Final Project Code

MA 590 Special Topics: Causal Inference

Aukkawut Ammartayakun

16 February, 2023

Implementation of Classifier Two-Sample Test (C2ST)[1]

The algorithm itself is to generate the training and testing set of combined dataset (from treatment and control group). Then, fit the classifier (like logistic regression) to the training set and predict the testing set. The empirical loss should be really close to near-chance level (bootstrap loss) if there is no difference between treatment and control group.

```
# based on https://gist.github.com/oddskool/409018f61d432f10fe00223e2b93cb51
ttsplit <- function(X,y, p = 0.8){
    #test train split
    train <- sample(1:nrow(X), size = floor(p*nrow(X)), replace = FALSE) # 80% of data for training
    test <- setdiff(1:nrow(X), train) # 20% of data for testing
    X_train <- X[train,]
    X_test <- X[test,]
    y_train <- y[train]
    y_test <- y[test]
    return(list(X_train = X_train, X_test = X_test, y_train = y_train, y_test = y_test))
}
c2st <- function(X, y, bf = 100){
    # make sure X is dataframe
    X <- as.data.frame(X)
    # split data into training and testing sets
    X_train <- ttsplit(X, y)$X_train
    X_test <- ttsplit(X, y)$X_test
    y_train <- ttsplit(X, y)$y_train
    y_test <- ttsplit(X, y)$y_test

    # fit logistic regression model
    model <- glm(y_train ~ ., data = X_train, family = binomial(link = "logit"))
    y_pred <- predict(model, X_test, type = "response")
    y_pred <- ifelse(y_pred > 0.5, 1, 0)
    emp_loss <- mean(y_pred != y_test)

    # bootstrap random target
    y_bar <- mean(y)
    bs_losses <- c()
    for (b in 1:bf){
        y_random <- rbinom(nrow(X), 1, y_bar)
        #test train split
        X_train <- ttsplit(X, y_random)$X_train
        X_test <- ttsplit(X, y_random)$X_test
```

```
        y_train <- ttsplit(X, y_random)$y_train
        y_test <- ttsplit(X, y_random)$y_test
        model <- glm(y_train ~ ., data = X_train, family = binomial(link = "logit"))
        y_pred <- predict(model, X_test, type = "response")
        y_pred <- ifelse(y_pred > 0.5, 1, 0)
        bs_losses <- c(bs_losses, mean(y_pred != y_test))
    }

    # calculate p-value
    # need percentile function
    pc <- mean(bs_losses >= emp_loss) # wrong...
    pvalue <- ifelse(pc < y_bar, pc, 1 - pc)
    return(list(emp_loss = emp_loss, pvalue = pvalue))
}
```

Let's test the algorithm with the random data.

```
#set seed
set.seed(1337)
n <- 1000
X <- data.frame(x1 = rnorm(n,0,1), x2 = rnorm(n,0,1))
y <- rbinom(n, 1, 0.5)
c2st(X, y)
```

```
## $emp_loss
## [1] 0.44
##
## $pvalue
## [1] 0.09
```

```
set.seed(1337)
n <- 1000
X <- data.frame(x1 = rnorm(n,2,3), x2 = rexp(n,4))
y <- rbinom(n, 1, 0.5)
c2st(X, y)
```

```
## $emp_loss
## [1] 0.465
##
## $pvalue
## [1] 0.15
```

Reference

[1] https://arxiv.org/abs/1905.12837