Appendix for Holistic Causal Learning with Causal Graphs

Robert Kubinec

October 30, 2024

Table of contents

Case Study: Oil Wealth and Democratization

One of the most actively-research questions in comparative political science concerns the theorized relationship between a country discovering oil resources and subsequent resistance to democratization. Known as the rentier state theory (Mahdavy, Hossein 1970; Beblawi and Luciani 1987), this influential hypothesis has received both significant support (Ross 2001, 2012) and resistance from scholars (Haber and Menaldo 2011; Cherif 2015). Of particular relevance for the application of causal entropy is the recent work of Waldner and Smith (2021), who review the literature and produce a compelling causal graph that is able to summarize much of the debate while also making their new claims about the underlying causal structure.

I reproduce their article's Figure 5 in Figure 1 where variable A stands for "autocratic resilience", B for "British policy", P for "protection", O for "oil" and S for "survival." Waldner and Smith's argument is that there is no actual causal arrow between oil and autocratic resilience. Rather, there appears to be such a relationship because of what is known as collider bias (or selection bias) caused by variable P. Waldner and Smith make the case that British policy caused autocratic resilience among the heavily authoritarian Gulf states, and oil is associated with authoritarian institutions because oil endowments helped these states secure protection from foreign rivals such as Saudi Arabia and Iran. As a result, these conservative monarchies were more likely to survive and more likely to be authoritarian, inducing a spurious association between oil resources and authoritarian institutions.

```
resource_dag <- "dag {
                O -> P
                P -> S
                B -> P
                B -> A
```

```
                }"


rd <- dagitty(resource_dag)
ggdag(rd) + theme_tufte() +
  labs(x="",y="") +
  theme(text=element_text(family=""),
        axis.text = element_blank())
```

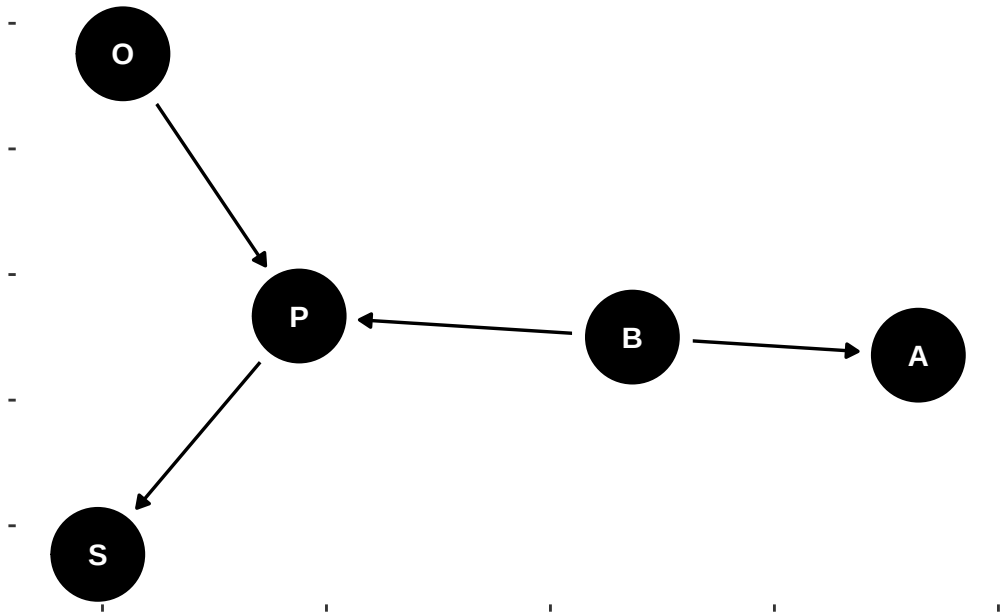


Figure 1: Replication of Figure 5 from Waldner and Smith (2020)

The aim of this case study is not to adjudicate this original claim but rather to use their causal graph as a basis for evaluating possible research studies using causal entropy. This causal graph is far more involved than the previous one, and as a result, the conditional probability tables are more complex. To simplify matters, we will focus on the probability of authoritarian resilience (A) conditional on British policy (B) and survival $S \in \{\text{High}, \text{Low}\}$:

Table 1: $Pr(A|B, S = \text{Low})$

	$A =$ Endure	$A =$ Democratize
$B =$ Intervention	0.5	0.5
$B =$ No Intervention	0.5	0.5

Table 2: $Pr(A|B, S = \text{High})$

	$A =$ Endure	$A =$ Democratize
$B =$ Intervention	0.5	0.5
$B =$ No Intervention	0.5	0.5

As can be seen in both tables, given that this is a new argument, we will use as our prior distribution complete uncertainty. Otherwise, we would be biasing our results in favor of Waldner and Smith's contention. Furthermore, there is an additional problem with arriving at a more informative prior graph: we cannot manipulate these variables, and we also cannot observe variation naturally. The Gulf states have always been sovereign since their independence, and British policy in question occurred between the 1920s and 1950s as the monarchies consolidated their rule. Because we cannot go back in time to collect more data, we are much more limited in what tools we can use to arrive at more informative distributions for our causal graph.

```
prob_out_post <- crossing(A=c("Endure","Democratize"),
                          B=c("Intervention","No Intervention"),
                          S=c("Low","High"),
                          Omega=c("Interventionist","Non-Interventionist")) %>%
  mutate(pr_joint=case_when(Omega=="Interventionist" ~ .5^3 * .9,
```

```
                        TRUE ~ .5^3 * .1))

prob_out_null <- crossing(A=c("Endure","Democratize"),
                          B=c("Intervention","No Intervention"),
                          S=c("Low","High"),
                          Omega=c("Interventionist","Non-Interventionist")) %>%
  mutate(pr_joint=.5^4)

ent_orig <- -sum(prob_out_null$pr_joint*log(prob_out_null$pr_joint,1.01))
ent_post <- -sum(prob_out_post$pr_joint*log(prob_out_post$pr_joint,1.01))
```

As a result, the mechanisms are very important. Waldner and Smith present evidence on a crucial mechanism–the nature of British intervention in Gulf politics. In other words, what were the intentions of British policymakers towards these states? Did they have an explicit policy of promoting authoritarianism? We can label this mechanism $\Omega = \{\text{Interventionist}, \text{Non-Interventionist}\}$, and then consider the joint distribution $Pr(A, S|\Omega)Pr(\Omega|B)Pr(B)$. While evaluating the quality of Waldner and Smith's historical work is beyond the scope of this short study, we will give them the benefit of the doubt and say that they can establish the $Pr(\Omega|B = \text{Interventionst} = 0.9)$. We can then calculate the reduction of log entropy for $Pr(A, S|\Omega)Pr(\Omega|B)Pr(B)$ vis-a-vis a null mechanism of $Pr(\Omega = \text{Interventionst}|B) = 0.5)$, which leads to a reduction in entropy from 279 to 242. As a result, their process-tracing reduces our uncertainty considerably, although learning about only one mechanism still leaves substantial uncertainty.

Another way we can learn about this causal graph is to focus on the hypothesized null relationships. The graph in Figure 1 has no direct connection between oil O and authoritarian resilience A. There have been experimental and quasi-experimental studies that have studied

a hypothesized mediator known as demands for accountability in which higher reliance of the state on rents will cause reduced demands for accountability. We can then update our causal graph with this hypothesized relation in Figure 2 by denoting demand for accountability as D.

```
resource_dag2 <- "dag {
                    O -> P
                    P -> S
                    B -> P
                    B -> A
                    O -> D
                    D -> A}"

rd2 <- dagitty(resource_dag2)
ggdag(rd2) + theme_tufte() +
  labs(x="",y="") +
  theme(text=element_text(family=""),
        axis.text = element_blank())
```

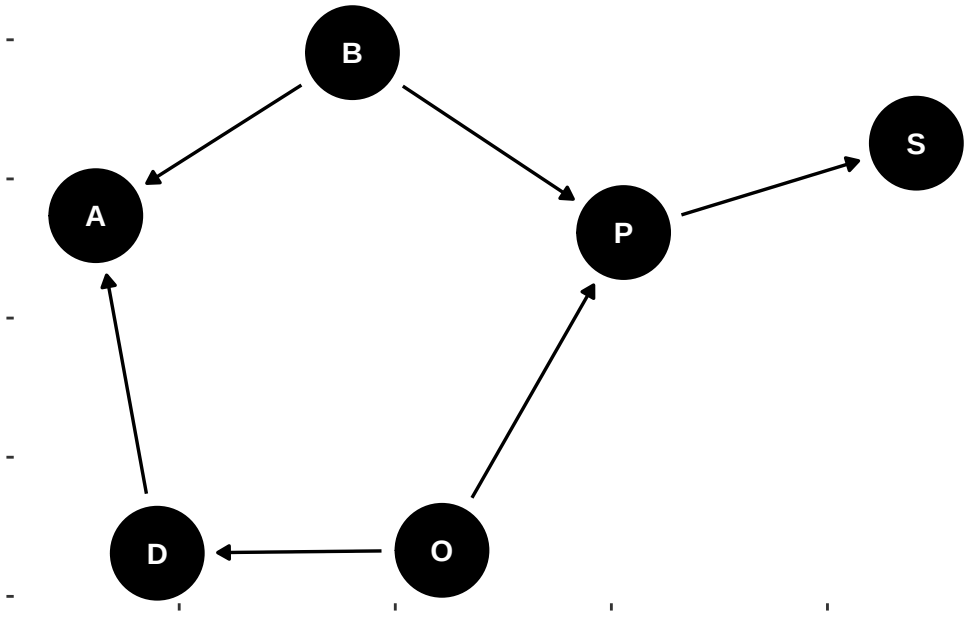


Figure 2: Causal Diagram with Hypothesized Mediator D

We now have an open causal pathway between oil resources O and authoritarian resilience A. However, because we still have the collider relationship between protection P and survival S, we should be worried about studies that do not take into account this possibility for selection bias. Experimental inference would seem to be a good option here, although we are limited in what we can focus on in the graph. We could either try to manipulate oil resources (or find plausibly exogenous increases in natural resources) and see whether demands for accountability change (path $O \rightarrow A$), or we could look at whether increases in demands for accountability lead to lower authoritarian resilience (path $D \rightarrow A$).

Neither is particularly easy to achieve, though recent work has shown it can be done. Paler (2013) employed an experimental treatment to prime Indonesians into thinking either about taxes or "windfall" natural resources as a proportion of their local government's budget. She found a 5-6 pp increase in accountability actions by survey respondents in the taxation versus windfall condition. On the other hand, Cuesta et al. (2019) deployed a lab-in-the-field experiment with a similar informational treatment in Ghana and Uganda but found

little difference between taxation and other sources of revenue when it came to demands for accountability.

Estimating the $D \rightarrow A$ path may be even harder as it requires measures of authoritarian persistence. Kubinec and Milner (2022) deployed an interactive informational treatment during Algeria's national protest movement in 2019 and found that informing Algerians about the extent of government redistribution had strongly varying effects by respondent wealth. More wealthy respondents were less willing to join protests opposing the regime, while poorer respondents became more likely to do so following the treatment. In sum, the experimental evidence appears to be mixed on this question, possibly justifying Waldner and Smith's lack of a causal arrow from $O \rightarrow A$.

Finally, we could also consider studies that are observational in nature, such as Ross (2001) and Haber and Menaldo (2011). From the causal graph, it should be clear that these studies will be unable to be causally-identified because it is not possible to adjust for selection bias with "control" variables in regression (Hünermund and Louw 2020), and a selection model would require at least some observations of countries that did not survive (Waldner and Smith 2021, 898). As such, we would have to interpret observational studies that measure the association between natural resources and authoritarianism as assessing the joint distribution $Pr(A, P, S, B|O)Pr(O)$–in other words, we can only factor out oil resources. One advantage is that oil resources have considerable variation, but a disadvantage is that oil resources change little over time. As such, our learning from these studies predominantly depends on how much oil varies vis-a-vis the joint distribution, and in any case, we will not be able to isolate the effect of oil on authoritarianism apart from the proposed selection process.

As such, in this case study it would appear that experimental approaches and qualitative or mechanistic studies would be most aptly suited to research progress–unless observational work was able to better model the possible selection processes. Simply observing oil resources is unlikely to offer much type I causal learning without an ability to adjust for other important

variables.

Tutorial for Estimating Entropy of Causal Graphs with R

In this section I show R code that can produce causal graphs, estimate conditional probability tables, and calculate entropy quickly and easily. To do so, I use four R packages: dplyr for data management, HydeNet for creating conditional probability tables, dagitty for defining causal graphs and ggdag for creating visualizations of causal graphs. Each of these packages has many more options and possibilities than I show in this brief tutorial.

```
library(dplyr)
library(HydeNet)
library(dagitty)
library(ggdag)
```

I further define a convenience function for calculating entropy given a list of vectors or tables of conditional probabilities. The function's arguments include a list of each conditional probability table, the list of values for each variable and the ability to set the base of the logarithm used, which defaults to 1.01.

```
entropy <- function(x,base=1.01) {

  if(class(x)!="list") {
    stop("Please pass the vectors or tables in a list to the function.")
  }

  if(length(x)==1) return(-sum(c(x[[1]])*log(c(x[[1]]),base)))
```

```
# figure out dimensions of each table

dims <- sapply(x, function(y) {

  if('cpt' %in% class(y)) {

    length(dim(y))

  } else {

    1

  }

})



all_data_frames <- lapply(1:length(x), function(i) {
      if("cpt" %in% class(x[i][[1]])) {
        y <- attr(x[i][[1]], "model")
        y[length(y)] <- NULL
        y <- rename(y, !!paste0("wt",i) := wt)

      } else {
        y <- x[i][[1]]
```

```
      }
    y
  })

# need to make a unique combination for smallest element
biggest <- all_data_frames[dims==max(dims)][[1]]



# need to create all combinations

lapply(1:length(all_data_frames), function(i) {

    biggest <<- left_join(biggest, all_data_frames[i][[1]])

  return(NULL)
  })

biggest <- biggest %>%
            rowwise %>%
             mutate(joint_pr=prod(c_across(matches('wt'))))

-sum(biggest$joint_pr*log(biggest$joint_pr,base))

}
```

We will re-create our simple confounding graph from Figure 1 in the main text with the

dagitty package. This involves specifying each path in the DAG separately in a character/text variable with one path per line, and then passing it to the dagitty function:

```
dag <- dagitty("dag { T -> Y
                Z -> T
                Z -> Y
               }")
```

We can then use the ggdag package to visualize the causal graph:

```
ggdag(dag)
```

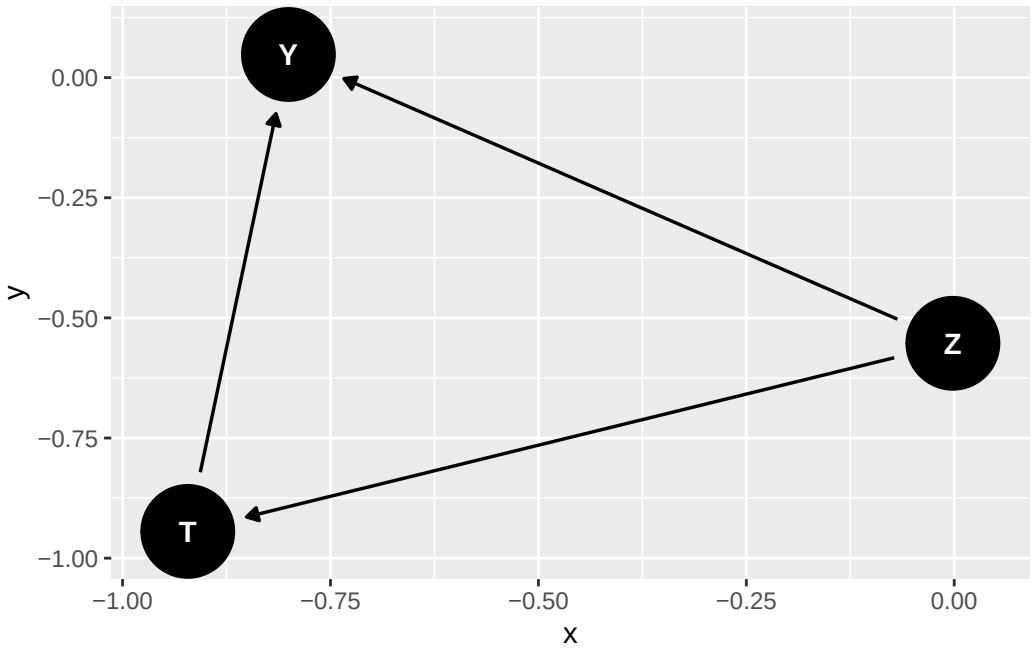


Figure 3: Example Visualized Directed Acyclic Graph

To be able to calculate entropy, we need to create conditional probability tables that together make a joint probability distribution for the outcome, Y. The Markov property of causal graphs means we only need to consider each variable's immediate ancestors when doing so.

If our outcome is Y, then we need to estimate the conditional distribution of $Pr(Y|T, Z)$ as these are Y's two ancestors. We then need to make conditional probability tables for any of Y's ancestors that have "open paths" to Y, which in this case is only T due to the influence of Z on Y via T, so we also need $Pr(T|Z)$. Any variable that does not have arrows going into it, and is connected to Y, needs to be included as an unconditional probability, or $Pr(Z)$. As such, we multiply three terms to obtain the joint probability distribution of the causal process that produces Y: $Pr(Y|T, Z)Pr(T|Z)Pr(Z)$.

We can check and see which variables we need to include by identifying the ancestor variables of Y with the parents function from dagitty:

```
parents(dag, "Y")
```

[1] "T" "Z"

To create plausible conditional probability tables, we can use the inputCPT function from the HydeNet package to interactively add in probability values for each conditional probability relationship. The function takes in the names of variables using R's formula notation with the predicted variable first and any conditioning variables after the ~ sign. The user must specify probabilities for all except one level of the conditional probabilities (assuming that the last level is equal to remainder). In this case, we need two conditional probability tables:

1. $Pr(Y|T, Z)$
2. $Pr(T|Z)$

```
joint_table1 <- readRDS("joint_table1.rds")
joint_table2 <- readRDS("joint_table2.rds")
```

I create them using the command interactively by entering in default values ("Yes" or "No") and probability values for each possible value:

```
# commented out as it is interactive
#joint_table1 <- inputCPT(Y ~ T + Z)
#joint_table2 <- inputCPT(T ~ Z)
print(joint_table1)
```

```
, , Y = No

     Z
T      No Yes
  No  0.9 0.8
  Yes 0.5 0.2

, , Y = Yes

     Z
T      No Yes
  No  0.1 0.2
  Yes 0.5 0.8
```

```
print(joint_table2)
```

```
     T
Z      No Yes
  No  0.4 0.6
  Yes 0.8 0.2
```

For $Pr(Z)$, we will have to create a data frame manually as this probability is not conditional

on anything. We need to include the values of Z and the probability of each outcome as the column wt:

```
joint_table3 <- tibble(Z=c("Yes","No"),
                       wt=c(0.9,0.1))
```

Given these conditional probability tables and one unconditional probability distribution, we can then quickly estimate the entropy of the causal graph by passing all of the tables and data frames as a list:

```
entropy(list(joint_table1,joint_table2,joint_table3))
```

```
[1] 135.4428
```

To calculate a different causal graph given a potential study, simply create a new conditional probability table with new probability values with the inputCPT function.

Using CausalQueries to Calculate Entropy with Uncertainty

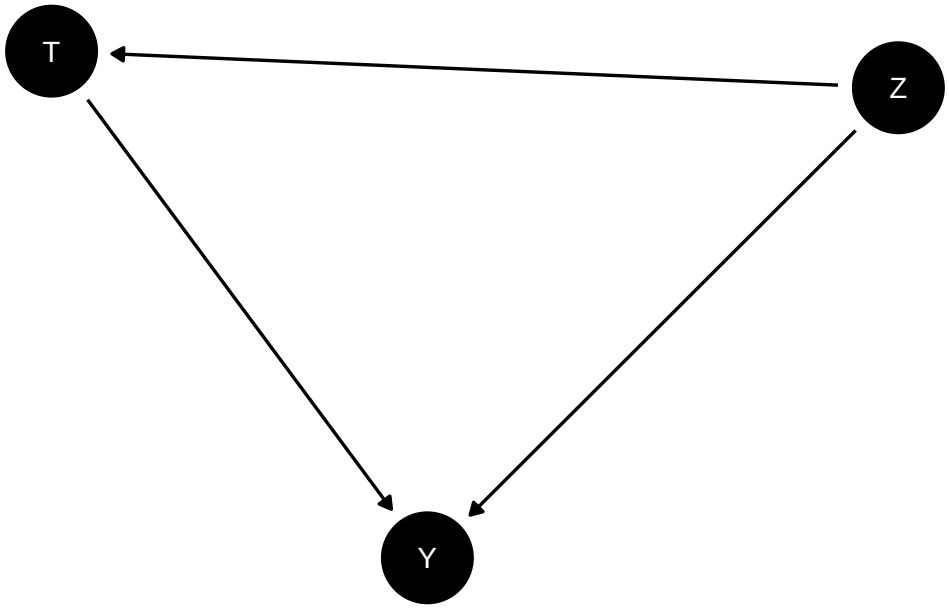
The R package CausalQueries, which is the package the implements the research design methods proposed in Humphreys and Jacobs (2023), permits novel types of inferences over causal graphs as defined in the previous section. CausalQueries allows the user to specify both point estimates for the probability of edge relations in a causal graph, or what Humphreys and Jacobs (2023) call "parameters", along with uncertainty in those parameters through the use of Dirichlet priors over the values of different node-to-node relationships. With this uncertainty, it is possible to talk about a distribution of entropies from a given research design/causal graph rather than just a point estimate. Helpfully, CausalQueries is also compatible with the dagitty package for specifying DAGs, and so we can employ the same example above.

We will first use the function make_model() to create a CausalQueries object given our DAG specification:

```
library(CausalQueries)


dag_cq <- make_model("{ T -> Y
                        Z -> T
                        Z -> Y }")


plot(dag_cq)
```



By default, CausalQueries creates null graphs or maximum entropy graphs with uniform priors over edge relations. The function below contributed by Macartan Humphreys shows how to calculate entropy from a CausalQueries object given its value of parameters and priors:

```
get_entropy <- function(model, parameters = NULL,base=1.01) {
  p <- get_event_probabilities(model, parameters)
  -sum(p*log(p,base=base))
}


get_entropy(dag_cq)
```

[1] 208.9822

This value matches the value of the maximum entropy or null graph described in the paper in the case study of the COVID-19 vaccine.

To discuss the entropy of plausible interventions, we need to change the parameters or priors of this object to reflect our beliefs about the relationships between nodes in the causal graph. Doing so is a more involved process than specifying the conditional probability tables as in the previous section as CausalQueries can be used for a range of tasks beyond entropy calculation, and as such requires more information about the graph. However, specifying such a graph permits the usage of a range of powerful CausalQueries features, including the role of plausible omitted confounders and, as we show, obtaining uncertainty in the estimates of entropy.

CausalQueries uses a syntax in which each node can take on binary values (0 or 1) and then allows the user to specify the probability of inter-node relationships. This can result in quite a large number of probabilities. For the three-node causal graph above, we end up with needing to specify 16 different probabilities for the terminal node Y because Y can take on values for each value of the preceding nodes Z and T as in the code below. The set_parameters function allows the user to specify these probabilities for each node outcome, which follows a syntax described in their package working paper. In brief, a nodal value of

0000 for Y corresponds to the probability that is $Y = 0$ for all possible combinations of Z and T. Since Z and T each have 2 values, this corresponds to four separate probabilities that must be multiplied together-0.002499-as in the code below.

```
cq_true <- set_parameters(dag_cq,
                          parameters=c(.25,.75),
                          node="Z",nodal_type=c("0","1")) %>%
        set_parameters(parameters=c(.9*.1,.1*.1,.9*.9,.1*.9),
                       node="T",
                       nodal_type=c("00","10","01","11")) %>%
 set_parameters(parameters=c(.02*.15*.98*.85, # 0000
                             .98*.15*.98*.85, # 1000
                             .02*.85*.98*.85, # 0100
                             .98*.85*.98*.85, # 1100
                             .02*.15*.02*.85, # 0010
                             .98*.15*.02*.85, # 1010
                             .02*.85*.02*.85, # 0110
                             .98*.85*.02*.85, # 1110
                             .02*.15*.98*.15, # 0001
                             .98*.15*.98*.15, # 1001
                             .02*.85*.98*.15, # 0101
                             .98*.85*.98*.15, # 1101
                             .02*.15*.02*.15, # 0011
                             .98*.15*.02*.15, # 1011
                             .02*.85*.02*.15, # 0111
                             .98*.85*.02*.15), # 1111
                node="Y",
```

```
                nodal_type=c("0000","1000",
                             "0100","1100",
                             "0010","1010",
                             "0110","1110",
                             "0001","1001",
                             "0101","1101",
                             "0011","1011",
                             "0111","1111"))
```

To find these probabilities, they can be identified from the conditional probability tables listed above. The probability that $Y = 0$ when $Z = 0$ and $T = 0$, for example, is equal to the probability that T=No Vaccine and I=Not Infected in Table 2 in the main text (i.e., when Z=Young or 0). This value of .02 is equal to the first of the four nodal values of Y for the first row 0000, i.e., the first element of that row is equal to the probability of Y for that particular combination of values for Z and T. To learn what those nodal types signify, you can access the formula from a CausalQueries object with this function:

```
grab(cq_true,"nodal_types")
```

```
Nodal types:
$Z
0  1


  node position display interpretation
1    Z       NA      Z0         Z = 0
2    Z       NA      Z1         Z = 1
```

$T

00 10 01 11

node position display interpretation

1 T 1 T[*]* T | Z = 0

2 T 2 T*[*] T | Z = 1

$Y

0000 1000 0100 1100 0010 1010 0110 1110 0001 1001 0101 1101 0011 1011 0111 1111

node position display interpretation

1 Y 1 Y[*]*** Y | Z = 0 & T = 0

2 Y 2 Y*[*]** Y | Z = 1 & T = 0

3 Y 3 Y**[*]* Y | Z = 0 & T = 1

4 Y 4 Y***[*] Y | Z = 1 & T = 1

Number of types by node

Z T Y

2 4 16

As can be seen, the first element in the nodal types of Y (i.e. Y[*]***) is equal to the probability of Y when $Z = 0$ and $T = 0$. So the user has to match of these four outcomes of Y given Z and T from the conditional probability tables to specific cells in the conditional probability tables. While this involves specifying a significant number of values for Y, there is also much repetition in the formula that simplifies the task. For example, the difference between the nodal types $Y = 0000$ and $Y = 1000$ is only one element, or the probability

of Y when $Z = 0$ and $T = 0$. The user only needs to change the code r.02.15.98*.85‘ to 0.122451, that is, change the probability of $Y = 0$ to $Y = 1$: 1 - 0.02 = 0.98.

However, the payoff to specifying the probabilities in this way is to access all of the functions of CausalQueries as previously mentioned. While we have changed the parameters of the CausalQueries object, we can also look at the priors:

```
grab(cq_true,"prior_hyperparameters")
```

Z.0	Z.1	T.00	T.10	T.01	T.11	Y.0000	Y.1000	Y.0100	Y.1100	Y.0010
1	1	1	1	1	1	1	1	1	1	1

Y.1010	Y.0110	Y.1110	Y.0001	Y.1001	Y.0101	Y.1101	Y.0011	Y.1011	Y.0111	Y.1111
1	1	1	1	1	1	1	1	1	1	1

The priors in a CausalQueries object are Dirichlet priors over nodal types in which each nodal type is given a partition of the total probability. The prior number, called alpha, for a nodal type can be any positive number, and if all the of the alphas are the same for a node, such as $Z.0 = 1$ and $Z.1 = 1$ as in the code above, then both outcomes have equal probability. This would correspond to a maximum entropy distribution for Z (uniform distribution). Note that these priors are separate from the parameters (point estimates) that we specified previously.

To add more information into these priors, we would want to increase the value for specific nodes. These can be given the interpretation of “prior counts of successes” or what can be more easily understood as “prior sample size”. Suppose we form our priors based on 20 independent observations for the probability of Z and based on these observations we believe both outcomes of Z to be equally probable. In that case, we would update the priors to be equal to 10 for both values of Z:

```
cq_true <- set_priors(cq_true,alphas=c(10,10),node="Z")
grab(cq_true,'prior_hyperparameters')
```

```
  Z.0   Z.1  T.00  T.10  T.01  T.11 Y.0000 Y.1000 Y.0100 Y.1100 Y.0010
   10    10     1     1     1     1      1      1      1      1      1
Y.1010 Y.0110 Y.1110 Y.0001 Y.1001 Y.0101 Y.1101 Y.0011 Y.1011 Y.0111 Y.1111
     1      1      1      1      1      1      1      1      1      1      1
```

We now believe that both outcomes of Z are equally plausible, but we are more certain that they are equally plausible than the default prior for which we had only two observations. If from our prior sample size we believe that 15 observations supported $Z = 1$ and 5 observations supported $Z = 0$, we could use that prior specification for Z:

```
cq_true <- set_priors(cq_true,alphas=c(5,15),node="Z")
grab(cq_true,'prior_hyperparameters')
```

```
  Z.0   Z.1  T.00  T.10  T.01  T.11 Y.0000 Y.1000 Y.0100 Y.1100 Y.0010
    5    15     1     1     1     1      1      1      1      1      1
Y.1010 Y.0110 Y.1110 Y.0001 Y.1001 Y.0101 Y.1101 Y.0011 Y.1011 Y.0111 Y.1111
     1      1      1      1      1      1      1      1      1      1      1
```

It can of course be hard to know exactly what to set these priors to. These priors could be set based on relative heuristics, i.e., use a prior sample size of 50 for Z, 30 for T and 100 for Y. An alternative strategy is to sample from the parameters we set earlier and then see what a certain amount of data would imply about the posterior distribution of these nodal types. It is easy to sample from a CausalQueries object:

```
sample_data1 <- make_data(cq_true,n=100)
table(select(sample_data1,Y,T,Z))
```

```
, , Z = 0

   T
Y    0  1
  0  1  6
  1 19  0

, , Z = 1

   T
Y    0  1
  0  2 59
  1  5  8
```

We can see that we have data generated for our two cases for we have a young ($Z = 0$) and old ($Z = 1$) population. For young people, the vaccine is very effective (0 people who get the vaccine also get COVID-19 (T=1)) while it is less effective among the elderly (10 people get COVID-19 who get the vaccine). With this data, we can see how our the priors update as well, which is called updating the model in CausalQueries:

```
cq_true_update1 <- update_model(cq_true,data=sample_data1,refresh=0,
                                keep_fit=TRUE,
                                keep_event_probabilities = TRUE)
head(cq_true_update1$posterior_distribution)
```

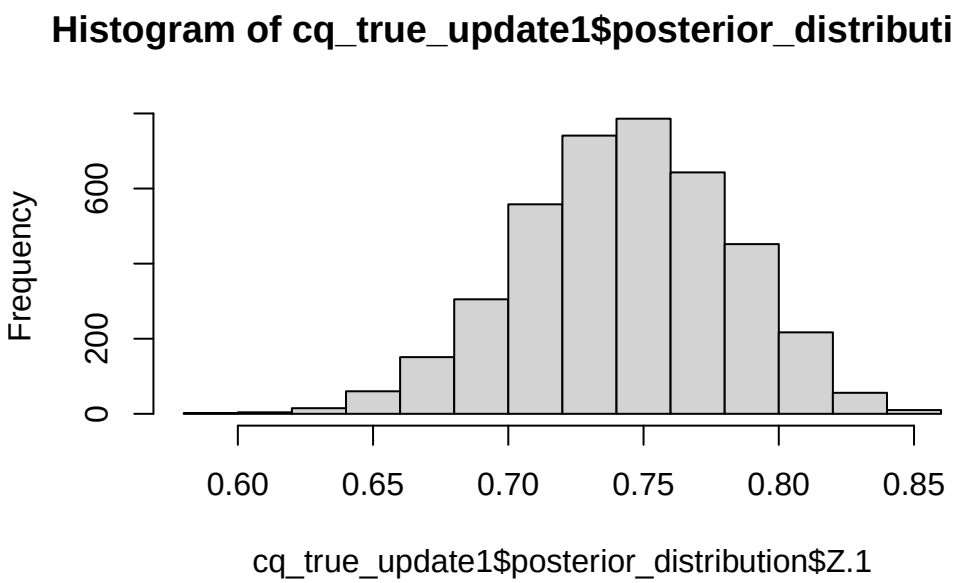
Summary statistics of model parameter posterior distributions:

: 6 rows (draws) by 22 cols (parameters)

	mean	sd
Z.0	0.24	0.04
Z.1	0.76	0.04
T.00	0.05	0.04
T.10	0.06	0.04
T.01	0.73	0.11
T.11	0.16	0.10
Y.0000	0.04	0.04
Y.1000	0.18	0.13
Y.0100	0.02	0.02
Y.1100	0.31	0.09
Y.0010	0.02	0.02
Y.1010	0.07	0.05
Y.0110	0.03	0.03
Y.1110	0.11	0.10
Y.0001	0.02	0.01
Y.1001	0.05	0.04
Y.0101	0.02	0.03
Y.1101	0.02	0.01
Y.0011	0.02	0.02
Y.1011	0.02	0.02
Y.0111	0.02	0.02
Y.1111	0.04	0.03

These are the posterior estimates for our nodal values, with each a plausible draw from the Dirichlet distribution over nodes. We can visualize the uncertainty via histograms:

```
hist(cq_true_update1$posterior_distribution$Z.1)
```

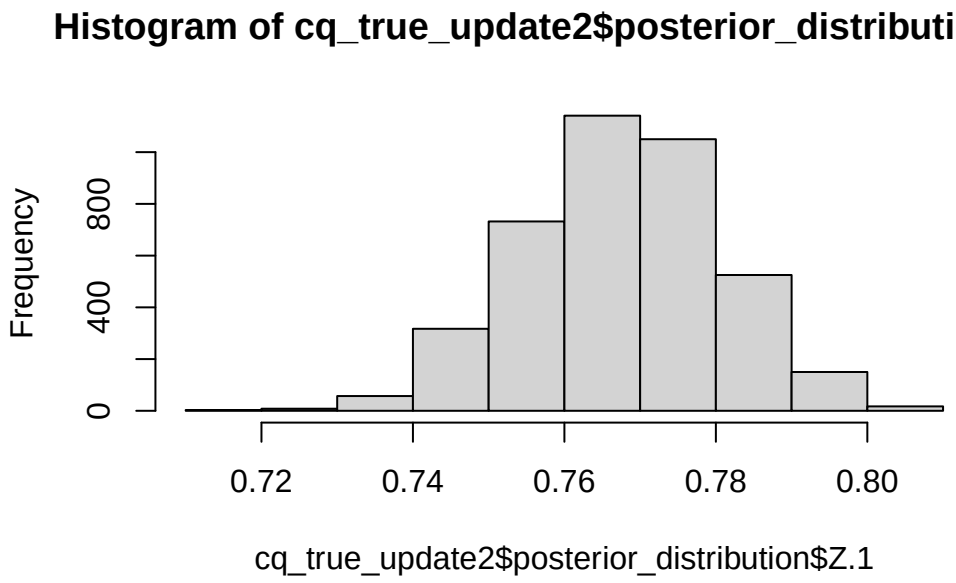


We see that we expect there to be about 70 - 75% old people in the distribution with a confidence interval roughly 62 to 82 percent.

Utilizing this feature of CausalQueries is quite useful as it allows us to understand that, given a set research design and parameter values that we can stand to learn, what is our residual uncertainty about the causal relationships for varying amounts of data? For example, we can repeat the above analysis but generate 1000 independent samples from the true causal graph:

```
sample_data2 <- make_data(cq_true,n=1000)
cq_true_update2 <- update_model(cq_true,data=sample_data2,refresh=0,
                                keep_fit=TRUE,
```

```
                                keep_event_probabilities = TRUE)
hist(cq_true_update2$posterior_distribution$Z.1)
```



We can see now that the average value is the same but our uncertainty has shrunk.

Given these posterior draws, we can also calculate differences in our uncertainty in entropy. Below I update both distributions with our two samples (100 and 1,000 observations) and then plot the resulting distributions in entropy:

```
# need a new function for calculating the entropy of a distribution of probabilities

ent_func <- function(p,base) -sum(p*log(p,base=base))

entropy_dist <- function(mod, base=1.01) {

  apply(mod$stan_objects$event_probabilities, 1, ent_func, base=base)
```

```
}
```

```
print("Entropy of 100 observations:")
```

```
[1] "Entropy of 100 observations:"
```

```
summary(entropy_dist(cq_true_update1))
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  112.3   141.3   147.9   147.7   154.3   181.6 
```

```
print("Entropy of 1,000 observations:")
```

```
[1] "Entropy of 1,000 observations:"
```

```
summary(entropy_dist(cq_true_update2))
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  111.2   120.7   122.8   122.8   125.0   134.7 
```

The average entropies are not identical due to sampling variation. However, it is still easy to see that the 100 observations model has a much wider spread in entropy than the 1,000 observations model. This meta-uncertainty can be helpful in making inferences over plausible research designs more robust.

References

Beblawi, Hazem, and Giacomo Luciani, eds. 1987. The Rentier State. Croom Helm Ltd.

Cherif, Youssef. 2015. "The Leadership Crisis of Nidaa Tounes." http://carnegieendowment.org/sada/62216.

Cuesta, Brandon de la, Helen V. Milner, Daniel L. Nielson, and Stephen F. Knack. 2019. "@." Proceedings of the National Academy of Sciences 116 (36): 17717–22.

Haber, Stephen, and Victor Menaldo. 2011. "Do Natural Resources Fuel Authoritarianis: A Reappraisal of the Resource Curse." American Political Science Review 105 (1): 1–26.

Humphreys, Macartan, and Alan M. Jacobs. 2023. Integrating Inferences: Causal Models for Qualitative and Mixed-Method Research. Cambridge University Press. https://books.google.com/books?hl=en&lr=&id=0E3XEAAAQBAJ&oi=fnd&pg=PR9&dq=info:2K-gFfO67ZYJ:scholar.google.com&ots=cmUb1DyO6s&sig=OG0g6Cr3tsPGVMf01QNhKFmkgUE.

Hünermund, Paul, and Beyers Louw. 2020. "On the Nuisance of Control Variables in Regression Analysis." https://doi.org/10.48550/arXiv.2005.10314.

Kubinec, Robert, and Helen Milner. 2022. "Taxes in the Time of Revolution: An Experimental Test of the Rentier State During Algeria's Hirak." https://doi.org/10.31235/osf.io/hu3vq.

Mahdavy, Hossein. 1970. "Patterns and Problems of Economic Development in Rentier States : The Case of Iran." In. Routledge.

Paler, Laura. 2013. "Keeping the Public Purse: An Experiment in Windfalls, Taxes, and the Incentives to Restrain Government." American Political Science Review 107 (4): 706–25. https://doi.org/10.1017/S0003055413000415.

Ross, Michael. 2001. "Does Oil Hinder Democracy?" World Politics 53 (3): 325–61.

———. 2012. The Oil Curse: How Petroleum Wealth Shapes the Development of Nations. Princeton: Princeton University Press.

Waldner, David, and Benjamin Smith. 2021. "Survivorship Bias in Comparative Politics: Endogenous Sovereignty and the Resource Curse." Perspectives on Politics 19 (3): 890–905. https://doi.org/10.1017/S1537592720003497.