Supplementary Materials for *Comparing the Effectiveness of a Brief Intervention to Reduce Unhealthy Alcohol Use Across Adult Primary Care Patients with and without Depression: A Machine Learning Approach with Augmented Inverse Probability Weighting*

Table of Contents

Appendix 1: Electronic Health Records Codes

Item	Code Type and Code(s)
Alcohol Brief Intervention (BI)	ICD-9 (V65.42 and V65.49 before October 2015) ICD-10 codes (Z71.41 and Z71.89 after October 2015) Current Procedural Terminology codes (96160, 99420, 99408, and 99409) Healthcare Common Procedure Coding System codes (G0396, G0397, G0443, and H0050)
Alcohol Use Disorders	ICD9 (291*, 303*, 305.0*) ICD10 (F10.1*m F10.2*, F10.9*)
Depression	ICD-9 (296.2*, 296.3*, 296.82, 300.4, 301.12, 309.0, 309.1, 309.28) ICD-10 (F32*, F33*, F34.1, F43.21, F43.23)
Drug Use Disorders	ICD9 (292*, 304*, 305* except 305.0*) ICD10 (F11, F11.1*, F11.2*, F11.9*, F12, F12.1*, F12.2*, F12.9, F13, F13.1*, F13.2*, F13.9*, F14, F14.1*, F14.2*, F14.9*, F15, F15.1*, F15.2*, F15.9*, F16, F16.1*, F16.2*, F16.9*, F18, F18.1*, F18.2*, F18.9*, F19, F19.1*, F19.2*, F19.9*)
Other Mental Health Conditions	

	Anxiety disorders	ICD9 (300.0*, 300.21, 300.3, 309.21, 309.24, 309.81) ICD10 (F40.01, F41*, F42*, F43.1*, F43.22)
	Bipolar disorders	ICD9 (296.0*, 296.1*, 296.4*, 296.5*, 296.6*, 296.7*, 296.80, 296.81, 296.89, 301.13) ICD10 (F30*, F31*, F34.0)
	Eating disorders	ICD9 (307.1, 307.51) ICD10 (F50.0*, F50.2)
	Other mood disorders	ICD9 (293.83, 296.9, 296.90, 296.99, 799.24) ICD10 (F34.8, F34.89, F34.9, R45.86, F06.3*)
	Pervasive developmental disorders	ICD9 (299*) ICD10 (F84*)
	Schizophrenia, schizoaffective disorders and other psychosis	ICD9 (295*, 297*, 298*) ICD10 (F20*, F22*, F23*, F25*, F28*, F29*)

Augmented Inverse Probability Weighting

Here we briefly outline the construction of our estimators and required assumptions. For a didactic explanation of these kinds of methods see Schuler & van der Laan (Schuler and van der Laan, 2022) .

Setup Let Y, D, A and X represent our outcome (possibly missing), an indicator of outcome observed, an indicator of treatment, and a vector of covariates. Let $Y(a)$ represent the potential outcome that would obtain if treatment were forced to $A = a$. Define $\psi_a^* = E[Y(a)]$ to be the counterfactual population average outcomes. The causal average treatment effect (ATE) is defined as $\psi^* = \psi_1^* - \psi_0^*$. For notational convenience define

- $\mu_a(X) = E[Y(a)|X]$ (conditional potential outcome means)
- $\pi_a(X) = P\{A = a|X\}$ (propensity scores)
- $\delta(X) = P\{D = 1|X\}$ (conditional probability of being observed)

Identification The measured outcome $Y = D \times Y(A)$ depends on which treatment is given and on the observation indicator (without loss of generality we arbitrarily say $Y = 0$ when it goes unobserved). We can identify the causal ATE under the following assumptions:

- $P\{A = a|X = x\} > 0 \ \ \forall \ a, x$ (treatment positivity)
- $(Y(a) \perp A)|X$ (treatment unconfounded)
- $P\{D = 1|A = a, X = x\} > 0 \ \ \forall a, x$ (missingness positivity)
- $D \perp Y(a)|X, A$ (missingness unconfounded)

Given these, standard conditioning arguments show that $E[Y|D = 1, A = a, X] = \mu_a(X)$. Define the statistical counterfactual mean $\psi_a = E[E[Y|D = 1, A = 1, X]]$ and define the

statistical ATE to be $\psi = \psi_1 - \psi_0$ (which is equal to the *causal* ATE ψ^* when our identifying assumptions hold).

Inference Standard derivation techniques (Kennedy, 2022) show that the efficient influence function for ψ_a is

$$\phi_a = \frac{D1_a(A)}{\delta(X)\pi_a(X)}(Y - \mu_a(X)) + (\mu_a(X) - \psi_a)$$

and the efficient influence function for ψ is therefore $\phi_1 - \phi_0$.

We can thus obtain an efficient estimating equations (i.e. AIPW-style) estimator

$$\hat{\psi}_a = \frac{1}{n}\sum_i \frac{D_i 1_a(A_i)}{\hat{\delta}(X_i)\hat{\pi}_a(X_i)}(Y_i - \hat{\mu}_a(X_i)) + \hat{\mu}_a(X_i)$$

where the hat quantities are cross-fit estimates of their true counterparts. We obtain our estimate of the ATE by taking $\hat{\psi} = \hat{\psi}_1 - \hat{\psi}_0$.

By standard theory, this estimator is asymptotically normal with asymptotic sampling variance $V[\phi]$. We can therefore obtain a consistent estimate $\hat{\sigma}^2_\infty$ by taking the empirical sample variance of the estimated influence function $\hat{\phi} = \hat{\phi}_1 - \hat{\phi}_0$ where

$$\hat{\phi}_a = \frac{D1_a(A)}{\hat{\delta}(X)\hat{\pi}_a(X)}(Y - \hat{\mu}_a(X)) + (\hat{\mu}_a(X) - \hat{\psi}_a)$$

.

An estimate of the finite-sample sampling variance is therefore $\hat{\sigma}^2 = \hat{\sigma}^2_\infty / n$, which we can use to build confidence intervals (e.g. 95% CI is $\hat{\psi} \pm 1.96 \times \hat{\sigma}$) and compute p-values (use a Z-test to compare the estimated effect to the null $H_0 : \hat{\psi} \sim \mathcal{N}(0, \hat{\sigma}^2)$).

Difference in ATEs Let G represent a moderator of interest, which is one of the covariates in $X = [X_1 \dots G \dots X_p]$. Let $\psi^*_{a,g} = E[Y(a)|G = g]$ and define a difference in causal ATEs between two groups $G = 0$ and $G = 1$ to be $(\psi^*_{a=1,g=1} - \psi^*_{a=0,g=1}) - (\psi^*_{a=1,g=0} - \psi^*_{a=0,g=0})$. This transparently and nonparametrically represents a measure of the extent to which G moderates the causal effect of A on Y.

Identification proceeds along the same lines as the standard ATE. Again using standard techniques we obtain that the efficient influence function for this estimand is $\phi = (\phi_{a=1,g=1} - \phi_{a=0,g=1}) - (\phi_{a=1,g=0} - \phi_{a=0,g=0})$ where

$$\phi_{a,g} = \frac{1_g(G)}{\gamma_g} \left[\frac{D 1_a(A)}{\delta(X)\pi_a(X)} (Y - \mu_a(X)) + (\mu_a(X) - \psi_{a,g}) \right]$$

and we define the population group proportion $\gamma_g = P\{G = g\}$. The appropriate efficient estimating equations estimator and inference follow immediately in similar fashion to the above.

Appendix 3. *R* Analysis Code

I. Function for the machine learning analyses

```
model_avs <- function(preprocess, features, outcome, treatment, estimand) {
  # load preprocessed data
  df <- preprocess
  # select outcome distribution
  distribution <- ifelse(grepl("BI|uncensored", outcome), "bernoulli",
"gaussian")
  # add treatment indicator for uncensored models
  if(grepl("uncensored", outcome) == TRUE) {features <- append(features,
"BI")}
  cat("Estimand:", estimand, "\nFeatures:", features)
  # run analysis
  Sys.sleep(5)
  h2o.init(nthreads = 32, max_mem_size = "64G")
  h2o.removeAll()
  list_preds <- list()
  for(i in 1:5){
    # for treatment outcome models (mu1 and mu0), training uses uncensored
cases that received the specific treatment,
    # trained model is applied to entire test set
    h2o_train <- as.h2o(df[df$fold != i & is.na(df[,outcome]) == FALSE &
df$BI %in% treatment,])
    h2o_test <- as.h2o(df[df$fold == i, features])
    h2o_model <- h2o.automl(x = features,
                            y = outcome,
                            max_models = 50,
                            training_frame = h2o_train,
                            keep_cross_validation_predictions = TRUE,
                            fold_column = "fold",
                            distribution = distribution,
                            seed = 2001,
                            verbosity = "error"
                            )
    h2o_model_best <- h2o.get_best_model(h2o_model, algorithm =
"stackedensemble")
    h2o.saveModel(h2o_model_best, path = "/bd-fs-
mnt/sam5_root/sasg_project/sri/projects/papini/git/sbirt/models/", filename =
paste("model", outcome, i, sep = "_"), force = TRUE)
    df_preds_raw <- as.data.frame(h2o.predict(h2o_model_best, h2o_test))
    # extract predictions for binary and continuous outcomes
    prediction <- ifelse("p1" %in% names(df_preds_raw), "p1", "predict")
    df_preds <- data.frame(id = df[df$fold == i, "id"], y =
df_preds_raw[,prediction])
    names(df_preds) <- c("id", estimand)
    list_preds[[i]] <- df_preds
    }
  # save predictions
  df_preds <- do.call(rbind, list_preds)
```

```
  filename <- paste("/bd-fs-
mnt/sam5_root/sasg_project/sri/projects/papini/git/sbirt/predictions/",
estimand, ".rds", sep = "")
  saveRDS(df_preds, filename)
  h2o.shutdown(prompt = FALSE)
  return(filename)
}
```

II. Inputs for each model

```
# treatment model
model_avs(preprocess, features, "BI", c("1", "0"), "pi")
# follow-up participation model
model_avs(preprocess, features, "uncensored", c("1", "0"), "delta")
# outcome models
model_avs(preprocess, features, "uncensored", c("1", "0"), "delta")
## change in drinking days
model_avs(preprocess, features, "drinking_days_delta", "1",
"mu1_drinking_days_delta")
model_avs(preprocess, features, "drinking_days_delta", "0",
"mu0_drinking_days_delta")
## change in heavy drinking days
model_avs(preprocess, features, "binge_days_delta", "1",
"mu1_binge_days_delta")
model_avs(preprocess, features, "binge_days_delta", "0",
"mu0_binge_days_delta")
## change in drinks per drinking day
model_avs(preprocess, features, "drinks_day_delta", "1",
"mu1_drinks_day_delta")
model_avs(preprocess, features, "drinks_day_delta", "0",
"mu0_drinks_day_delta")
## change in drinks per week
model_avs(preprocess, features, "drinks_week_delta", "1",
"mu1_drinks_week_delta")
model_avs(preprocess, features, "drinks_week_delta", "0",
"mu0_drinks_week_delta")
```

III. AIPW Estimates

```
# df: dataframe with columns:
#  DATA:
#     Y: the continuous outcome (except NA in rows where D=1)
#     A: binary exposure: 1="exposed", 0="control"
#     D: outcome observation indicator: 1="outcome observed", 0="outcome
missing" (df$D = !is.na(df$Y))
#     X: covariates (multiple columns)
#     G: binary group indicator (also counts as a covariate, e.g. use as a
predictor when fitting models below)
#  PREDICTIONS:
#     mu1: outcome prediction assuming exposure
#     mu0: outcome prediction assuming no exposure
#     pi1: predicted probability of exposure
#     pi0: predicted probability of no exposure (same as 1-pi1)
#     delta: predicted probability of observing outcome
```

```
list_outcomes <- c("drinking_days_delta", "binge_days_delta",
"drinks_week_delta", "drinks_day_delta")
list_results <- list()
for(i in 1:length(list_outcomes)){
  outcome <- list_outcomes[[i]]
  df$Y <- df[ ,paste0(outcome)]
  df$A <- as.numeric(as.character(df$BI))
  df$D <- as.numeric(as.character(df$uncensored))
  df$G <- ifelse(df$dep0 == "Y", 1, 0)
  df$mu1 <- df[ ,paste0("mu1_", outcome)]
  df$mu0 <- df[ ,paste0("mu0_", outcome)]
  df$pi1 <- df$pi
  df$pi0 <- 1- df$pi
  df$delta <- df$delta

  # Run AIPW estimators
  df_xi = df %>%
    mutate(Y = ifelse(is.na(Y), 0, Y)) %>%  # replace NAs with 0 so that NA *
(D==0) = 0
    mutate(
      xi1 = (D*(A==1))*(Y-mu1) / (delta*pi1)  +  mu1,
      xi0 = (D*(A==0))*(Y-mu0) / (delta*pi0)  +  mu0
      )

  psi_ = list()
  phi_ = list()
  for(g in c(0,1)) {
    df_g = df_xi %>% filter(G==g)
    psi_[[paste0("a1g",g)]]  = df_g %$% mean(xi1)
    psi_[[paste0("a0g",g)]]  = df_g %$% mean(xi0)
    phi_[[paste0("a1g",g)]]  = df_xi %$% (((G==g)/mean(G==g)) * (xi1 -
psi_[[paste0("a1g",g)]]))
    phi_[[paste0("a0g",g)]]  = df_xi %$% (((G==g)/mean(G==g)) * (xi0 -
psi_[[paste0("a0g",g)]]))
    }

  # ATE all
  psi_all <- mean(df_xi$xi1) - mean(df_xi$xi0)
  sigma_all <- sqrt( sum(((df_xi$xi1 - df_xi$xi0) - psi_all)^2)/nrow(df)^2 )

  # ATE depressed
  psi_g1 <- psi_[[paste0("a1g1")]] - psi_[[paste0("a0g1")]]
  df_g1 <- df_xi %>% filter(G==1)
  sigma_g1 <- sqrt( sum(((df_g1$xi1 - df_g1$xi0) - psi_g1)^2)/nrow(df_g1)^2 )

  # ATE non-depressed
  psi_g0 <- psi_[[paste0("a1g0")]] - psi_[[paste0("a0g0")]]
  df_g0 <- df_xi %>% filter(G==0)
  sigma_g0 <- sqrt( sum(((df_g0$xi1 - df_g0$xi0) - psi_g0)^2)/nrow(df_g0)^2 )

  # Difference in ATE (depressed - non-depressed)
  psi_g1minusg0 <- psi_g1 - psi_g0
  phi_g1minusg0 <- ( phi_[[paste0("a1g1")]] - phi_[[paste0("a0g1")]] )  -  (
phi_[[paste0("a1g0")]] - phi_[[paste0("a0g0")]] )
  sigma_g1minusg0 <- sqrt( mean(phi_g1minusg0^2) / nrow(df) )
```

```
  # Compile results
  results <- data.frame(outcome = outcome,
                        comparison = c("All", "Depressed", "Non-depressed",
"Depressed -\nNon-depressed"),
                        ATE = round(c(psi_all, psi_g1, psi_g0,
psi_g1minusg0), 2),
                        CI = NA,
                        ci.lo = NA,
                        ci.hi = NA,
                        z = NA)
  list_models <- c("all", "g1", "g0", "g1minusg0")
  for(j in 1:nrow(results)){
    psi <- get(paste0("psi_", list_models[[j]]))
    sigma <- get(paste0("sigma_", list_models[[j]]))
    results[j, "CI"] <- paste0("[", round(psi - 1.96*sigma, 2), ", ",
round(psi + 1.96*sigma, 2), "]")
    results[j, "ci.lo"] <- round(psi - 1.96*sigma, 2)
    results[j, "ci.hi"] <- round(psi + 1.96*sigma, 2)
    results[j, "z"] <- round(psi/sigma, 3)
  }

list_results[[i]] <- results
}

# combine all results
results <- do.call(rbind, list_results)
# add p-values
results$p <- round(2*pnorm(q=abs(results$z), lower.tail=FALSE), 4)
```

Appendix 4: AIPW Estimates of the Average Treatment (BI) Effect

Outcome	**Comparison**	**ATE**	**95% CI**	**z**	***p***
Change in Drinking Days					
	All	-0.05	[-0.08, -0.02]	-3.33	< .001
	Depressed	-0.01	[-0.11, 0.09]	-0.2	.84
	Non-depressed	-0.05	[-0.08, -0.02]	-3.44	< .001
	Depressed - Non-depressed	0.04	[-0.06, 0.15]	0.78	.44
Change in Heavy Drinking Days					
	All	-0.41	[-0.59, -0.22]	-4.26	< .001
	Depressed	-0.33	[-1.25, 0.58]	-0.71	.48
	Non-depressed	-0.41	[-0.6, -0.23]	-4.38	< .001
	Depressed - Non-depressed	0.08	[-0.85, 1.02]	0.17	.87
Change in Drinks per Week					
	All	-0.17	[-0.28, -0.05]	-2.88	.004
	Depressed	-0.07	[-0.54, 0.40]	-0.29	.77
	Non-depressed	-0.18	[-0.29, -0.06]	-2.97	.003
	Depressed - Non-depressed	0.11	[-0.37, 0.59]	0.44	.66
Change in Drinks per Drinking Days					
	All	-0.06	[-0.09, -0.03]	-4.06	< .001
	Depressed	-0.06	[-0.17, 0.05]	-1.08	.28
	Non-depressed	-0.06	[-0.1, -0.03]	-3.92	< .001
	Depressed - Non-depressed	0	[-0.11, 0.12]	0.07	.95