Appendix to paper forthcoming in *Neural Information Processing Systems (NIPS) Conference, Advances in Neural Information Processing Systems*, Curran Associates, Inc., 2016.

TECHNICAL REPORT
R-444-Appendix
September 2016

Identification and Overidentification of Linear Structural Equation Models

Bryant Chen
University of California, Los Angeles
Computer Science Department
Los Angeles, CA, 90095-1596, USA

Algorithm 1 HT-ID(G, Σ, IDEdges)

```
Initialize: EdgeSets ← all connected edge sets in G
repeat
    for each ES in EdgeSets do
        v ← He(ES)
        for each E ⊂ ES such that E ⊄ IDEdges do
            A_E ← Allowed(E, IDEdges, G)
            Y_E ← MaxFlow(G, E, A_E))
            if |Y_E| = |Ta(E)| then
                Identify E using Theorem 1
                IDEdges ← IDEdges ∪ E
            end if
        end for
    end for
until All coefficients have been identified or no coefficients have been identified in the last iteration
return IDEdges
```

Theorem 1. *If a g-HT-admissible set for directed edges E_v with head v exists then E_v is identifiable. Further, let $Y_{E_v} = \{y_1, ..., y_k\}$ be a g-HT-admissible set for E_v, $Ta(E_v) = \{p_1, ..., p_k\}$, and Σ be the covariance matrix of the model variables. Define $\mathbf{A}$ as*

$$\mathbf{A_{ij}} = \begin{cases} [(I-\Lambda)^T \Sigma]_{y_i p_j}, & y_i \in htr(v) \text{ or } y_i \text{ connected} \\ & \text{to } Pa(v) \setminus Ta(E_v), \\ \Sigma_{y_i p_j}, & y_i \notin htr(v) \end{cases} \tag{1}$$

and $\mathbf{b}$ *as*

$$\mathbf{b_i} = \begin{cases} [(I-\Lambda)^T \Sigma]_{y_i v}, & y_i \in htr(v) \text{ or } y_i \text{ connected} \\ & \text{to } Pa(v) \setminus Ta(E_v), \\ \Sigma_{y_i v}, & y_i \notin htr(v) \end{cases} \tag{2}$$

Then $\mathbf{A}$ is an invertible matrix and $\mathbf{A} \cdot \Lambda_{Ta(E_v),V} = \mathbf{b}$.

Proof. The proof for this theorem is similar to the proof of Theorem 1 in Foygel et al. (2012). Rather than giving a complete proof, we simply explain why our changes are valid. The g-HTC identifies arbitrary sets of directed edges belonging to a node rather than all of the directed edges belonging to a node. It is able to do this because of two changes. First, sets that contain nodes that are connected to $Pa(v) \setminus Ta(E)$ via half-treks cannot be half-trek admissible for E (see Definition 6). As a result, the paths from half-trek admissible set, Y_E, to v travel only through coefficients of E and no other coefficients of E. This ensures that $\mathbf{A} \cdot \Lambda_{Ta(E),v} = \mathbf{b}$. Second, nodes that are connected to $Pa(v) \setminus Ta(E)$ are not allowed unless their coefficients that lie on paths to $Pa(v) \setminus Ta(E)$ are identified. Likewise, nodes that are half-trek reachable from v are not allowed unless their coefficients that lie on the half-treks from v are identified. This ensures that $\mathbf{A}$ and $\mathbf{b}$ are computable. Other coefficients need not be identified because they will vanish from $\mathbf{A}$ and $\mathbf{b}$ during the computations, $((I-\Lambda)^T \cdot \Sigma)_{y_i p_j}$ and $((I-\Lambda)^T \cdot \Sigma)_{y_i v}$, due to zeroes in the matrix Σ. □

Theorem 2. *Let Y_E be a set of maximal size that satisfies conditions (ii)-(iv) of the g-HTC for a set of edges, E, with head v. If there exists a node w such that*

(i) there exists a half-trek from w to Ta(E),

(ii) $w \notin (v \cup Sib(v))$, and

(iii) w is g-HT-allowed for E,

then we obtain the equality constraint, $\mathbf{a_w}\mathbf{A}^{-1}_{\text{right}}\mathbf{b} = b_w$, where $\mathbf{A}^{-1}_{\text{right}}$ is the right inverse of $\mathbf{A}$.

Proof. As long as conditions (ii) and (iii) of the g-HTC are satisfied by Y_e, the rows of $\mathbf{A}$ are linearly independent and $\mathbf{A} \cdot \Lambda_{Ta(e),v} = \mathbf{b}$. (See the proof of Theorem 1 in (Foygel et al., 2012).) Similarly, if w satisfies the above conditions then $\mathbf{a_w} \cdot \Lambda_{Ta(e),v} = b_w$. Additionally, since Y_E is a maximal set for which there exists a system of half-treks from Y_E to $Ta(E)$ with no sided intersection, there does not exist a system of half-treks from $Y_E \cup \{w\}$ to $Ta(E)$ with no sided intersection. According to Foygel et al. (2012), this implies that $\mathbf{a_w} = \mathbf{d}^T \cdot \mathbf{A}$ and $\mathbf{b_w} = \mathbf{d}^T \cdot \mathbf{b}$ for some vector $\mathbf{d}$. In other words, the equation $\mathbf{a_w} \cdot \Lambda_{Ta(e),v} = b_w$ is a linear combination of the equations represented by $\mathbf{A} \cdot \Lambda_{Ta(e),v} = \mathbf{b}$. As a result, we obtain the constraint $\mathbf{a_w}\mathbf{A}^{-1}_{\text{right}}\mathbf{b} = b_w$. □

Algorithm 2 can be used to identify coefficients and find HT-constraints given a graph of the model, G. Like Algorithm 1, it iterates through each connected edge set, attempting to identify it. However, after finding a maximal set that satisfies (ii)-(iv) of the g-HTC using MaxFlow, it looks for a node w that satisfies the conditions of Theorem 2 in order to obtain a HT-constraint. Prior to recursive decomposition, if a node z is d-separated from a node v, we trivially obtain the constraint that $\Sigma_{zv} = 0$. However, when we introduce recursive decomposition, we will see that the independence constraint on the sub-model corresponds to a non-conditional independence constraint in the joint distribution, $P(V)$. As a result, Algorithm 2 also outputs when variables are d-separated from one another given the empty set.

Algorithm 2 HT-Constraints(G, Σ, IDEdges)

```
Initialize: EdgeSets ← all connected edge sets in G
repeat
    for each ES in EdgeSets do
        for each E ⊂ ES such that E ⊄ IDEdges do
            A_E ← Allowed(E, IDEdges, G)
            Y_E ← MaxFlow(G, E, A_E))
            if |Y_E| = |Ta(E)| then
                Identify E using Theorem 1
                IDEdges ← IDEdges ∪ E
            end if
            for each w in A_E \ Y_E do
                if v ∈ htr(w) then
                    Output constraint: b_w = a_w · A^{-1}_Right · b
                else if w ∉ htr(v) then
                    Output constraint: Σ_wv = 0
                end if
            end for
        end for
    end for
until one iteration after all edges are identified or no new edges have been identified in the last
iteration return IDEdges
```

Algorithm 3 decomposes the graph according to its c-components and then applies Algorithm 2 to each sub-model. If there are still unidentified coefficients, then it removes descendant sets and decomposes again. The whole process is repeated until one iteration after every coefficient is identified or no new coefficients are identified in an iteration. $\Sigma_{P_{S_i}}$ is the covariance matrix of P_{S_i}, where S_i is a c-component. $\Sigma_{V \setminus D_i}$ is the covariance matrix after marginalizing D_i from Σ. Finally, $G_{V \setminus D_i}$ is the graph with the set D_i removed.

Algorithm 3 Decomp-HT(G, Σ)

```
Initialize: IDEdges ← ∅
repeat
    IDEdges ← IDEdges∪Rec-Decomp(G, Σ, IDEdges)
until One iteration after all coefficients have been identified or no coefficients have been identified
return IDEdges
```

Algorithm 4 Rec-Decomp(G, Σ, IDEdges)

```
V ← vertices in G
Edges ← all edges in G
for each c-component, S_i, in G do
    IDEdges = IDEdges∪ HT-Constraints(G_{S_i}, Σ_{S_i}, IDEdges)
end for
if IDEdges = Edges then
    Return IDEdges
else
    for each descendant set, D_i, in G do
        IDEdges ← IDEdges∪Rec-Decomp(G_{V\D_i}, Σ_{V\D_i}, IDEdges)
    end for
end if
return IDEdges
```

Theorem 3. *Let M be a linear SEM with variables V. Let $M^{'}$ be a non-parametric SEM with identical structure to M. If the direct effect of x on x for $x, y \in V$ is identified in $M^{'}$ then the coefficient Λ_{xy} in M is g-HTC identifiable and can be identified using Algorithm 3.*

Proof. Let G be the causal graph of M and $M^{'}$. Suppose the direct effect of x on y is identified in $M^{'}$. Then according to Theorem 3 of (Shpitser, 2008), there does not exist a subgraph of G that is a y-rooted c-tree (Shpitser, 2008). This implies that $MACS(y) = y$. By recursively decomposing the graph into c-components and marginalizing descendant sets, we can obtain a graph where only $MACS(y)$ and its parents remain in the graph. Since $MACS(Y) = y$, the parents of y in this graph represent a g-HT admissible set that allows the identification of all coefficients of y. □

Theorem 4. *Any Q-constraint, $Q_S \perp Z$, in a linear SEM, has an equivalent set of HT-constraints that can be discovered using Algorithm 3.*

Proof. Consider a Q-constraint, Q_S is not a function of Z. This constraint is obtained through some sequence of c-component decomposition and marginalization of descendant sets. In the last step, Q_S is identified from $Q_{S \cup W}$ for some W such that $Z \subset W \cup Pa(W)$. Let $G^{'} = G_{S \cup W}$. Now Q_S is not a function of Z implies that $Z \perp\!\!\!\perp_{G'} S | Pa(S)$ since Z must be ordered before S and, therefore, $Z \notin De(S)$. Similarly, $Z \perp\!\!\!\perp_{G'} S | Pa(S)$ implies that Q_S is not a function of Z. As a result, the Q-constraint is obtained if and only if $Z \perp\!\!\!\perp_{G'} S | Pa(S)$, where Z is ordered before S, and a Q-constraint is equivalent to a conditional independence constraint in the distribution, $P_{S \cup W}$.

Since pairwise independence implies independence in normal distributions, the constraint $Z \perp\!\!\!\perp S | Pa(S)$ is equivalent to the set of conditional independences, $\{z_i \perp\!\!\!\perp S | Pa(S)\}$, where $z_i \in Z$. We now show that there exists an equivalent HT-constraint for each conditional independence, $z_i \perp\!\!\!\perp S | Pa(S)$ in the distribution $P_{S \cup W}$. $G^{'}$ is obtainable from recursive c-component decomposition, and, in G', $Pa(S)$ satisfies conditions (i)-(iii) of the g-HTC for the edges from $Pa(S)$ to S. Additionally, z_i is not half-trek reachable from S and either has a half-trek to S or is separated entirely from S. In both cases, we obtain a HT-constraint that is equivalent to the conditional independence constraint $Z \perp S|Pa(S)$ in the distribution, $P_{S \cup W}$.

If z_i is separated entirely from S then the constraint is that z_i is independent of s. In Algorithm 2, this is exactly the constraint that is outputted. If z_i is separated from S by $Pa(S)$, then Algorithm 2 outputs a constraint that is equivalent to z_i is independent of S given Pa(S). One way to see this is that the conditional covariance matrix of $\{z\} \cup S$ given $Pa(S)$ in $P_{S \cup W}$ is the Schur complement of $\Sigma_{\{z\} \cup S}$ in Σ, where Σ is the covariance matrix of $\{z\} \cup S \cup Pa(S)$ in $P_{S \cup W}$ and $\Sigma_{z \cup S}$ is the entries of Σ for $\{z\} \cup S$. If we rearrange the constraint outputted by Algorithm 2 to read $b_w - a_w * A^{-1}_{\text{right}} * b = 0$, then we see that it is simply stating the conditional independence constraint. □

Lemma 3. *Any dormant independence, $x \perp\!\!\!\perp y|w, do(Z)$, with x and y singletons has an equivalent Q-constraint.*

Proof. Let $MACS(Z)$ denote the *maximal ancestral confounded set* of Z (Shpitser and Pearl, 2008), the maximal set in which $MACS(Z) = Anc(Z)_{G(MACS(Z))} = C(Z)_{G(MACS(Z))}$, where $G(MACS(Z))$ is the subgraph of G containing only the variables in $MACS(Z)$ and $C(Z)$ is the c-component of Z.

According to Theorem 6 of Shpitser and Pearl (2008), there exists a dormant independence between singletons, x and y, if and only if x is not a parent of $MACS(y)$, y is not a parent of $MACS(x)$, and there is no bidirected arc between $MACS(x)$ and $MACS(y)$. In this case, $x \perp\!\!\!\perp y|do(Pa(MACS(x) \cup MACS(y)), (MACS(x) \cup (MACS(y) \setminus \{x, y\}$. Now, it is not hard to show using results from (Tian, 2002) that $Q_{MACS(y)}$ is identifiable. Further, since there is no bidirected arc between $MACS(x)$ and $MACS(y)$, it is possible to identify $Q_{MACS(y)}$ without marginalizing over x. Finally, we know that $x \notin Pa(MACS(y))$ so we obtain the Q-constraint, $Q_{MACS(y)}$ is not a function of x.

Now, we will show that the Q-constraint, $Q_{MACS(y)} \perp x$ also implies the dormant independence, $x \perp\!\!\!\perp y|do(Pa(MACS(x) \cup MACS(y)), (MACS(x) \cup (MACS(y) \setminus \{x, y\}$. In proving Theorem 5, we showed that $Q_{MACS(y)} \perp x$ implies that $y \perp\!\!\!\perp x|Pa(S)$ in some distribution, $Q_{S \cup W}$, where $y \in S$ and $x \in W$. Recalling that a Q-factor is just an interventional distribution, we have a dormant independence between x and y. Since Theorem 6 of (Shpitser and Pearl, 2008) gives a necessary and sufficient condition for dormant independence between x and y, we also have that $x \perp\!\!\!\perp y|do(Pa(MACS(x) \cup MACS(y)), (MACS(x) \cup (MACS(y) \setminus \{x, y\}$. □

References

FOYGEL, R., DRAISMA, J. and DRTON, M. (2012). Half-trek criterion for generic identifiability of linear structural equation models. *The Annals of Statistics* **40** 1682–1713.

SHPITSER, I. (2008). *Complete Identification Methods for Causal Inference*. Ph.D. thesis, Computer Science Department, University of California, Los Angeles, CA.

SHPITSER, I. and PEARL, J. (2008). Dormant independence. In *Proceedings of the Twenty-Third Conference on Artificial Intelligence*. AAAI Press, Menlo Park, CA, 1081–1087.

TIAN, J. (2002). *Studies in Causal Reasoning and Learning*. Ph.D. thesis, Computer Science Department, University of California, Los Angeles, CA.