Thinking with causal models

The process of constructing a causal model provokes a depth of thinking about the system you are trying to model.

- What do we include / exclude in the abstraction?
- What *level* of abstraction is relevant? (Where do we make the epistemic cut)

At each stage we are prompted to think about:

- Can I justify the nodes / variables in the abstraction, and those excluded?
- Can I justify the paths in the model? (Here I think it is important to think in terms of paths, not in terms of edges - it is the nodes along with the collection of *paths* between each two pair of d-connected variables that are important, not just the edges connecting adjacent variables)

With a graphical tool for thinking with (the DAG) we have the capacity to communicate and collaborate (think together) about the system. This means that a collaborator can be brought in (to the fold?) to help interrogate the model and clarify, possibly using the prompts above.

Starting simple

The approach here will follow a

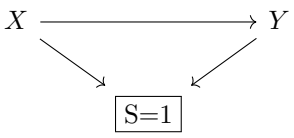


Figure 1: Initial structural causal model relating intervention X to outcome Y with selection process S. $S = 1$ to represent that selection into the intervention is in effect already conditioned upon.

Variables:

- X is the intervention, having a dialogue with the student.
- Y is the outcome, per student per session. Y^1 denotes the student *did not* recieve any NPE grades that session.
- A is the (unmeasured) academic capacity and decision making of the student. This influences both their success in the session (Y) and their likelihood of answering the phone call (S)
- S is the selection process; if the student answered the phone call or not.
- C are the selection of covariates that influence both whether the student answered the call, and their final outcome. How these covariates effect the final outcome is mediated through the intervention X and then their academic capacity A

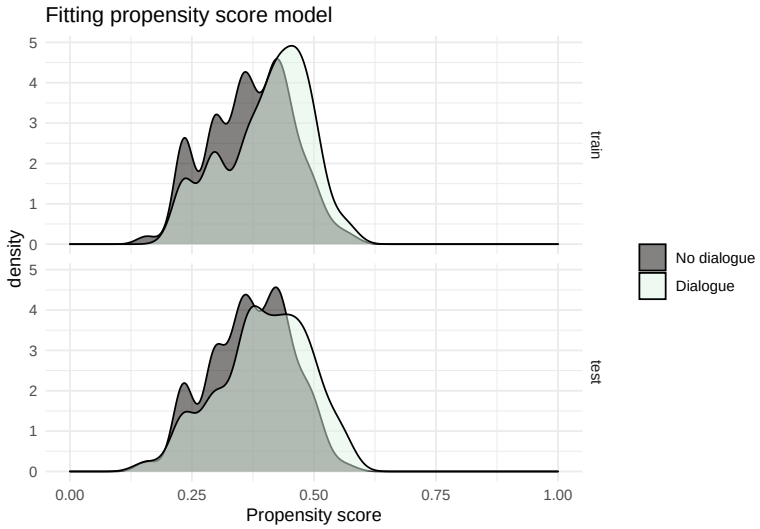
Arrows not included:

- $S \to A$, as this effect of involvement in the program is expressed through $X \to A \to Y$

Adding data

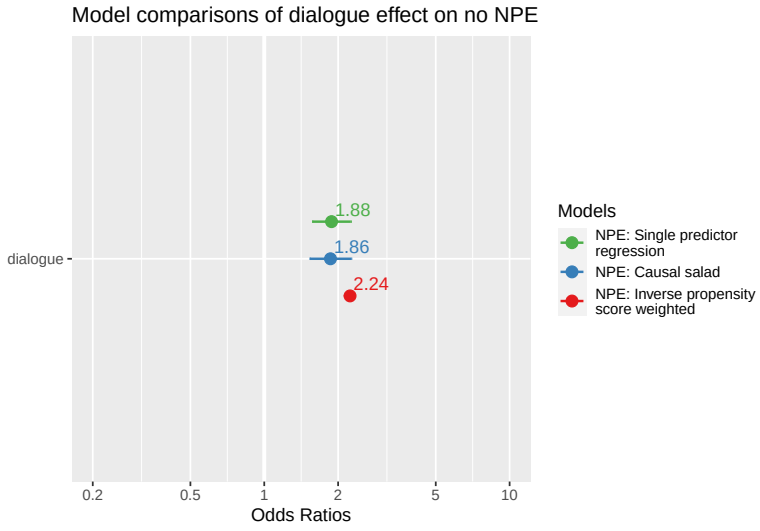
Propensity model

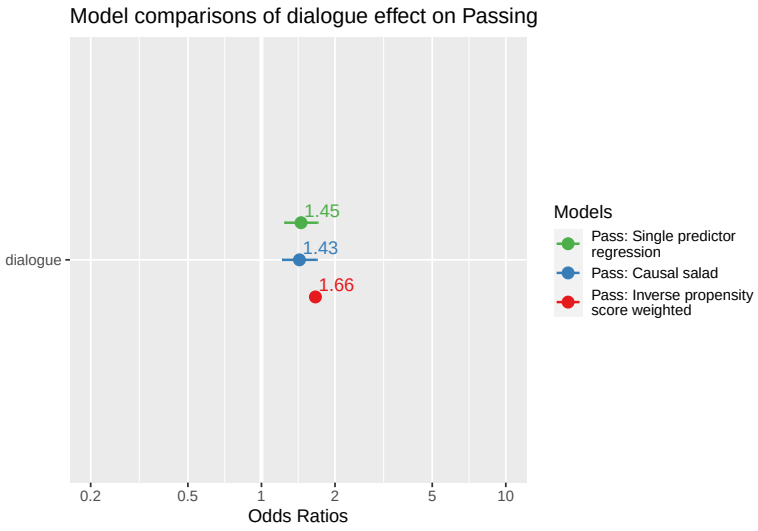
The aim here is to estimate the propensity score $p(S = 1|C)$ to glean information about the selection process, and then use this to adjust the model accordingly. There are three ways to use this information; matching, inverse weights, and sub classes. Weighting has issues (King and Nielsen, 2019), so we explore the other two methods here.

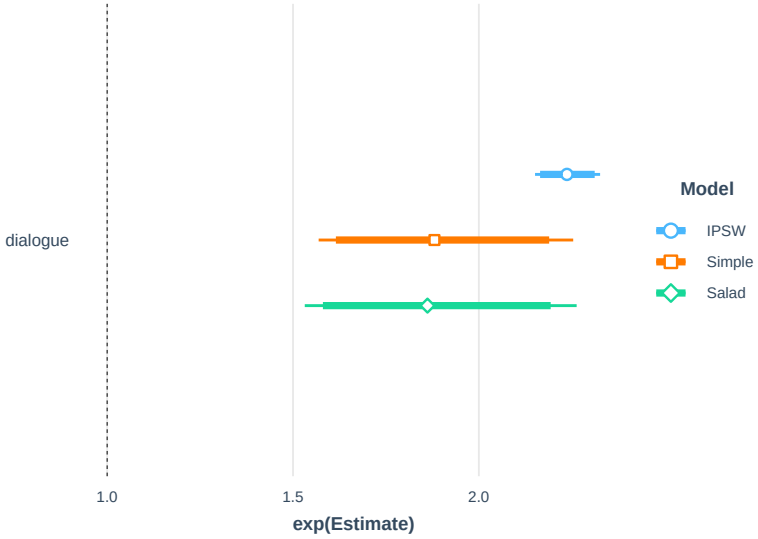


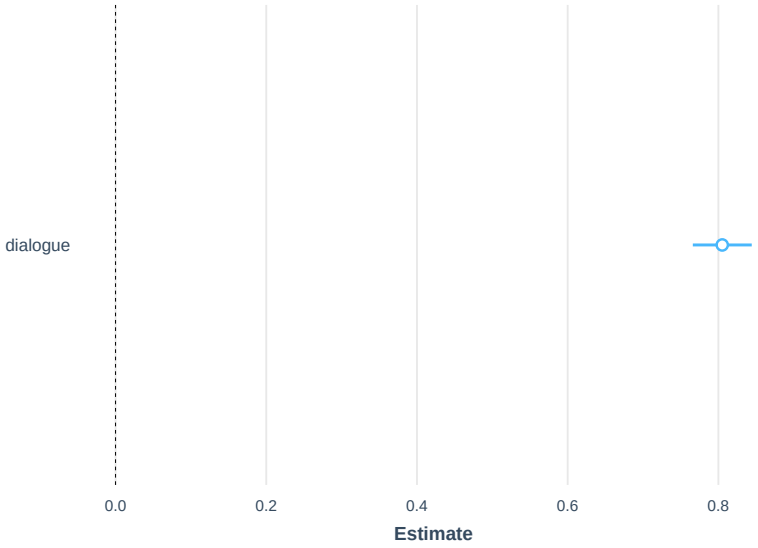
Inverse weighting

Once we have the propensity scores we can then fit the model using the inverse propensity score as weights.




Model comparisons of dialogue effect on Passing
dialogue
1.45
1.43
1.66
0.2
0.5
1
2
5
10
Odds Ratios
Models
Pass: Single predictor regression
Pass: Causal salad
Pass: Inverse propensity score weighted


dialogue
1.0
1.5
2.0
exp(Estimate)
Model
IPSW
Simple
Salad


dialogue
0.0
0.2
0.4
0.6
0.8
Estimate

Appendix

Context of the data

Data collected over...

All code for this report

R

```
library(tidyverse)
library(lubridate)
library(knitr)
library(readxl)
library(janitor)
library(retention.data)
library(retention.helpers)
library(sjPlot)
library(dagitty)
library(ggdag)
dag_coords <- list(
  x = c(X = 0, A = 1, Y = 2, C = 0, S = 1),
  y = c(X = 1, A = 1, Y = 1, C = 0, S = 0)
)

dag <- dagify(
  X ~ C,
  A ~ X,
  Y ~ A,
  S ~ A + C,
  coords = dag_coords,
  exposure = "X", outcome = "Y", latent = "A"
)
SESSIONS <- c(201930, 201960, 201990,
              202030, 202060, 202090)

aca_agg <- academic %>%
  filter(!str_detect(subject, "SSS"), session %in% SESSIONS) %>%
  add_grade_helpers() %>%
  group_by(id, session) %>%
  summarise(
    any_complete = any(grade_finalised),
    no_npe = !any(grade_npe),
    all_pass = all(grade_pass | !grade_finalised),
    .groups = "drop")

fla_agg <- flags |>
  filter(campaign == "pre census") %>%
  filter(!str_detect(reason_for_no_upload, "[Mm]istake")
         | is.na(reason_for_no_upload)) %>%
  filter(session %in% SESSIONS) %>%
```

```
  group_by(id, session) %>%
  summarise(
    n_flags = n(),
    any_non_submission = any(concern == "non submission")  )

heppp_covered <- flags |>
  filter(campaign == "pre census") %>%
  filter(!str_detect(reason_for_no_upload, "[Mm]istake")
         | is.na(reason_for_no_upload)) %>%
  filter(session %in% SESSIONS) %>%
  distinct(offering)

fla_agg <- enrolments %>%
  inner_join(heppp_covered) %>%
  distinct(id, session) %>%
  left_join(fla_agg) %>%
  mutate(
    n_flags = replace_na(n_flags, 0),
    any_non_submission = replace_na(any_non_submission, FALSE)
  )

flaca_agg <- full_join(fla_agg, aca_agg) %>%
  mutate(
    any_complete = replace_na(any_complete, FALSE),
    retention_covered = !is.na(n_flags)
  )

rm(fla_agg, heppp_covered, aca_agg)

calls_agg <- contact %>%
  filter(campaign == "pre census", contact_type == "call", session %in% SESSIONS) %>%
  group_by(id, session) %>%
  summarise(
    dialogue = any(contact_result == "dialogue"),
    hour_of_day = lubridate::hour(max(timestamp, na.rm = TRUE)),
    .groups = "drop"
  )

sms_agg <- contact %>%
  filter(
    campaign == "pre census",
    contact_type == "sms",
    session %in% SESSIONS,
    contact_result == "dialogue") %>%
  distinct(id, session) %>%
  mutate(
    sms_dialogue = TRUE
  )

con_agg <- full_join(calls_agg, sms_agg) %>%
  mutate(
    dialogue = replace_na(dialogue, FALSE),
    sms_dialogue = replace_na(sms_dialogue, FALSE))
```

```r
flacacon_agg <- full_join(flaca_agg, con_agg)


dat <- flacacon_agg %>%
  inner_join(
    student_progress %>%
      select(id, session, commencing, attendance_type, gpa)
  ) %>%
  inner_join(student_demographics) %>%
  mutate( # refactoring key variables
    female = gender == "Female",
    domestic = domesticity == "Domestic",
    indigenous = atsi == "Australian Indigenous",
    nesb = nesb == "NESB",
    low_ses = ses == "Low SES",
    disability = disability_support_status != "no Disability",
    parent_uni_ed = parental_education == "University Level",
    regional_or_remote = str_detect(tolower(remoteness), "regional|remote"),
    on_campus = attendance_type == "Internal",
    commencing = commencing == "Commencing",
    summer_course = session %% 100 == 90) %>%
  add_year_from_session() %>%
  select(-gender, -domesticity, -atsi, -firstname, -lastname, -atar_group,
         -parental_education, -ses, -disability_support_status,
         -remoteness, -attendance_type) %>%
  select(no_npe, all_pass, dialogue, everything())

ids <- unique(dat$id)

anon_ids <- tibble(id =ids) |>
  mutate(
    student = paste0("Student", row_number()))

datn <- dat |>
  inner_join(anon_ids) |>
  select(-id) |>
  mutate(across(-student, as.numeric))

rm(dat, anon_ids, ids, calls_agg, con_agg, flaca_agg, flacacon_agg, sms_agg)
mod.simple.glm <- glm(
  formula = no_npe ~ dialogue,
  data = datn,
  family = binomial(link = "logit")
)

mod.simple.glm.pass <- glm(
  formula = all_pass ~ dialogue,
  data = datn,
  family = binomial(link = "logit")
)

salad_formula_npe = as.formula(no_npe ~ dialogue + domestic + commencing +
                               indigenous + low_ses + regional_or_remote + age +
```

```
                              summer_course + any_non_submission)
salad_formula_pass = as.formula(all_pass ~ dialogue + domestic + commencing +
                              indigenous + low_ses + regional_or_remote + age +
                              summer_course + any_non_submission)

mod.salad.glm <- glm(
  formula = salad_formula_npe,
  data = datn,
  family = binomial(link = "logit")
)

mod.salad.glm.pass <- glm(
  formula = salad_formula_pass,
  data = datn,
  family = binomial(link = "logit")
)
set.seed(8991)

datn_cln <- datn %>%
  filter(
    !is.na(hour_of_day),
    !is.na(dialogue),
    hour_of_day > 8 # early values likely from calls made previous day
    )

train <- datn %>%
  group_by(session) %>%
  slice_sample(prop = 0.6)

test <- datn %>%
  anti_join(train, by = c("student", "session"))

mod.propensity <- glm(
  formula = dialogue ~ indigenous + commencing + on_campus + hour_of_day,
  data = train,
  family = binomial(link = "logit")
)

test_predictions <- test %>%
  bind_cols(tibble(
    propensity = predict(mod.propensity, test, type = "response")
  ))

train_predictions <- train %>%
  bind_cols(tibble(
    propensity = predict(mod.propensity, train, type = "response")
  ))

dat_prop <- train_predictions %>%
  mutate(dataset = "train") %>%
  bind_rows(test_predictions %>% mutate(dataset = "test")) %>%
  mutate(dataset = fct_relevel(dataset, "train")) %>%
  arrange(propensity) %>%
```

```
  select(no_npe, all_pass, dialogue, student, session, dataset, propensity,
         any_non_submission) %>%
  mutate(weight = 1 / propensity)

dat_prop %>%
  ggplot(aes(x = propensity, fill = factor(dialogue), group = dialogue)) +
  geom_density(alpha = 0.5, position = "dodge") +
  # geom_point(aes(y = dialogue, color = factor(dialogue))) +
  xlim(c(0,1)) +
  theme_minimal() +
  viridis::scale_fill_viridis(
    option = "G",
    discrete = T,
    labels = c("No dialogue", "Dialogue"),
    name = "") +
  viridis::scale_color_viridis(option = "G", discrete = T) +
  xlab("Propensity score") +
  facet_grid(dataset ~ .) +
  ggtitle("Fitting propensity score model")
WEIGHT_SCALE <- 10 # a bit hacky to use binomial regression

glm_causal <- function(d) {
  glm(
    no_npe ~ dialogue,
    data = d,
    weights = round(d$weight * WEIGHT_SCALE, 0),
    family = binomial(link = "logit")
  )
}
glm_causal.pass <- function(d) {
  glm(
    all_pass ~ dialogue,
    data = d,
    weights = round(d$weight * WEIGHT_SCALE, 0),
    family = binomial(link = "logit")
  )
}

mod.causal <- glm_causal(dat_prop)
mod.causal.pass <- glm_causal.pass(dat_prop)

plot_models(
  mod.simple.glm,
  mod.salad.glm,
  mod.causal,
  # axis.lim = c(1, 3),
  rm.terms = c("gpa", "age", "summer_course",
               "low_ses", "regional_or_remote",
               "indigenous", "domestic", "commencing", "any_non_submission"),
  # terms = c("dialogueTRUE"),
  m.labels = c(
    "NPE: Single predictor regression",
    "NPE: Causal salad",
```

```
    "NPE: Inverse propensity score weighted"),
  spacing = 0.3,
  legend.title = "Models",
  title = "Model comparisons of dialogue effect on no NPE",
  show.values = T,
  show.p = F)

plot_models(
  mod.simple.glm.pass,
  mod.salad.glm.pass,
  mod.causal.pass,
  # axis.lim = c(1, 3),
  rm.terms = c("gpa", "age", "summer_course",
               "low_ses", "regional_or_remote",
               "indigenous", "domestic", "commencing", "any_non_submission"),
  # terms = c("dialogueTRUE"),
  m.labels = c(
    "Pass: Single predictor regression",
    "Pass: Causal salad",
    "Pass: Inverse propensity score weighted"),
  spacing = 0.3,
  legend.title = "Models",
  title = "Model comparisons of dialogue effect on Passing",
  show.values = TRUE, show.p = FALSE)


jtools::plot_summs(
  mod.causal,
  mod.simple.glm,
  mod.salad.glm,
  model.names = c("IPSW", "Simple", "Salad"),
  inner_ci_level = 0.9,
  coefs = "dialogue",
  exp = TRUE)

jtools::plot_coefs(mod.causal)
library(brms)
library(jtools)

brm.simple.pass <- brm(
  formula = all_pass ~ dialogue,
  data = datn_cln,
  family = bernoulli()
)

mcmc_plot(brm.simple.pass)

brm.salad.pass <- brm(
  formula = all_pass  ~ dialogue + commencing + age + nesb + female + domestic + indigenous + low_ses +
  data = datn_cln,
  family = bernoulli()
)
```

```r
mcmc_plot(brm.salad.pass)

plot_summs(brm.salad.pass, inner_ci_level = 0.9)

brm.ipsw.pass <- brm(
  formula = all_pass | weights(weight) ~ dialogue,
  data = dat_prop,
  family = bernoulli()
)

datn_cln_sesh <- dat_prop %>%
  mutate(
    s1 = as.numeric(session == 202030),
    s2 = as.numeric(session == 202060),
    s3 = as.numeric(session == 202090)
  )

brm.ipsw.pass.sesh <- brm(
  formula = all_pass | weights(weight) ~ dialogue + s1 + s2 + s3,
  data = datn_cln_sesh,
  family = bernoulli()
)

mcmc_plot(brm.ipsw.pass.sesh)

# slow!!!
# brm.ipsw.pass.sesh <- brm(
#   formula = all_pass | weights(weight) ~ dialogue + (1 | s1 + s2 + s3),
#   data = datn_cln_sesh,
#   family = bernoulli()
# )

mcmc_plot(brm.ipsw.pass.sesh)


labs = knitr::all_labels()
labs = setdiff(labs, c("setup", "banner", "sql-query")) # removes irrelevant chunks
```