,

$$\gamma_{z_1z_2,xy} \quad = \quad \frac{\alpha_{xy,z_1z_2}\beta_{z_1z_2}}{\sum_{z_1',z_2'} \alpha_{xy,z_1'z_2'}\beta_{z_1'z_2'}} \quad (18)$$

Now, for a given assignment $< X = 0, Y = 0 >$, let us list all independent parameters $\gamma_{z_1z_2,00}$,

$$\begin{aligned}
\gamma_{00,00} &= \frac{\alpha_{00,00}\beta_{00}}{\sum_{z_1',z_2'} \alpha_{00,z_1'z_2'}\beta_{z_1'z_2'}} \\
\gamma_{01,00} &= \frac{\alpha_{00,01}\beta_{01}}{\sum_{z_1',z_2'} \alpha_{00,z_1'z_2'}\beta_{z_1'z_2'}} \\
\gamma_{10,00} &= \frac{\alpha_{00,10}\beta_{10}}{\sum_{z_1',z_2'} \alpha_{00,z_1'z_2'}\beta_{z_1'z_2'}} \quad (19)
\end{aligned}$$

Note that $\gamma_{11,00}$ is not an independent parameter because it is completely determined by the other three equations in (19) given the integrality constraint. For now, we have 3 equations and 4 unknown variables ($\{\alpha_{00,00}, \alpha_{00,01}, \alpha_{00,10}, \alpha_{00,11}\}$.)

Similarly, we write the constraints for the assignments $< X = 1, Y = 0 >$ and $< X = 0, Y = 1 >$, respectively,

$$\begin{aligned}
\gamma_{00,10} &= \frac{\alpha_{10,00}\beta_{00}}{\sum_{z_1',z_2'} \alpha_{10,z_1'z_2'}\beta_{z_1'z_2'}}, \ldots \quad (20) \\
\gamma_{00,01} &= \frac{\alpha_{01,00}\beta_{00}}{\sum_{z_1',z_2'} \alpha_{01,z_1'z_2'}\beta_{z_1'z_2'}}, \ldots \quad (21)
\end{aligned}$$

Now, we can write the equations for the constraints relative to the variables α_{11,z_1z_2} as a function of the previous variables $\{\alpha_{00,z_1z_2}, \alpha_{01,z_1z_2}, \alpha_{10,z_1z_2}\}$,

$$\gamma_{00,11} = \Big((1-(\alpha_{00,00}+\alpha_{01,00}+\alpha_{10,00}))\beta_{00}\Big) / \left(\sum_{Z_1',Z_2'} \Big(1-\left(\alpha_{00,z_1'z_2'}+\alpha_{01,z_1'z_2'}+\alpha_{10,z_1'z_2'}\right)\Big)\beta_{z_1'z_2'}\right), \ldots \quad (22)$$

Notice that the parameters $\gamma_{z_1z_2,11}$ are independent, and we have 12 equations and 12 unknowns, but it remains to show that the equations are all independent (notice that the last three constraints in eq. (22) involve variables of the other constraints). Another fact to observe is that the system is indeed linear. We show that the matrix M, induced by the eqs. (19, 20, 21, 22), is linear and (almost surely) invertible, and generates an unique solution. M is invertible if and only if its determinant is non-zero. For convenience, let us display the variables α_{xy,z_1z_2} column-wise, renaming $\beta_{z_1z_2}$ as constants $b_1 - b_4$, and $\gamma_{z_1z_2,xy}$ as constants $c_1 - c_{12}$. The matrix is shown on the top of the previous page.

In what follows, we exploit the block structure of M and apply the following transformations to better visualize its determinant.

1. First note that all columns $\{1, 5, 9\}$ are multiplied by b_1, which can be factored out by the determinant property. Similarly for the other columns in respect to $\{b_2, b_3, b_4\}$, which can be expressed as $det(M) = (b_1b_2b_3b_4)^3 det(M^{(1)})$, where $M^{(1)}$ is the resultant matrix.

2. Let us sum lines $\{1, 4, 7\}$ to line 10, lines $\{2, 5, 8\}$ to line 11, and $\{3, 6, 9\}$ to line 12, which generate matrix $M^{(2)}$.
3. We now sum the columns of $M^{(2)}$, -1 times column 4 to column 1, -1 times column 4 to column 2, and -1 times column 4 to column 3 (similarly for the other blocks), which yields $M^{(3)}$.
4. Sum the columns of $M^{(3)}$, c_1 times column 1, c_2 times column 2 and c_3 times column 3 to column 4 (similarly for the other blocks), yielding $M^{(4)}$.
5. Now, reorder the columns, "pushing" column 4 and 8 towards the end, call the resultant matrix $M^{(5)}$.

Now we are done, notice that the $det(M) = (b_1 b_2 b_3 b_4)^3 det(M^{(5)})$, and the determinant of $M^{(5)}$ is the determinant of two block matrices, the square matrix $M_1^{(5)}$ from lines 1-9 multiplied by another square matrix $M_2^{(5)}$ from lines 10-12. Note that $det(M_1^{(5)}) = -1$, and remains to show that $det(M_2^{(5)})$ is almost always different than zero. The parameters c_1 to c_{12} are independent, and given the form obtained to $M_2^{(5)}$ where all entries are independent, this implies that $M_2^{(5)}$ is non-singular almost surely, and so it is $M^{(5)}$ – coincidental cancellations will occur with Lebesgue measure zero.

Therefore, we consider M as full rank, which can be solved algebraically with standard techniques yielding the solution $\alpha = M^{-1}\gamma$. This result, together with $P(\mathbf{Z})$ yields the joint distribution $P(Y, X, \mathbf{Z})$. The case for non-binary variables follows in a straightforward way, just noticing the requirement for agreement between the dimensions of the IV set $\mathbf{Z}$ and $\{X, Y\}$. □

Corollary 5

First, apply Theorem 4 to the variables $\{W, Y\}$ replacing X with W, and obtain $P(W, Y)$. Further note that $P(X \mid Y, W, S = 1) = P(X \mid Y, W)$, which together with the first observation finishes this part of proof. The proof for when we do not rely on $\mathbf{Z}$ is essentially the same. □

1 Acknowledgement

The authors would like to thank the reviewers for their comments that help improve the manuscript. This paper also benefited from discussions with Humberto Silva Naves, Onyebuchi Arah, and Sander Greenland. This research was supported in parts by NIH #1R01 LM009961-01, NSF #IIS-0914211, #IIS-1018922, ONR #N000-14-09-1-0665, and #N00014-10-1-0933.

References

Evans, R. J. and Richardson, T. S. (2011). Marginal log-linear parameters for graphical markov models. arXiv:1105.6075 [stat.ME].

Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann, San Mateo, CA.

Pearl, J. and Paz, A. (2010). Confounding equivalence in causal equivalence. In *Proceedings of the Twenty-Sixth Conference on Uncertainty in Artificial Intelligence*. AUAI, Corvallis, OR, 433–441.