TECHNICAL REPORT
R-329-APPENDUM
May 2007

# Appendum to Identification of Conditional Interventional Distributions


**Ilya Shpitser**
Cognitive Systems Laboratory
Department of Computer Science
University of California, Los Angeles
Los Angeles, CA. 90095
$ilyas@cs.ucla.edu$


function **c-identify**(C, T, Q[T])
INPUT: $T, C \subseteq T$ are both are C-components,
$Q[T]$ a probability distribution
OUTPUT: Expression for $Q[C]$ in terms of $Q[T]$ or **FAIL**

let $A = An(C)_{G_T}$

1 if $A = C$, return $\sum_{T \setminus C} P$

2 if $A = T$, return **FAIL**

3 if $C \subset A \subset T$, there exists a C-component $T'$ such that $C \subset T' \subset A$. return **c-identify**$(C, T', Q[T'])$ ($Q[T']$ is known to be computable from $\sum_{T \setminus A} Q[T]$)

Figure 1: A C-component identification algorithm from [Tian, 2004].

function **cond-identify**(**y**, **x**, **z**, P, G)
INPUT: **x**,**y**,**z** value assignments, P a probability distribution, G a causal diagram (an I-map of P).
OUTPUT: Expression for $P_{\mathbf{x}}(\mathbf{y}|\mathbf{z})$ in terms of P or **FAIL**.

1 let $D = An(\mathbf{Y} \cup \mathbf{Z})_{G_{\underline{\mathbf{X}}}}, F = D \setminus (\mathbf{Y} \cup \mathbf{Z})$

2 assume $C(D) = \{D_1, ..., D_k\}$

3 let $N = \{D_i | \textbf{c-identify}(D_i, C_{D_i}, Q[C_{D_i}]) = \textbf{FAIL}\}$

4 if $N = \emptyset$, return $\frac{\sum_f \prod_i Q[D_i]}{\sum_{y,f} \prod_i Q[D_i]}$

5 let $F_0 = F \cap (\bigcup_{D_i \in N} Pa(D_i)), I = C(D) \setminus N$

6 remove the set $\{D_i | Pa(D_i) \cap F_0 \neq \emptyset\}$ from $I$ and add it to $I_0$ (which is initially empty)

7 let $B = (F \setminus F_0) \cap \bigcup_{D_i \in I_0} Pa(D_i)$

8 if $B \neq \emptyset$, add all nodes in $B$ to $F_0$, and go to line 6

9 if $\mathbf{Y} \cap (\bigcup_{D_i \in (N \cup I_0)} Pa(D_i)) \neq \emptyset$, return **FAIL**,

else return $\frac{\sum_{f_1} \prod_{D_i \in I_1} Q[D_i]}{\sum_{y,f_1} \prod_{D_i \in I_1} Q[D_i]}$

Figure 2: An identification algorithm from [Tian, 2004]. For each $D_i$, we denote $C_{D_i} \in C(G)$ such that $D_i \subseteq C_{D_i}$.

The following algorithm, **cond-identify**, appears in [Tian, 2004]. A proof in [Shpitser & Pearl, 2006a] claims this algorithm is not sound, but was based on a misunderstanding of notation. In fact, this algorithm can be shown to be complete.

**Theorem 1** ***cond-identify*** *is complete.*

*Proof:* We want to show that whenever the algorithm fails, the corresponding effect is not identifiable. The operation of the algorithm can be thought of as follows – it starts with a set of 'bad' C-components (containing a non-identifiable effect). These 'bad' C-components 'infect' other C-components which were initially 'good' (identifiable). The 'infection' proceeds until no more C-components can be 'infected' or until we encounter the set **Y** of effect variables. In the latter case the algorithm fails.

Our proof is by induction.

A known result in [Shpitser & Pearl, 2006b] is that if **identify** fails to identify $D_i$ from $C_i$ then $D_i$ is not identifiable. If the algorithm fails, such a $D_i$ is guaranteed to exist, and is in the ancestor set of $\mathbf{Y} \cup \mathbf{Z}$. Fix such a $D_i$.

Fix the minimal set $\mathbf{W} \subseteq \mathbf{Y} \cup \mathbf{Z}$ such that $D_i \in An(\mathbf{W})_{G_{\underline{\mathbf{y},\mathbf{z}}}}$. Our base case will be that $\mathbf{Y} \cap \mathbf{W} \neq \emptyset$. Note that because $D_i$ is a C-component, every element in $\mathbf{Z} \cap \mathbf{W}$ has a backdoor path to $\mathbf{Y} \cap \mathbf{W}$. Then our conclusion follows by the backdoor hedge criterion [Shpitser & Pearl, 2006a].

Assume $\mathbf{Y} \cap \mathbf{W} = \emptyset$. Since the algorithm failed, $\mathbf{Y}$ intersects the set of 'infected' C-components. We want to show that $\mathbf{W}$ has a backdoor path to $\mathbf{Y}$, which follows the 'infected' C-components. Our conclusion will then follow by the backdoor hedge criterion.

Without loss of generality, assume all variables outside $C_i$ are observable. That means all C-components that are not $C_i$ contain one variable. We prove inductively that all 'infected' nodes are d-connected. The base case is nodes in in $D_i$. Note that each node in $D_i$ has a descendant in $\mathbf{Z}$. Since $D_i$ is a C-component, each pair of nodes in $D_i$ are d-connected by a bidirected path.

For the inductive hypothesis, consider a set of infected nodes $N$ which are in the ancestor set of $\mathbf{Z}$ (but not $\mathbf{Y}$) and which are pairwise d-connected. A new node $I$ can become 'infected' in one of three ways:

If $I$ and $N$ share a parent which is not in $\mathbf{Z} \cup \mathbf{Y}$, that means some node in $N$ has a d-connected path to $I$ through the common parent. Since this node in $N$ is an ancestor of $\mathbf{Z}$, this path can be extended to a d-connected path to any node in $N$. If $I$ is a parent of some node in $N$, that node has a d-connected path to $I$. Moreover, since that node is an ancestor of $\mathbf{Z}$, that path can be extended to a d-connected path to any node in $N$. If some node in $N$ is a parent of $I$, the reasoning is the same.

$I$ itself can either be in $\mathbf{Y}$, an ancestor of $\mathbf{Y}$, or an ancestor of $\mathbf{Z}$. In the first case, we are done since we constructed a d-connected path from a parent of $\mathbf{Z}$ to a node in $\mathbf{Y}$, which translates into a backdoor path from $\mathbf{Z}$ to $\mathbf{Y}$. In the second case, the d-connected path from a parent of $\mathbf{Z}$ to an ancestor of $\mathbf{Y}$ easily extends to a backdoor path from $\mathbf{Z}$ to $\mathbf{Y}$. In the last case, we simply continue the induction until we reach either of the first two cases. We know we reach these cases eventually since the algorithm failed. □

## References


[Shpitser & Pearl, 2006a] Shpitser, I., and Pearl, J. 2006a. Identification of conditional interventional distributions. In *Uncertainty in Artificial Intelligence*, volume 22.

[Shpitser & Pearl, 2006b] Shpitser, I., and Pearl, J. 2006b. Identification of joint interventional distributions in recursive semi-markovian causal models. In *Twenty-First National Conference on Artificial Intelligence*.

[Tian, 2004] Tian, J. 2004. Identifying conditional causal effects. In *Conference on Uncertainty in Artificial Intelligence (UAI)*.