Contents

1 Integrated inferences

In this chapter we generalize the model developed in Chapter **??** to research situations in which we have data on multiple cases.

We start with a conceptual point: the structure introduced in Chapter 6 for single-case analysis can be used *as is* for multi-case analysis. Thus, the conceptual work for mixed methods inference from models has been done already. Our goal for the rest of the chapter is thus more technical than conceptual—to show how to shift focus beyond sample level queries and to exploit assumptions regarding independence across cases to generate simpler models of causal processes that affect many units. As we do so, we provide microfoundations for the models in Chapter **??** (as with those in **?**) with the probative value of clues derivable from a causal structure and data rather than provided directly by researchers.

1.1 Sample inference

Conceptualized correctly, there is no deep difference between the logic of inference used in single-case and in multi-case studies. This is not because any single "case" can be disaggregated into many "cases," thereby allowing for large n analysis on small problems (**?**). Rather, the opposite: fundamentally, model-based inference always involves comparing *a* pattern of data with the logic of the model. Studies with multiple cases can, in fact, be conceptualized as single-case studies: we always draw our inferences from a single *collection* of clues, whether those clues have come from one or from many units.

In practice, when we move from a causal model with one observation to a causal model with multiple observations, we can use the structure we introduced in Chapter **??** but simply replace nodes that have a single value (i.e., scalars) with nodes containing multiple values (i.e., vectors) drawn from multiple cases. We then make inferences about causal relations between nodes from seeing the values of those nodes' (or other nodes') vectors.

To illustrate, consider the following situation. Suppose that our model includes a binary treatment X that is assigned to 1 with probability 0.5; an outcome, Y; and a third "clue" variable, K, all observable. We posit an unobserved variable θ^Y, representing Y's nodal type, with θ^Y taking on values in $\{a, b, c, d\}$ with equal probability. (We interpret the types in $\{a, b, c, d\}$ as defined in Section **??**.) In addition to pointing into Y, moreover, θ^Y affects K. In particular, $K = 1$ whenever X has an effect on Y, while $K = 1$ with a 50% probability otherwise. In other words, our clue K is informative about θ^Y, a unit's nodal type for Y. As familiar from Chapters **??** and **??**, when we observe K in a case we can update on causal effects within the case since that K value will have different likelihoods under different values of θ^Y.

So far, we have described the problem at the unit level. Let's now consider a two-case setup. We do this by exchanging scalar nodes for vectors:

- We have a treatment node, X, that can take on one of four values, $(0, 0), (0, 1), (1, 0), (1, 1)$ with equal probability.
- θ^Y is now a vector with two elements that can take on one of 16 values $(a, a), (a, b), ... (d, d)$ as determined by λ_θ. We might imagine a uniform distribution over these 16 elements.
- Y is a vector that is generated by θ^Y and X in the obvious way (e.g., $X = (0, 0), \theta^Y = (a, b)$ generates outcomes $Y = (1, 0)$)
- The vector K has the same domain as X and Y, and element $K[j] = 1$ if $\theta^Y[j] = b$.

Now, consider a causal estimand. In a single-case setup, we might ask whether X has an effect on Y in the case. For a multi-case setup, we might ask what the Sample Average Treatment Effect, τ, is. Note a subtle difference in the nature of the answers we seek in these two situations. In the first (single-case) instance, our estimand is binary—of the form: "is the case a b type?"—and our answer is a probability. In the multi-case estimation of the sample average treatment effect ("SATE"), our estimand is categorical and our answer is a probability distribution: we are asking "what is the probability that τ is 0?," "what is the probability that τ is .5?", and so on.

While the estimand shifts, we can use the tools introduced for single-case process tracing in Chapters **??** and **??** to analyze this (superficially) multi-case study. To begin, our prior on the probability that $\tau = 1$ is the prior that X has a positive effect on Y in both cases, that is, that $\theta^Y = (b, b)$: just 1 in 16.

Now, suppose that we observe that, for both units, $X = 1$ and $Y = 1$. This data pattern is consistent only with four possible θ vectors: $(b, b), (d, d), (b, d), (d, b)$. Moreover each of these four is equally likely to produce the data patter we see). So our belief that $\tau = 1$ now shifts from 1 in 16 to to 1 in 4. Next, suppose that we further observe the data pattern $\mathbf{K} = (1, 1)$. The probability of this pattern for Θ vector (b, b) $(\tau = 1)$ is 1. And for the type vectors $(d, d), (b, d), (d, b)$, the probability of this $\mathbf{K}$ pattern is .25, .5, and .5, respectively. Applying Bayes' rule, our updated belief that $\tau = 1$ is then $1/(1 + .25 + .5 + .5) = 4/9$.

We can similarly figure out the posterior probability on any possible value of τ and so build up a full posterior distribution. And we can do so given any $\mathbf{K}$ pattern (i.e., $\mathbf{K}$ realization) across the cases. Thus, if we observe the data pattern $\mathbf{K} = (0, 1)$, the probability of this pattern for type vector (b, b) $(\tau = 1)$ is 0. For the type vectors $(d, d), (b, d), (d, b)$ it is .25, 0, .5, respectively. The table below represents the posterior distribution over a set of discrete treatment effect values given different K patterns observed.

X pattern	Y pattern	K pattern	$\tau = -1$	$\tau = -.5$	$\tau = 0$	$\tau = .5$	$\tau = 1$
(1,1)	(1,1)	(1,1)	0	0	1/9	4/9	4/9
(1,1)	(1,1)	(1,0)	0	0	1/3	2/3	0
(1,1)	(1,1)	(0,0)	0	0	1	0	0

The conceptual point is that the general logic of inference with multiple units is the same as that with one unit. In both situations, we work out the likelihood of any given data *pattern* for each possible set of values of model parameters and update our beliefs about those parameters. And, from our posterior distribution over model parameters (e.g., Θ^Y), we then derive a posterior distribution over the possible answers to our query (e.g., values of τ).[1]

1.2 General queries

Although the core conceptual logic is the same for multi-case and single-case inference, going forward, we operationalize these problems somewhat differently.

For the remainder of this chapter, and for the rest of the book, when we focus on multi-case studies, we will set our sights primarily on models that describe general processes. Rather than seeking to understand the average effect in a set of cases, we seek to understand the causal relations that gave rise to the set of cases. From these we sometimes draw inferences to cases but in general our models will involve queries pitched in general terms.

There are two reasons for this. The first is that we are interested in learning across cases: To figure out how what we see in one case provides insight for what is happening in another. We do this by using data on some cases to update our beliefs about a general model that we think is of relevance for other cases. Thus we seek to learn about a general model. The second reason is more practical. If we can think of units as draws from a large population, and then invoke independence assumptions across types, then we can greatly reduce complexity by analyzing problems at the unit level rather than at the population level. In the 2-case example above, the vector θ^Y could take on any of 16 values $((a, a), (a, b), \ldots (d, d))$. At the case level, however, the

[1]Representing node values in vector forms like this allows for vector-level mappings that imply more complex dependencies between units. For instance we might imagine instead that we observe $K = 1$ if and only if $\theta^Y = (b, b)$, in which case observation of K lets us distinguish between $\tau = 1$ and $\tau = .5$ but not between $\tau = .5$ and $\tau = 0$.

node θ^Y can take on only 4 values ($\{a, b, c, d\}$), yet we can learn about each case's θ^Y value from data drawn from all the cases. Thinking about it this way simplifies the problem by greatly reducing the parameter space, but it is not free. It requires invoking the assumption that (potential) outcomes across units do not depend on each other. If we cannot stand by that assumption, then we will need to build independence failures into our models.

Taking this step, the procedure we now use in the mixed methods works as follows.

1.2.1 Set up

1. **A DAG**. As for process tracing, we begin with a graphical causal model specifying possible causal linkages between nodes. Our "chain" model for instance has DAG: $X \rightarrow M \rightarrow Y$.

2. **Nodal types**. Just as in process tracing, the DAG and variable ranges define the set of possible nodal types in the model—the possible ways in which each variable is assigned (if exogenous) or determined by its parents (if endogenous). For the $X \rightarrow M \rightarrow Y$ model there are 2 types for θ^X, 4 for θ^M, and 4 for θ^Y.

3. **Causal types**. A full set of nodal types gives rise to a full set of causal types, encompassing all possible combinations of nodal types across all nodes in the model. We let θ denote an arbitrary causal type. For a $X \rightarrow M \rightarrow Y$ model, one possible causal type would be $\theta = (\theta^X_1, \theta^M_{01}, \theta^M_{01})$.

4. **Parameters.** As before, we use λ^V to denote the probabilities of θ^V for a given node, V. Recall that in process tracing, we sought to learn about θ and our priors were given by λ. When we shift to multi-case inference, λ becomes the parameter that we want to learn about: we seek to learn about the probability of different types arising in a population (or the *shares* of types in a large population).

5. **Priors**. In the process tracing setup, we treat λ as given: we do not seek to learn about λ, and uncertainty over λ plays no role. When we get to observe data on multiple cases, however, we have the opportunity to learn *both* about the cases at hand *and* about the population. Moreover, our level of uncertainty about population-level parameters will shape our inferences. We thus want our parameters (the λ's) to be drawn from a prior *distribution* — a distribution that expresses our uncertainty and over which we can update once we see the data. While different distributions may be appropriate to the task in general, uncertainty over proportions (of cases, events, etc.) falling into a set of discrete categories is usefully described by a Dirichlet distribution, as discussed in Chapter **??**. Recall that the parameters of a Dirichlet distribution (the α's) can be thought of as conveying both the relative expected proportions in each category and our degree of uncertainty.

1.2.2 Inference

Inference then works by figuring out the probability of the data given different possible parameter vectors, λs, and then applying Bayes' rule. In practice we proceed as follows.

Distributions over causal types. We first need characterize our beliefs about causal types given any possible parameter vector λ. Imagine a draw of one possible value of λ from the prior. This λ vector implies a set of nodal type shares for all nodes. That set of nodal type shares implies, in turn, a distribution over *causal* types (θ). For instance, the probability of causal type $\theta = (\theta^X_1, \theta^Y_{01}, \theta^M_{01})$ is simply $p(\theta|\lambda) = \lambda^X_1 \lambda^M_{01} \lambda^Y_{01}$. More generally:

$$p(\theta|\lambda) = \prod_{k,v:\theta^v_k \in \theta} \lambda^v_k$$

Event probabilities. Each causal type in turn implies a single data realization, or data type. For instance $\theta = (\theta^X_1, \theta^M_{01}, \theta^Y_{01})$ implies data $X = 1, M = 1, Y = 1$. Let $D(\theta)$ denote the data type implied by causal type

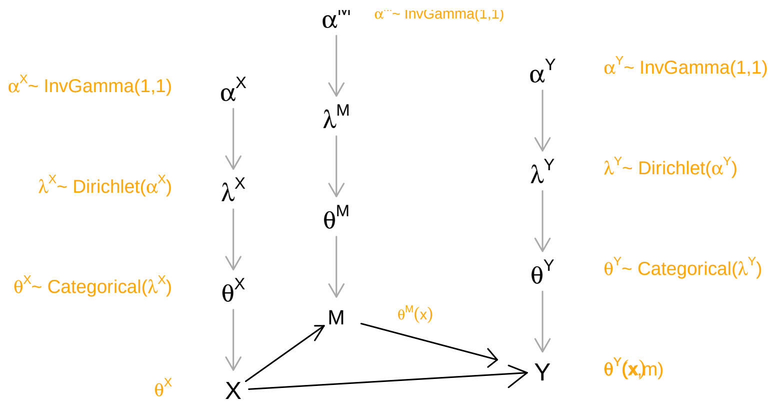

Figure 1: Types, parameters, and priors

θ. A single data type, however, may be implied by multiple causal types. We use $\Theta(d)$ to denote the set of causal types that imply a given data type:

$$\Theta(d) : \{\theta | D(\theta) = d\}$$

The probability of a given data type d, is then:

$$w_d = \sum_{\theta \in \Theta(d)} p(\theta|\lambda)$$

And we use $\mathbf{w}$ to denote the vector of event probabilities over all data types.

To illustrate, a data type $d = (X = 1, M = 1, Y = 1)$ is consistent with four different causal types in the $X \to M \to Y$ model: $\Theta(d) = \{(\theta_0^X, \theta_{01}^M, \theta_{01}^Y), (\theta_0^X, \theta_{11}^M, \theta_{01}^Y), (\theta_0^X, \theta_{01}^M, \theta_{11}^Y), (\theta_0^X, \theta_{11}^M, \theta_{11}^Y)\}$. The probability of the data type is then calculated by summing up the probabilities of each causal type that implies the event: $w_{111} := \lambda_1^X(\lambda_{01}^M + \lambda_{11}^M))(\lambda_{01}^Y + \lambda_{11}^Y)$.

In practice, calculating the full $\mathbf{w}$ vector is made easier by the construction of a "parameter matrix" and an "ambiguity matrix", just as for process tracing, that tells us which causal types are consistent with a particular data type.

We use Tables **??** and **??** to illustrate how to calculate the event probability for each data type for a given parameter vector λ. Starting with data type $X = 0, Y = 0$ (first column of the ambiguity matrix), we see that the consistent causal types are $(\theta_0^X, \theta_{00}^Y)$ and $(\theta_0^X, \theta_{01}^Y)$, in rows 1 and 4. We then turn to columns 1 and 4 of the parameter matrix to read off the probability of each of these causal types—in each case given by the probability of the nodal types that it is formed out of. This gives $.4 \times .3$ and $.4 \times .2$ giving a total probability of 0.2 for the $X = 0, Y = 0$ event. All four event probabilities, for the four data types, are then calculated in the same way.

In practice we do this all using matrix operations.

Table 2: An ambiguity matrix for a simple $X \to Y$ model (with no unobserved confounding). Rows are causal types, columns are data types.

	X0Y0	X1Y0	X0Y1	X1Y1
X0Y00	1	0	0	0
X1Y00	0	1	0	0
X0Y10	0	0	1	0
X1Y10	0	1	0	0
X0Y01	1	0	0	0
X1Y01	0	0	0	1
X0Y11	0	0	1	0
X1Y11	0	0	0	1

Table 3: A parameter matrix for a simple $X \to Y$ model (with no unobserved confounding), indicating a single draw of λ values from the prior distribution.

	X0.Y00	X1.Y00	X0.Y10	X1.Y10	X0.Y01	X1.Y01	X0.Y11	X1.Y11	λ
X.0	1	0	1	0	1	0	1	0	0.4
X.1	0	1	0	1	0	1	0	1	0.6
Y.00	1	1	0	0	0	0	0	0	0.3
Y.10	0	0	1	1	0	0	0	0	0.2
Y.01	0	0	0	0	1	1	0	0	0.2
Y.11	0	0	0	0	0	0	1	1	0.3

Likelihood. Now that we know the probability of observing each data pattern in a *single* case given λ, we can use these event probabilities to aggregate up to the likelihood of observing a data pattern across multiple cases (given λ). For this aggregation, we make use of an independence assumption: that each unit is independently drawn from a common distribution. Doing so lets us move from a categorical distribution that gives the probability that a single case has a particular data type to a *multinomial* distribution that gives the probability of seeing an arbitrary data pattern across any number of cases.

Specifically, with discrete variables, we can think of a given multiple-case data pattern simply as a set of counts across categories. For, say, X, Y data, we will observe a certain number of $X = 0, Y = 0$ cases (which we notate as n_{00}), a certain number of $X = 1, Y = 0$ cases (n_{10}), a certain number of $X = 0, Y = 1$ cases (n_{01}), and a certain number of $X = 1, Y = 1$ cases (n_{11}). A data pattern, given a particular set of variables observed (a search strategy), thus has a multinomial distribution. The likelihood of a data pattern under a given search strategy, in turn, takes the form of a multinomial distribution conditional on the number of cases observed, n, and the probability of each data type, given a λ draw. More formally, we write:

$$d \sim \text{Multinomial}(n, w(\lambda))$$

To illustrate, assume now that we have a 3-node model, with X, Y, and M all binary. Let $\mathbf{n}_{XYM}$ denote an 8-element vector recording the number of cases in a sample displaying each possible combination of X, Y, M data, thus: $\mathbf{D} = \mathbf{n}_{XYM} := (n_{000}, n_{001}, n_{100}, \dots, n_{111})$. The elements of $\mathbf{n}_{XYM}$ sum to n, the total number of cases studied. Likewise, let the event probabilities for data types given λ be registered in a vector, $\mathbf{w}_{XYM} = (w_{000}, w_{001}, w_{100}, \dots, w_{111})$. The likelihood of a data pattern, $\mathbf{D}$ is then:

$$p(d|\lambda) = \text{Multinom}\left(n_{XYM} | \sum n_{XYM}, w_{XYM}(\lambda)\right)$$

In other words, the likelihood of observing a particular data pattern given λ is given by the corresponding value of the multinomial distribution given the data probabilities.

Table 4: An illustration of a posterior distribution for a $X \rightarrow M \rightarrow Y$ model. Each row is a draw from $p(\lambda|d)$). SUch a posterior would typically have thousands of rows and capture the full joint posterior distribution over all parameters.

X.0	X.1	M.00	M.10	M.01	M.11	Y.00	Y.10	Y.01	Y.11
0.47	0.53	0.21	0.07	0.17	0.55	0.20	0.23	0.15	0.41
0.68	0.32	0.02	0.41	0.38	0.19	0.12	0.20	0.07	0.61
0.33	0.67	0.16	0.45	0.27	0.12	0.08	0.02	0.81	0.09
0.68	0.32	0.15	0.10	0.70	0.05	0.03	0.07	0.00	0.90
0.17	0.83	0.02	0.11	0.64	0.22	0.44	0.06	0.30	0.20
0.83	0.17	0.16	0.08	0.02	0.73	0.49	0.28	0.12	0.11

Table 5: Inferences on a chain model given different amounts of data (all on the diagonal, with X=0, Y=0 or X=1, Y=1). Columns 1-4 are shares of a, b, c, d causal types (as described in Chapter 2); columns 5 - 8 show τ_{ij}—average effects of i on j; the last column shows the probability of causation: the probability that X caused Y in a $X = Y = 1$ case.

Data	a	b	c	d	$\tau_{XM}$$	$\tau_{MY}$$	$\tau_{XY}$$	PC
No data	0.13	0.13	0.37	0.38	0.00	-0.01	0.00	0.27
2 cases X, Y data only	0.12	0.14	0.37	0.37	0.00	0.00	0.02	0.29
2 cases, X, M, Y data	0.12	0.16	0.36	0.36	0.20	0.20	0.04	0.32
10 cases: X, Y data only	0.11	0.27	0.31	0.31	0.00	0.00	0.17	0.45
10 cases: X, M, Y data	0.09	0.44	0.23	0.23	0.59	0.59	0.35	0.66

4. **Estimation**. We now have all the components for updating on λ. Applying Bayes rule (see Chapter **??**), we have:

$$p(\lambda|d) = \frac{p(d|\lambda)p(\lambda)}{\int_{\lambda'} p(d|\lambda')p(\lambda')}$$

In the `CausalQueries` package this updating is implemented in `stan`, and the result of the updating is a dataframe that contains a collection of draws from the posterior distribution for λ. Table **??** illustrates what such a dataframe might look like for an $X \rightarrow M \rightarrow Y$ model. Each row represents a single draw from $p(\lambda|d)$. The 10 columns represent shares for each of the 10 nodal types in the model, under each λ draw.

5. **Querying**.

Once we have generated a posterior distribution for λ, we can then query that distribution. The simplest queries relate to values of λ. For instance, if we are interested in the probability that M has a positive effect on Y, given an updated $X \rightarrow M \rightarrow Y$ model, we want to know about the distribution of λ^M_{01}. This distribution can be read directly from column 9 ($Y01$) of Table **??**. More complex queries can all be described as summaries of combinations of these columns. For instance, the query, "What is the average effect of M on Y" is a question about the distribution of $\lambda^M_{01} - \lambda^M_{10}$, which is given by the difference between columns 9 and 8 of the table. Still more complex queries may require keeping some nodes constant while varying others, yet all of these can be calculated as summaries of the combinations of columns of the posterior distribution, following the rules described in Chapter **??**.

Table **??** shows examples of a full mapping from data to posteriors. We begin with a simple chain model of the form $X \rightarrow M \rightarrow Y$ with flat priors over nodal types and report inferences on a set of queries (columns) for difference data types (rows).

1.2.3 Wrinkles

The procedure we described works in the same way for a very wide class of causal models. More attention is needed for special cases in which there is confounding, complex sampling, or sample estimands.

1.2.3.1 Unobserved confounding. When there is unobserved confounding, we need parameter sets that allow for a joint distribution over nodal types. Unobserved confounding, put simply, means that there is confounding across nodes that is not captured by nodes and edges represented on the DAG. More formally, in the absence of unobserved confounding, we can treat the distribution of nodal types for a given node as independent of the distribution of nodal types for every other node. Unobserved confounding means that we believe that nodal types may be correlated across nodes. Thus, for instance, we might believe that those units assigned to $M = 1$ have different potential outcomes for Y than those assigned to $M = 0$ – i.e., that the probability of $M = 1$ is correlated with whether or not M has an effect on Y. To allow for such a correlation, we have to allow θ^M and θ^Y to have a joint distribution. There are different ways to do this in practice, but a simple approach is to split the parameter set corresponding to the Y node into two: we specify one distribution for θ^Y when $M = 0$ and a separate distribution for θ^Y when $M = 1$. For each of these parameter sets, we specify two α parameters representing our priors. We can draw λ values for these conditional nodal types from the resulting Dirichlet distributions, as above, and can then calculate causal type probabilities in the usual way. Note that if we do this in an $X \to M \to Y$ model, we have one 2-dimensional Dirichlet distribution corresponding to X, one 4-dimensional Dirichlet distribution corresponding to M, and two 4 dimensional distributions corresponding to Y. In all, with 1+3+3+3 degrees of freedom: exactly the number needed to represent a joint distribution over all $\theta^X, \theta^M, \theta^Y$ combinations.

In the figure below we represent this confounding by indicating parameters values λ_{MY} that determine the joint distribution over θ_M and θ_Y.

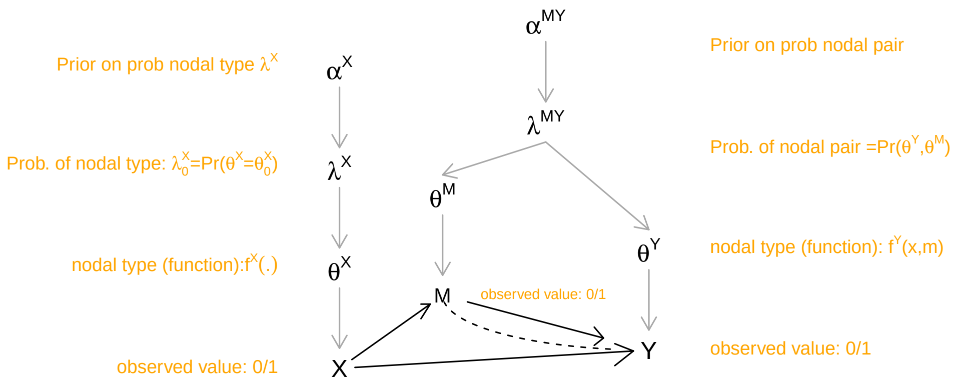
Prior on prob nodal type λ^X
Prob. of nodal type: λ^X_0=Pr(θ^X=θ^X_0)
nodal type (function):f^X(.)
observed value: 0/1
α^X
λ^X
θ^X
X
α^MY
λ^MY
θ^M
M
observed value: 0/1
θ^Y
Y
Prior on prob nodal pair
Prob. of nodal pair =Pr(θ^Y,θ^M)
nodal type (function): f^Y(x,m)
observed value: 0/1

Figure 2: Types, parameters, and priors, with confounding

1.2.3.2 Sampling and the likelihood principle In constructing a likelihood function, we sometimes need to take the sampling strategy into account. Sometimes however we can ignore the sampling procedure if we can invoke the “likelihood principle”—the principle that the relevant information for inference is contained in the likelihood.

To see the likelihood principle in operation, consider the following *conditional* data strategy: we collect data on X and Y in 2 cases, and we then measure M in any case in which we observe $X = 1, Y = 1$.

We draw data and end up with one case with $X = Y = 0$ (M not observed) and one case with $X = 1, M = 0, Y = 1$ (M measured, following the strategy).

One way to think of the event probabilities is to think of a set of 5 possible events, as described in table below:

data type:	prob:
$X1M0Y1$	$\lambda^X_1(\lambda^M_{00} + \lambda^M_{10})(\lambda^Y_{11} + \lambda^Y_{10})$
$X1M1Y1$	$\lambda^X_1(\lambda^M_{11} + \lambda^M_{01})(\lambda^Y_{11} + \lambda^Y_{01})$
$X0Y0$	$\lambda^X_0(\lambda^M_{00} + \lambda^M_{01})(\lambda^Y_{00} + \lambda^Y_{01}) + \lambda^X_0(\lambda^M_{10} + \lambda^M_{11})(\lambda^Y_{00} + \lambda^Y_{10})$
$X0Y1$	$\lambda^X_0(\lambda^M_{00} + \lambda^M_{01})(\lambda^Y_{10} + \lambda^Y_{11}) + \lambda^X_0(\lambda^M_{10} + \lambda^M_{11})(\lambda^Y_{01} + \lambda^Y_{11})$
$X1Y0$	$\lambda^X_1(\lambda^M_{00} + \lambda^M_{10})(\lambda^Y_{00} + \lambda^Y_{01}) + \lambda^X_1(\lambda^M_{01} + \lambda^M_{11})(\lambda^Y_{00} + \lambda^Y_{10})$

In this conditional strategy view (draw X, Y first and then draw M based on what you find) we have

- $2P(X = 0, Y = 0)P(X = 1, Y = 1)P(M = 0|X = 1, Y = 1)$

The two observations could however also be thought of as coming from a simple multinomial draw from the five event types in the table above. Call this the single multinomial view.

In the single multinomial view we have the probability of seeing data with $X = Y = 0$ in one case and $X = 1, M = 0, Y = 1$ in another is:

- $2P(X = 0, Y = 0)P(X = 1, M = 0, Y = 1)$

But since $P(X = 1, Y = 1)P(M = 0|X = 1, Y = 1) = P(M = 0|X = 1, Y = 1)$ these two expressions are the same “up to a constant” and so the inferences we make will be the same under both views.

Consider now a third strategy in which rather than conditioning $X = Y = 1$ to examine M, one of the two cases were chosen at random to observe M and it just so happened to be be a case with $X = Y = 1$. Or another strategy in which for each data point researchers randomly determined whether to gather data on M or not. In all of these cases the probability of observing the data we do in fact observe has the same basic form, albeit with possibly different constants.

In other words, these details of sampling can be ignored.

Other sampling procedures do have to be taken into account however, in particular, sampling—or more generally missingness—that is related to potential outcomes. As the simplest illustration consider a model in which $X \rightarrow Y$, but data is only recorded in cases in which $Y = 1$. Then a naive implementation of our procedure would infer that $Y = 1$ regardless of X and so X has no effect on Y. The problem here is that the likelihood is not taking account of the process through which cases enter our data. In this case the correct likelihood would make use of event probabilities of the form:

$$x_d = \sum_{\theta \in \Theta(d)} p(\theta|\lambda)$$

$$x_d = \sum_{\theta \in \Theta(d)} p(\theta|\lambda)$$

Let D^* denote the set of data types involving Y=1\$. Then:

$$w_d = \begin{cases} 0 & \text{if } d \notin D^* \\ \frac{x_d}{\sum_{d' \in D^*} x_{d'}} & \text{otherwise} \end{cases}$$

While this kind of sampling can be handled relatively easily (it is implemented also in the `CausalQueries` package) the general principle holds that sampling (missingness) that is related to potential outcomes is a part of the data generating process and needs to be taken into account in the likelihood. For strategies to address non random sampling by blocking, see **?**.

1.2.3.3 Case inference following population updating We are often in situations in which we observe patterns in n units and then seek to make an inference about one or more of the n cases conditional on *both* the case level data and the broader patterns in the full data.

Divide cases into set S^0, S^1 where S^0 is the set for which we wish to make case level inferences and S^1 is the collection of other cases for which we have data.

In such cases should one use the data from S^0 when updating on population estimands or rather update using S^1 only and use information on S^1 for the case level inferences only?

The surprising answer is that it is possible to do both, though exactly how queries are calculated depends on the method used.

Let Λ denote a collection of possible population parameters with typical element λ^i. Let p denote a distribution over Λ (after updating on data from set S^1), with typical element λ^i. Let X denote possible data for cases in S^0 with realization x.

Let d^i denote the probability of observing data $X = x$ for a case (or set of cases) given λ^i.

Let $\tau^{|x}$ denote a query of interest—where the query is conditional in the sense that it relates to cases with data x. An example might be: what is the effect of X on Y in a case in which $M = 1$ and $Y = 1$. Let q^i_j denote the probability that $\tau^{|x} = \tau^{|x}_j$ when $\lambda = \lambda^i$ for a case with data $X = x$. Note q^i_j can be written z^i/d^i where $z^i_j = \Pr(\tau^{|x} = \tau^{|x}_j, X = x|\lambda^i)$.

To illustrate say in an $X \rightarrow Y$ model we were interested the effect of X on Y in a case with $X = 1, Y = 1$. Then $d^i = (\lambda^i)^X_1((\lambda^i)^Y_{01} + (\lambda^i)^Y_{11})$ is the probability of observing $((X = 1, Y = 1)$. Then for query $\tau^{|x}_j = 1$ (did X cause Y) we have $z^i_j = (\lambda^i)^X_1((\lambda^i)^Y_{01})$, and so the probability of this query for this case given λ^i is: $q^i_j = \frac{(\lambda^i)^Y_{01}}{(\lambda^i)^Y_{01} + (\lambda^i)^Y_{11}}$

The posterior on τ^{S^0} for the cases in S^0 that provide data x, is then:

$$\Pr(\tau^{|x} = \tau^{|x}_j) = \frac{p^i z^i_j}{\sum_k p^k d^k}$$

This can be calculated from the prior p (that is the distribution on Θ after updating on cases in S^1 only).

Notice however that (a) the *posterior* distribution on λ^i given observation of x in the S^0 set is $\frac{p^i d^i}{\sum_k p^k d^k}$ and (b) $p^i z^i_j = p^i d^i q^i$. It follows that this quantity can also be interpreted as the posterior mean of q^i, after observing both S^0 and S^1.

We therefore have two approaches to calculating these sample quantities: either take the posterior mean (posterior to S^0 and S^1), over the distribution of λ of the conditional probability of the estimand given the case data in S^0, or take the expected probability of τ given the prior (after observing S^1 only) and condition on the probability of the case level data in S^0).

1.3 Mixed methods

As can be seen already from our discussion of sampling, we do not need data on all nodes in order to implement the procedure. If we have data on only some of the nodes in a model, we follow the same basic logic as with partial process-tracing data. In calculating the probability of a pattern of partial data, we use all columns (data types) in the ambiguity matrix that are consistent with the partial data.

So, for instance, if we have an $X \rightarrow Y$ model but observe only $Y = 1$, then we would retain both the $X = 0, Y = 1$ column and the $X = 1, Y = 1$ column. We then calculate the probability of this data type by summing causal-type probabilities for all causal types that can produce *either* $X = 0, Y = 1$ *or* $X = 1, Y = 1$.

What if our data have been collected via a mixture of search strategies? Suppose, for instance, that we have collected X, Y data for a set of cases, and have additionally collected data on M for a random subset of these. We can think of this mixed strategy as akin to conducting quantitative analysis on a large sample while conducting in-depth process tracing on part of the large-N sample. We can then summarize our data in two vectors, an 8-element n_{XYM} vector $((n_{000}, n_{001}, \dots n_{111})$ for the cases with process-tracing (M) observations, and a 4-element vector $n_{XY*} = (n_{00*}, n_{10*}, n_{01*}, n_{11*}$ for the partial data on those cases on which we did not conduct process tracing. Likewise, we now have two sets of data probabilities: an 8-element vector for the set of cases with complete data, w_{XYM}, and a 4-element vector for those with partial data, w_{XY*}.

Let n denote the total number of cases examined, and k the number for which we have data on M. Assuming that each observed case represents an independent, random draw from the population, we can form the likelihood function as a *product* of multinomial distributions, one representing the complete-data (process-traced) cases and one representing those with only X, Y data:

$$\Pr(\mathcal{D}|\theta) = \text{Multinom}\left(n_{XY*}|n-k, w_{XY*}\right) \times \text{Multinom}\left(n_{XYM}|k, w_{XYM}\right)$$

The generalization is straightforward. Say that a strategy is a set of nodes on which data is gathered on n_s units. For example data may be gathered through three strategies: n_1 units for which data is gathered on nodes V_1 only, n_2 units for which data is gathered on nodes V_2 only, and n_3 units for which data is gathered on nodes V_3 only. The observed number of units for each data type under each data strategy is m_s and the event probabilities are w_s. The likelihood is:

$$L = \prod_s \text{multinomial}(m_s|n_s, w_s)$$

1.4 Considerations

In this last section we consider six implications and extensions of this approach.

1.4.1 Probative value can be derived from a causal structure plus data

In Chapter **??**, we discussed the fact that a DAG by itself is insufficient to generate learning about causal effects from data on a single case; we also need informative prior beliefs about population-level shares of nodal types.

When working with multiple cases, however, we *can* learn about causal relations when starting with nothing more than the DAG and data. In particular, we can simultaneously learn about case-level queries and justify our inferences from population-level data patterns.

For instance, in an $X \rightarrow M \rightarrow Y$ model, even if we start with flat priors over M's nodal types, observing a correlation (or no correlation) between X and M across multiple cases provides information about X's effect on M. Simply, a stronger, positive (negative) X, M correlation implies a stronger positive (negative) effect of X on M. In turn, a stronger X, M correlation implies a stronger effect of X on Y since, under this model, that effect has to run through an effect of X on M.

What's more, data from multiple cases can *provide* probative value for within-case inference. Suppose, for the $X \rightarrow M \rightarrow Y$ model, that we start with flat priors over all nodal types. As discussed in Chapter **??**, observing M in a single case cannot be informative about X's effect on Y in that case. If we have no idea of the direction of the intermediate causal effects, then we have no idea which value of M is more consistent with an $X \rightarrow M$ effect or with an $M \rightarrow Y$ effect. But suppose that we first observe data on X and M for a group of cases and find a strong positive correlation between the two variables. We now update to a belief that any effect of X on M is more likely to be positive than negative. Now, let's say we look at one of our other cases in which $X = 1$ and $Y = 1$ and want to know if $X = 1$ caused $Y = 1$. Knowing now that any such effect would most likely have operated via a positive $X \rightarrow M$ effect means that observing M will be informative: seeing $M = 1$ in this case will be more consistent with an $X \rightarrow Y$ effect than will $M = 0$. The same logic, of course, also holds for observing cross-case correlations between M and Y.

Our ability to draw probative value from cross-case data will depend on the causal model we start with. For instance, if our model allows X also to have a direct effect on Y, our ability to learn from M will be more limited. We explore this issue in much greater detail in Chapter **??**.

1.4.2 Learning without identification

Some causal queries are *identified* while others are not. When a query is identified, each true value for the query is associated with a unique data distribution given infinite data. Thus, as we gather more and more data, our posterior on the query should converge on the true value. When a query is not identified, multiple true values of the query will be associated with the same data distribution given infinite data. With a non-identified query, our posterior will never converge on a unique value regardless of how much data we collect since multiple answers will be equally consistent with the data. A key advantage of causal model framework, however, is that we can *learn* about queries that are not identified.

We can illustrate the difference between identified and non-identified causal questions by comparing an ATE query to a probability of causation (PC) query for a simple $X \rightarrow Y$ model. When asking about the ATE, we are asking about the average effect of X on Y, or the difference between λ^Y_{01} (the share of units with positive effects) and λ^Y_{10} (share with negative effects). When asking about the PC, we are asking, for a case with given values of X and Y, about the probability that X caused Y in that case. And a PC query is defined by a different set of parameters. For, say, an $X = 1, Y = 1$ case and a $X \rightarrow Y$ model, the probability of causation is given by just λ^Y_{01}.

Let us assume a "true" set of parameters, unknown to the researcher, such that $\lambda^Y_{01} = 0.6$, $\lambda^Y_{10} = 0.1$ while we set $\lambda^Y_{00} = 0.2$ and $\lambda^Y_{11} = 0.1$. Thus, the true average causal effect is 0.5. We now use the parameters and the model to simulate a large amount of data ($N = 10,000$). We then return to the model, set flat priors over nodal types, and update the model using the simulated data. We graph the posterior on our two queries, the ATE and the probability of positive causation in an $X = 1, Y = 1$ case, in Figure **??**.

The figure illustrates nicely the difference between an identified and non-identified query. While the ATE converges on the right answer, the probability of causation fails to converge even with a massive amount of data. We see instead a range of values for this query on which our updated model places roughly equal posterior probability.

Importantly, however, we see that we *do* learn about the probability of causation. Despite the lack of convergence, our posterior rules out a wide range of values. While our prior on the query was 0.5, we have correctly updated toward a range of values that includes (and happens to be fairly well centered over) the true value (≈ 0.86).

A distinctive feature of updating a causal model is that it allows us to learn about non-identified quantities in this manner. We will end up with "ridges" in our posterior distributions: ranges or combinations of parameter values that are equally likely given the data. But our posterior weight can nonetheless shift toward the right answer.

At the same time, for non-identified queries, we have to be cautious about the impact of our priors. As N becomes large, the remaining curvature we see in our posteriors may simply be function of those priors.

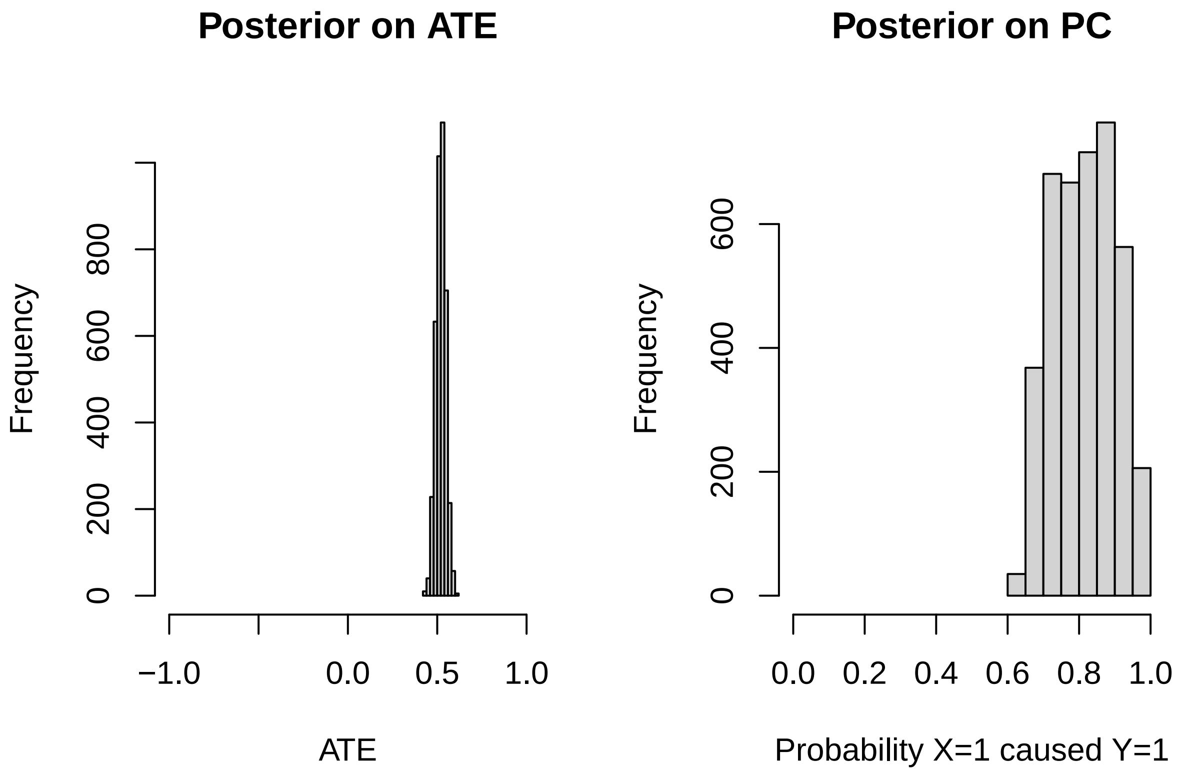

Figure 3: ATE is identified, PC is not identified but has informative bounds

Table 7: Data from non binary model (selection of rows)

X1	X2	Y1	Y2	X	Y
0	0	0	0	0	0
1	0	1	1	2	3
1	1	1	0	3	2
1	1	1	0	3	2

Table 8: Posteriors on potential outcomes for non binary model

Q	Using	True value	mean	sd
Y(0)	posteriors	0	0.37	0.08
Y(1)	posteriors	1	0.98	0.07
Y(2)	posteriors	3	2.60	0.09
Y(3)	posteriors	2	2.02	0.07

One way to inspect for this is to simulate a very large dataset and see whether variance shrinks. A second approach is to do sensitivity analyses by updating the model on the same data with different sets of priors to see how this affects the shape of the posterior.

1.4.3 Beyond binary data

While the setup used in this book involves only binary nodes, the approach readily generalizes to non-binary data. Moving beyond binary nodes allows for considerably greater flexibility in response functions. For instance, moving from binary to merely 3-level ordinal X and Y variables allows us to represent non-linear and even non-monotonic relationships. It also allows us pose more complex queries, such as, "What is the probability that Y is linear in X?", "What is the probability that Y is concave in X?", or "What is the probability that Y is monotonic in X?"

To move to non-binary measurement, we need to be able to expand the nodal-type space to accommodate the richer range of possible relations between nodes that can take on more than two possible values. Suppose, for instance, that we want to operate with variables with 4 ordinal categories. In an $X \rightarrow Y$ model, Y's nodal types have to accommodate 4 possible values that X can take on, and 4 possible values that Y can take on for any value of X. This yields $4^4 = 256$ nodal types for Y and 1024 causal types (compared to just 8 in a binary setup).

The `CausalQueries` package, set up to work most naturally with binary nodes, can be used to represent non-binary data as well. The trick, as it were, is to express integers in base-2 and then represent the integer as a series of 0's and 1's on multiple nodes. In base-2 counting we would represent four integer values for X (say, 0, 1, 2,3) using $00, 01, 10, 11$. If we use one binary node, X_1 to represent the first digit, and a second node X_2 to represent the second, we have enough information to capture the four values of X. The mapping then is: $X_1 = 0, X_2 = 0$ represents $X = 0$; $X_1 = 0, X_2 = 1$ represents $X = 1$; $X_1 = 1, X_2 = 0$ represents $X = 2$; and $X_1 = 1, X_2 = 1$ represents $X = 3$. We construct Y in the same way. We can then represent a simple $X \rightarrow Y$ relation as a model with two X nodes each pointing into two Y nodes: $Y_1 \leftarrow X_1 \rightarrow Y_2, Y_1 \leftarrow X_2 \rightarrow Y_2$. To allow for the full range of nodal types we need to allow a joint distribution over θ^{X_1} and θ^{X_2} and over θ^{Y_1} and θ^{Y_2}, which results in 3 degrees of freedom for X and 255 for Y, as required.

In the illustration below with two 4-level variables, we generate data ($N = 100$) from a non-monotonic process with the following potential outcomes: $Y(0) = 0, Y(1) = 1, Y(2) = 3, Y(3) = 2$. We then update and report on posteriors on potential outcomes.

Data from this model looks like this:

Updating and querying is done in the usual way:

We see that the model performs well. As in the binary setup, the posterior reflects both the data and the priors. And, as usual, we have access to a full posterior distribution over all nodal types and can thus ask arbitrary queries of the updated model.

The greatest challenge posed by the move to non-binary data is computational. If Y takes on m possible values and has k parents, each taking on r possible values, we then have m^{r^k} nodal types for Y. Thus, the cost of more granular measurement is complexity – an explosion of the parameter space – as the nodal type space expands rapidly with the granularity of measurement and the number of explanatory variables With three 3-level ordinal variables pointing into the same outcome, for instance, we have $3^{27} = 7.6$ *trillion* nodal types!

We expect that, as measurement becomes more granular, researchers will want to manage the complexity by placing structure onto the possible patterns of causal effects. Structure, imposed through model restrictions, can quite rapidly tame the complexity. For some substantive problems, one form of structure we might be willing to impose is monotonicity. In a $X \rightarrow Y$ model with 3-level variables, excluding non-monotonic effects brings down the number of nodal types from 27 to 17. Alternatively, we may have a strong reason to rule out effects in one direction: disallowing negative effects, for instance, brings us down to 10 nodal types. If we are willing to assume linearity, the number of nodal types falls further to 5.

1.4.4 Measurement error

One potential application of the approach we have described in this chapter to integrating differing forms of data is to addressing the problem of measurement error. The conceptual move to address measurement error in a causal model setup is quite simple: we incorporate the error-generating process into our model.

Consider, for instance, a model in which we build in a process generating measurement error on the dependent variable.

$$X \rightarrow Y \rightarrow Y_{\text{measured}} \leftarrow \text{source of measurement error}$$

Here X has an effect on the true value of our outcome of interest, Y. The true value of Y, in turn, has an effect on the value of Y that we measure, but so too does a potential problem with our coding process. Thus, the measured value of Y is a function of both the true value and error.

To motivate the setup, imagine that we are interested in the effect of a rule restricting long-term care staff to working at a single site (X) on outbreaks of the novel coronavirus in long-term care facilities (Y), defined as infections among two or more staff or residents. We do not directly observe infections, however; rather, we observe positive results of PCR tests. We also know that testing is neither comprehensive nor uniform. For some units, regular random testing is carried out on staff and residents while in others only symptomatic individuals are tested. It is the latter arrangement that potentially introduces measurement error.

If we approach the problem naively, ignoring measurement error and treating Y_{measured} as though it were identical to Y, a differences in means approach might produce attenuation bias—insofar as we are averaging between the true relationship and 0.

We can do better with a causal model, however. Without any additional data, we can update on both λ_Y and $\lambda^{Y_{\text{measured}}}$, and our posterior uncertainty would reflect uncertainty in measurement. We could go further if, for instance, we could reasonably exclude negative effects of Y on Y_{measured}. Then, if we observe (say) a negative correlation between X and Y_{measured}, we can update on the substantive effect of interest – λ^Y – in the direction of a larger share of negative effects: it is only *via* negative effects of X on Y that a negative correlation between X and Y_{measured} could emerge. At the same time, we learn about the measure itself as we update on $\lambda^{Y_{\text{measured}}}$: the negative observed correlation X and Y_{measured} is an indicator of the degree to which Y_{measured} is picking up true Y.

We can do better still if we can collect more detailed information on at least some units. One data strategy would be to invest in observing Y, the true outbreak status of each unit, for a subset of units for which we