4.3 When is the Effect of an Action Identifiable?

In Chapter 3 we developed several graphical criteria for recognizing when the effect of one variable on another, $P(y|do(x))$, is identifiable in the presence of unmeasured variables. These criteria, like the back-door (Theorem 3.3.2) and front-door (Theorem 3.3.4), are special cases of a more general class of semi-Markovian models for which repeated application of the inference rules of *do* calculus (Theorem 3.4.1) will reduce $P(y|\hat{x})$ to a hat-free expression, thus rendering it identifiable. The semi-Markovian model of Figure 3.1 (or Figure 3.8(f)) is an example where direct application of either the back-door or front-door criterion would not be sufficient for identifying $P(y|\hat{x})$ and yet the expression is reducible (hence identifiable) by a sequence of inference rules of Theorem 3.4.1. In this section we establish a complete characterization of the class of models in which the causal effect $P(y|\hat{x})$ is identifiable in *do* calculus.

4.3.1 Graphical Conditions for Identification

Theorem 4.3.1 characterizes the class of "*do*-identifiable" models in the form of four graphical conditions, anyone of which is sufficient for the identification of $P(y|\hat{x})$ when X and Y are singleton nodes in the graph. Theorem 4.3.2 then asserts the completeness (or necessity) of these four conditions; one of which must hold in the model for $P(y|\hat{x})$ to be identifiable in *do* calculus. Whether these four conditions are necessary in general (in accordance with the semantics of Definition 3.2.4) depends on whether the inference rules of *do* calculus are complete. This question, to the best of my knowledge, is still open.

Theorem 4.3.1 (Galles and Pearl 1995)
Let X and Y denote two singleton variables in a semi-Markovian model characterized by graph G. A sufficient condition for the identifiability of $P(y|\hat{x})$ is that G satisfy one of the following four conditions.

1. *There is no back-door path from X to Y in G, that is; $(X \perp\!\!\!\perp Y)_{G_{\underline{X}}}$.*

2. *There is no directed path from X to Y in G.*

3. *There exists a set of nodes B that blocks all back-door paths from X to Y so that $P(b|\hat{x})$ is identifiable. (A special case of this condition occurs when B consists entirely of nondescendants of X, in which case $P(b|\hat{x})$ reduces immediately to $P(b)$.)*

4. *There exist sets of nodes Z_1 and Z_2 such that:*

   (i) *Z_1 blocks every directed path from X to Y (i.e., $(Y \perp\!\!\!\perp X|Z_1)_{G_{\overline{Z_1}\,\overline{X}}}$);*

   (ii) *Z_2 blocks all back-door paths between Z_1 and Y (i.e., $(Y \perp\!\!\!\perp Z_1|Z_2)_{G_{\overline{X}\underline{Z_1}}}$);*

   (iii) *Z_2 blocks all back-door paths between X and Z_1 (i.e., $(X \perp\!\!\!\perp Z_1|Z_2)_{G_{\underline{X}}}$); and*

   (iv) *Z_2 does not activate any back-door paths from X to Y (i.e., $(X \perp\!\!\!\perp Y|Z_1, Z_2)_{G_{\overline{Z_1}\,\overline{X(Z_2)}}}$). (This condition holds if* (i)–(iii) *are met and no member of Z_2 is a descendant of X.)*

(A special case of condition 4 occurs when $Z_2 = \emptyset$ and there is no back-door path from X to Z_1 or from Z_1 to Y.)

Proof

Condition 1. This condition follows directly from Rule 2 (see Theorem 3.4.1). If $(Y \perp\!\!\!\perp X)_{G_{\underline{X}}}$ then we can immediately change $P(y|\hat{x})$ to $P(y|x)$, so the query is identifiable.

Condition 2. If there is no directed path from X to Y in G, then $(Y \perp\!\!\!\perp X)_{G_{\overline{X}}}$. Hence, by Rule 3, $P(y|\hat{x}) = P(y)$ and so the query is identifiable.

Condition 3. If there is a set of nodes B that blocks all back-door paths from X to Y (i.e. $(Y \perp\!\!\!\perp X|B)_{G_{\underline{X}}}$), then we expand $P(y|\hat{x})$ as $\sum_b P(y|\hat{x}, b)P(b|\hat{x})$ and, by Rule 2, rewrite $P(y|\hat{x}, b)$ as $P(y|x, b)$. If the query $(b|\hat{x})$ is identifiable, then the original query must also be identifiable. See examples in Figure 4.1.

Condition 4. If there is a set of nodes Z_1 that block all directed paths from X to Y and a set of nodes Z_2 that block all back-door paths between Y and Z_1 in $G_{\overline{X}}$, then we expand $P(y|\hat{x}) =$

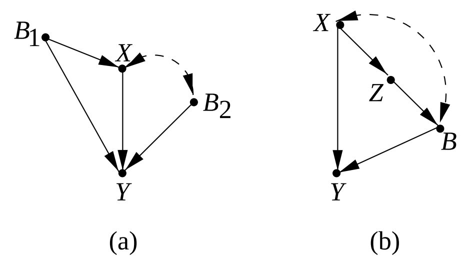

Figure 4.1: Condition 3 of Theorem 4.3.1. In (a), the set $\{B_1, B_2\}$ blocks all back-door paths from X to Y, and $P(b_1, b_2|\hat{x}) = P(b_1, b_2)$. In (b), the node B blocks all back-door paths from X to Y, and $P(b|\hat{x})$ is identifiable using condition 4.

$\sum_{z_1,z_2} P(y|\hat{x}, z_1, z_2)P(z_1, z_2|\hat{x})$ and rewrite $P(y|\hat{x}, z_1, z_2)$ as $P(y|\hat{x}, \hat{z}_1, z_2)$ using Rule 2, since all back-door paths between Z_1 and Y are blocked by Z_2 in $G_{\overline{X}}$. We can reduce $P(y|\hat{x}, \hat{z}_1, z_2)$ to $P(y|\hat{z}_1, z_2)$ using Rule 3, since $(Y \perp\!\!\!\perp X|Z_1, Z_2)_{G_{\overline{Z_1}\ \overline{X(Z_2)}}}$. We can rewrite $P(y|\hat{z}_1, z_2)$ as $P(y|z_1, z_2)$ if $(Y \perp\!\!\!\perp Z_1|Z_2)_{G_{\underline{Z_1}}}$. The only way that this independence cannot hold is if there is a path from Y to Z_1 through X, since $(Y \perp\!\!\!\perp Z_1|Z_2)_{G_{\overline{X}\underline{Z_1}}}$. However, we can block this path by conditioning and summing over X, and so derive $\sum_{x'} P(y|\hat{z}_1, z_2, x')P(x'|\hat{z}_1, z_2)$. Now we can rewrite $P(y|\hat{z}_1, z_2, x')$ as $P(y|z_1, z_2, x')$ using Rule 2. The $P(x'|\hat{z}_1, z_2)$ term can be rewritten as $P(x'|z_2)$ using Rule 3, since Z_1 is a child of X and the graph is acyclic. The query can therefore be rewritten as $\sum_{z_1,z_2} \sum_{x'} P(y|z_1, z_2, x')P(x'|z_2)P(z_1, z_2|\hat{x})$, and we have $P(z_1, z_2|\hat{x}) = P(z_2|\hat{x})P(z_1|\hat{x}, z_2)$. Since Z_2 consists of nondescendants of X, we can rewrite $P(z_2|\hat{x})$ as $P(z_2)$ using Rule 3. Since Z_2 blocks all back-door paths from X to Z_1, we can rewrite $P(z_1|\hat{x}, z_2)$ as $P(z_1|x, z_2)$ using Rule 2. The entire query can thus be rewritten as $\sum_{z_1,z_2} \sum_{x'} P(y|z_1, z_2, x')P(x'|z_2)P(z_1|x, z_2)P(z_2)$. See examples in Figure 4.2. □

Theorem 4.3.2 *The four conditions of Theorem 4.3.1 are necessary for identifiability in do calculus. That is, if all four conditions of The-*

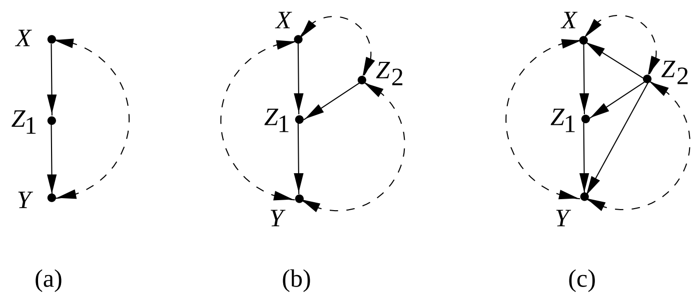

Figure 4.2: Condition 4 of Theorem 4.3.1. In (a), Z_1 blocks all directed paths from X to Y, and the empty set blocks all back-door paths from Z_1 to Y in $G_{\overline{X}}$ and all back-door paths from X to Z_1 in G. In (b) and (c), Z_1 blocks all directed paths from X to Y, and Z_2 blocks all back-door paths from Z_1 to Y in $G_{\overline{X}}$ and all back-door paths from X to Z_1 in G.

orem 4.3.1 fail in a graph G, then there exists no finite sequence of inference rules that reduces $P(y|\hat{x})$ to a hat-free expression.

A proof of Theorem 4.3.2 is given in Galles and Pearl (1995).

4.3.2 Remarks on Efficiency

In implementing Theorem 4.3.1 as a systematic method for determining identifiability, conditions 3 and 4 would seem to require exhaustive search. In order to prove that condition 3 does not hold, for instance, we need to prove that no such blocking set B can exist. Fortunately, the following theorems allow us to significantly prune the search space so as to render the test tractable.

Theorem 4.3.3 *If $P(b_i|\hat{x})$ is identifiable for one minimal set B_i, then $P(b_j|\hat{x})$ is identifiable for any other minimal set B_j.*

Theorem 4.3.3 allows us to test condition 3 with a single minimal blocking set B. If B meets the requirements of condition 3 then the query is identifiable; otherwise, condition 3 cannot be satisfied. In proving this theorem, we use the following lemma.

Lemma 4.3.4 *If the query $P(y|\hat{x})$ is identifiable and if a set of nodes Z lies on a directed path from X to Y, then the query $P(z|\hat{x})$ is identifiable.*

Theorem 4.3.5 *Let Y_1 and Y_2 be two subsets of nodes such that either* (i) *no nodes Y_1 are descendants of X or* (ii) *all nodes Y_1 and Y_2 are descendants of X and all nodes Y_1 are nondescendants of Y_2. A reducing sequence for $P(y_1, y_2|\hat{x})$ exists (per Corollary 3.4.2) if and only if there are reducing sequences for both $P(y_1|\hat{x})$ and $P(y_2|\hat{x}, y_1)$.*

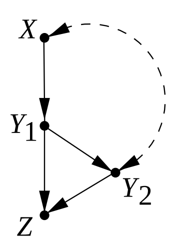

Figure 4.3: Theorem 4.3.1 ensures a reducing sequence for $P(y_2|\hat{x}, y_1)$ and $P(y_1|\hat{x})$, although none exists for $P(y_1|\hat{x}, y_2)$

The probability $P(y_1, y_2|\hat{x})$ might pass the test in Theorem 4.3.1 if we apply the procedure to both $P(y_2|\hat{x}, y_1)$ and $P(y_1|\hat{x})$, but if we try to apply the test to $P(y_1|\hat{x}, y_2)$ then we will not find a reducing sequence of rules. Figure 4.3 shows just such an example. Theorem 4.3.5 guarantees that, if there is a reducing sequence for $P(y_1, y_2|\hat{x})$, then we should always be able to find such a sequence for both $P(y_1|\hat{x})$ and $P(y_2|\hat{x}, y_1)$ by proper choice of Y_1.

Theorem 4.3.6 *If there exists a set Z_1 that meets all of the requirements for Z_1 in condition 4, then the set consisting of the children of X intersected with the ancestors of Y will also meet all of the requirements for Z_1 in condition 4.*

Theorem 4.3.6 removes the need to search for Z_1 in condition 4 of Theorem 4.3.1. Proofs of Theorems 4.3.3–4.3.6 are given in Galles and Pearl (1995).

4.3.3 Deriving a Closed-Form Expression for Control Queries

The algorithm defined by Theorem 4.3.1 not only determines the identifiability of a control query but also provides a closed-form expression for $P(y|\hat{x})$ in terms of the observed probability distribution (when such a closed form exists) as follows.

Function: $\text{ClosedForm}(P(y|\hat{x}))$.

Input: Control query of the form $P(y|\hat{x})$.

Output: Either a closed-form expression for $P(y|\hat{x})$, in terms of observed variables only, or FAIL when query is not identifiable

1. If $(X \perp\!\!\!\perp Y)_{G_{\overline{X}}}$ then return $P(y)$.
2. Otherwise, if $(X \perp\!\!\!\perp Y)_{G_{\underline{X}}}$ then return $P(y|x)$.
3. Otherwise, let $B = \text{BlockingSet}(X, Y)$, and $Pb = \text{ClosedForm}(b|\hat{x})$; if $Pb \neq \text{FAIL}$, then return $\sum_b P(y|b, x) * Pb$.
4. Otherwise, let $Z_1 = \text{Children}(X) \cap (Y \cup \text{Ancestors}(Y))$, $Z_3 = \text{BlockingSet}(X, Z_1)$, $Z_4 = \text{BlockingSet}(Z_1, Y)$, and $Z_2 = Z_3 \cup Z_4$; if $Y \notin Z_1$ and $X \notin Z_2$ then return $\sum_{z_1,z_2} \sum_{x'} P(y|z_1, z_2, x')P(x'|z_2)P(z_1|x, z_2)P(z_2)$.
5. Otherwise, return FAIL.

Steps 3 and 4 invoke the function $\text{BlockingSet}(X, Y)$, which selects a set of nodes Z that d-separate X from Y. Such sets can be found in polynomial time (Tian et al. 1998). Step 3 contains a recursive call to the algorithm $\text{ClosedForm}(b|\hat{x})$ itself, in order to obtain an expression for causal effect $P(b|\hat{x})$.

4.3.4 Summary

The conditions of Theorem 4.3.1 sharply delineate the boundary between the class of identifying models (such as those depicted in Figure

3.8) and nonidentifying models (Figure 3.9). These conditions lead to an effective algorithm for determining the identifiability of control queries of the type $P(y|\hat{x})$, where X is a single variable. Such queries are identifiable in *do* calculus if and only if they meet the conditions of Theorem 4.3.1. The algorithm further gives a closed-form expression for the causal effect $P(y|\hat{x})$ in terms of estimable probabilities.

Applications to causal analysis of nonexperimental data in the social and medical sciences are discussed in Chapter 3 and further elaborated in Chapters 5 and 6. In Chapter 9 (Corollary 9.2.17) we will apply these results to problems of *causal attribution*, that is, to estimate the probability that a specific observation (e.g., a disease case) is causally attributable to a given event (e.g., exposure).