BIOS 740 - Computing HW 1

Andrew Walther

September 15, 2021

1 Introduction

Precision Medicine is a sub-field of machine learning and statistics that has received significantly increased attention and research efforts in recent years. The field hinges around the goal of using observed characteristics about patients and populations to develop data-driven individualized treatment rules that are generalizable and reproducible. These individualized treatment rules involve compiling relevant information about patients and utilizing that data, along with a selection of available treatments to select the appropriate treatment that maximizes an outcome of interest with a treatment rule. Furthermore, effective treatment rules (or regimes) can be applied to larger groups of patients to provide a beneficial outcome across the whole sample. In short, individualize treatment rules or regimes (ITRs) seek to use data to determine who to apply a treatment to and how to do so in order to achieve an optimized clinical outcome. Recently, techniques such as random forests[1][3], doubly robust augmented inverse probability weighted estimators[5][4], and residual weighted learning[6][4] have been developed and applied to precision medicine problems as methods to estimate individualized treatment rules.

Here, we investigate the application of the aforementioned methods to data from the AIDS Clinical TRIALS Group Study 175 (ACTG175), a randomized clinical trial where single therapy techniques with the drugs zidovudine & didanosine were evaluated against dual therapies of zidovudine and didanosine or zidovudine and zalcitabine. These treatments were given to adults with HIV type I with CD4 T Cell counts in a pre-specified range[2]. The ACTG175 data set consists of 2139 observations with 27 recorded factors. The factors that are included in this particular analysis are noted in Table 1.

Table 1: ACTG175 Data Description

Factor	Description
age	age in years at baseline
wtkg	weight in kg at baseline
hemo	hemophilia (0=no, 1=yes)
homo	homosexual activity (0=no, 1=yes)
drugs	history of intravenous drug us (0=no, 1=yes)
karnof	Karnofsky score (scale of 0-100)
race	race (0=white, 1=non-white)
gender	gender (0=female, 1=male)
strat	antiretroviral history stratification
symptom	symptomatic indicator (0=asymptomatic, 1=symptomatic)
cd40	CD4 T cell count at baseline
cd496	CD4 T cell count at 96 weeks
cd80	CD8 T cell count at baseline
arms	treatment arm (1=zidovudine & didanosine, 2=zidovudine & zalcitabine)

Additionally, the original study contained 2139 observations. However, it has previously been determined that the combination therapies outperform both monotherapy treatments so this work will only consider patients assigned to the combination therapy arms. Finally the outcome of interest in this trial is the CD4 T cell count (cells/mm^3) at the time point of 96 weeks into the study. In general, cell counts with larger magnitudes are considered to be “better”.

2 Diagnostics

The ACTG175 data set contains no missing observations so no missing data imputation was required to be completed. Additionally, the set of observations was reduced to consist of only patients who received either of the combination therapies, Zidovudine 600mg daily & Didanosine 400mg daily or Zidovudine 600mg daily & Zalcitabine 2.25mg daily. As a note, Zalcitabine is also known as Dideoxycytidine.

As an assessment of the response variable CD496, a histogram plot suggested that there may be some right skew in the data. However, an additional plot of the log transformation of CD496 presented a serious amount of left skew in the distribution of observations. Subsequently, we conducted a Kolmogorov-Smirnov test for normality with H_0 : the distribution of CD496 follows a normal distribution and H_A : the distribution of CD496 does not follow a normal distribution. The test returned a p-value of $p = 0.1478$ so we failed to reject the null hypothesis and concluded that the response variable does in fact follow a normal distribution. Hence, for the sake of simplicity, no further transformations of the data were conducted.

Furthermore, we assessed the importance of all remaining covariates in the data and found that CD40, strat, wtkg, symptom, and karnof were considered to be most important for making predictions with regard to the outcome of interest, CD496. For the sake of clarity, "karnof" is representative of the Karnofsky score, which is a scaled response for an individuals level of impairment, health, and ability to take care of their self. A low score indicates that a patient is seriously ill or incapacitated and a high score means that a patient is living without any health difficulty. Regardless, all remaining covariates were included in the subsequent analysis of the three models of interest in this study.

3 Methods

As previously mentioned, the individualized treatment rule (ITR) methods of interest are random forests, doubly robust augmented inverse probability estimators, and residual weighted learning. In all cases, the goal is to select an optimal treatment, $\hat{d}^{opt}$ for each patient for a single-decision treatment plan and estimate the value function of that treatment plan, where the value function is represented by

$$\hat{V}\left(\hat{d}^{opt}\right) = \frac{\sum_{i=1}^{nte} Y_i I\left(A_i = \hat{d}^{opt}\left(X_i\right)\right) / P\left(A_i|X_i\right)}{\sum_{i=1}^{nte} I\left(A_i = \hat{d}^{opt}\left(X_i\right)\right) / P\left(A_i|X_i\right)}$$

where nte refers to the size of the test set, A_i is the selected treatment, the X_i are patient characteristics, and Y_i is the outcome value. Here, k-fold cross validation is utilized. After the patients who received the monotherapy treatments were removed, the data was split into 10 folds. Of these folds, one (approximately 10% of all observations) is held out as observations for model testing at each iteration of model evaluation. The remaining 9 folds or 90% of the data are used for model training. Each of the ten folds is subsequently held out for testing in order to minimize prediction error for optimal treatment. As an additional data processing step, we stratified the training data by treatment arm (1 & 2) for the random forest method prior to making outcome predictions on the full test data set (both arms 1 &) 2). This procedure was not necessary for AIPWE or RWL.

3.1 Random Forest

A random forest is a combination of numerous decision trees where each tree is grown based on randomly selected observations and randomly selected subsets of covariates. Outcomes in a random forest differ for classification problems and regression problems. In the former, a prediction is made based on the majority of the trees grown in the random forest. On the other hand, in a regression problem, predictions are made based on the average value of all trees in the random forest. Techniques such as pruning, random split selection, and bagging are utilized to improve classification accuracy for random forests[1]. Another key feature of random forests is that as the number of trees in a random forest gets large, the generalization error, or prediction error on test data, reaches a bound given by

$$PE^* = P_{X,Y}\left(mg\left(X,Y\right) < 0\right)$$

where we have an ensemble of classifiers $h_1(x), h_2(x), \cdots, h_k(x)$ and $mg\left(X,Y\right)$ is the margin function measuring how much the average number of votes for the correct class exceed the average number of votes for any other possible class. When the margin is large, there is in turn more confidence for a classification decision[1].In a random forest, accuracy is maximized when the randomness in the feature selection minimizes correlation between individual trees while also maintaining prediction strength.

For our implementation of the random forest algorithm, we stratified each training data set of the 10 folds into observations that were treated in arm 1 and observations in arm 2. Those observations were then use to train two separate random forests based on arms 1 and 2. Those fitted random forests were then used to make predictions on the whole set of testing observations (not stratified by arm). The resulting outcome gave estimated outcome predictions based an "assigned" treatment of arm 1 or arm 2. We then concluded that for each patient, the random forest prediction from the assigned arm with the largest value was the optimal treatment arm. Hence, a patient with a predicted value of 350 from arm 1 and 360 from arm 2 would be classified as having arm 2 as $\hat{d}^{opt}$. In the random forest model, we elected to consider all available covariates for training the model as well as maintaining default parameters such as 500 trees as the number of trees to grow. We had no reason to believe that any changes from the default parameters of the random forest method would be significantly beneficial aside from ensuring that the importance of predictors with respect to the outcome was assessed in training the random forest.

3.2 Augmented Inverse Probability Weighted Estimator (AIPWE)

The value function of the doubly robust augmented inverse probability weighted estimator is

$$\hat{V}_{AIPW}(d_\eta) = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{C_{d_\eta,i}Y_i}{\pi_{d_\eta},1\left(H_{1i};\eta_1,\hat{\gamma}_1\right)} - \frac{C_{d_\eta,i} - \pi_{d_\eta,1}(H_{1i};\eta_1,\hat{\gamma}_1)}{\pi_{d_\eta,1}(H_{1i};\eta_1,\hat{\gamma}_1)}Q_1\left(H_{1i}, d_1(H_{1i};\eta_1);\hat{\beta}_1\right)\right)$$

where C_{d_η} is an indicator of whether or not a patient received the appropriate treatment as defined by the treatment regime d_η, π_{d_η} is the propensity of getting a treatment consistent with the treatment regime d_η, and Q_1 is a proposed model for $Q_1(h_1, a_1) = E(Y|H_1 = h_1, A_1 = a_1)$[5]. It follows that estimation of the optimal treatment regime is based on maximizing this estimator to get $\hat{\eta}^{opt}$.

In our analysis, we implemented the AIPW estimator via the optimalSeq() function from the R package 'DynTxRegime'[4]. We defined the propensity model to be an intercept only model for the single decision point and followed the methods as directed in the package documentation. We set the main outcome model to be a regression model with all available covariates included in the model as the number of parameters was small enough that dimension reduction was not deemed to be necessary. Additionally, the regime function was set to be a function of the outcome of interest, CD496, with subsequent classifications into arms 1 or 2 based on a cutoff for the η_1 parameter. Finally, the domain of the optimalSeq function was set to match the domain of the CD496 response variable, the pop.size was set to be 100 based on recommendation by the documentation, and we set the starting value for η_1 to be 1 so that it could search over the entire domain of CD496 to find an appropriate "cutoff" point. Subsequently, estimates of the AIPW fit were compiled to determine an estimated value function for each fold of data and then an estimated value for all of the data along with counts for treatment classification.

3.3 Residual Weighted Learning (RWL)

For residual weighted learning, the decision function we wish to minimize is

$$f(x) = \frac{1}{n}\sum_{i=1}^{n}\frac{\hat{r}_i}{\pi(a_i, x_i)}T(a_i f(x_i)) + \frac{1}{2}\lambda\|f\|^2$$

where λ is a tuning parameter and $\hat{r}_i$ is an estimated residual[6]. Additionally, due to the fact that residuals may be negative, a non-convex loss function known as the smoothed ramp loss is specified by

$$T(u) = \begin{cases} 0 & u \geq 1 \\ (1-u)^2 & 0 \leq u < 1 \\ 2-(1+u)^2 & -1 \leq u < 0 \\ 2 & u < -1. \end{cases}$$

In our implementation of residual weighted learning, the rwl() function from 'DynTxRegime'[4] was used. This time, the propensity model was set to be a binomial glm model with all available covariates used as predictors for CD496. This was chosen in accordance with suggestions from the method documentation. Identical to the AIPWE, the main outcome model was again chosen to be a linear regression model with all available covariates as predictors. Inside the rwl function, the regime function was also chosen to be a function of all predictors, we selected the number of internal cross validation folds to be 1 for the sake of saving time, and we chose the estimator response to be continuous considering the continuous nature of CD496. This model also utilized the default linear kernel. Following evaluation of the model, we determined the average value function outcome across all of the data folds as well as the standard prediction error across folds. Lastly, we computed counts of which patients were assigned to each treatment arm as their specified optimal treatment.

4 Results

Following the implementation and evaluation of each proposed estimator, the following estimated value functions and standard error values were attributed to each estimation method in Table 2. Recall that the outcome of interest for this study is CD496 or the CD4 cell count at 96 weeks, which is measured in cells per cubic millimeter. The residual weighted learning method achieved the largest estimated value, but it also had the largest standard error between estimates in the different folds of observations. Furthermore, this estimator classified a 55% majority of all testing observations to be in arm 2. Next, the AIPW estimator had the second largest estimated value function of approximately 355 $cells/mm^3$ and the smallest standard error of only 2.48. Additionally, this estimator classified about 75% of test data observations to be in arm 2. Finally, the random forest estimator had the smallest estimated value function of just 349 $cells/mm^3$ with a standard error of 10.01 that was relatively close to that of the RWL estimator. This estimate was acquired from the majority of observations that were also selected to be in treatment arm 2 via the random forest estimator. Hence, for a treatment regime recommendation between the two proposed arms, treatment arm 2 consisting of zidovudine zalcitabine leads to the largest estimated values of CD4 cell count at 96 weeks.

Table 2: Estimated Outcome Values

Method	Estimated Value $\hat{V}$ ($cells/mm^3$)	Standard Error of $\hat{V}$
Random Forest	349.55	10.01
Augmented Inverse Probability Weighted Estimator	355.41	2.48
Residual Weighted Learning	369.81	12.61

5 Discussion of Advantages & Disadvantages

We see that our three estimators of interest (random forest, AIPWE, and RWL) all generated estimated value functions that are reasonably close to each other (within 5 percentage points). This is a clear indication that all three methods are valid means to predict the response outcome and classify optimal treatments for a given patient based on their characteristic information. Now, we consider some of the advantages and disadvantages of each of these estimation methods.

Estimation with random forests utilizes an ensemble of individually grown random decision trees and considers the average class specification given by the ensemble of trees. They can be used for both classification or regression problems. In classification problems, a decision is made based on a majority "vote" of all trees in the model. For regression, the outcome is simply the average response of the trees in the forest. One of the most advantageous aspects of random forests is their ability to reduce the prevalence of over fitting, which improves accuracy when applied to test data that hasn't been seen before. This in turn minimizes generalization error. Additionally, as previously mentioned, random forests are useful for both classification and regression and they also work well with continuous and categorical data since nodes of individual decision trees do not require a particular class of data. On the other hand, while singular decision trees are relatively easy to interpret, an ensemble of randomly grown trees in a random forest is much more difficult to interpret. Furthermore, specific to our estimation goal in this study, random forests do not have a direct method for selecting optimal treatment rules for individual patients. Rather, the optimal treatment must be determined outside of the random forest model.

AIPWE is a hybrid of both classification and regression. The propensity model is a logistic regression model and from the perspective of a missing data problem (when a single decision point is considered), a regression-based outcome is returned in the form of the value estimation. Additionally, the AIPW estimator can also be used from the perspective of classification. Hence, AIPWE is a hybrid method. Considering our estimation results, a strength of the AIPW estimator is that it had the smallest standard error in estimation among different folds. Another strength is that under the conditions of a randomized clinical trial (which ACTG175 is), the AIPW estimator is consistent and more efficient compared to an inverse probability weighted estimator[5]. However, implementation of the AIPWE method is far more complex than the development of a random forest, given that numerous parameters must be specified to generate a propensity model, main model, and regime decision function. Additionally, if the regression model (propensity score) is a poor representation of the data, treatment assignment based on the chosen regime may be unreliable.

Considering residual weighted learning, a positive aspect of RWL is that the various loss functions that can be altered in the estimator allow it to be robust to outliers (particularly with the smooth ramp loss). This method also stabilizes the variance of the value estimation function[6]. Additionally, RWL is amenable to continuous, binary, and count outcomes, which enhances its versatility to different types of data. We also note that the RWL method obtained the largest estimated value function for CD496 cell count compared to the models considered. However, RWL also had the largest standard error in value estimates among different folds. Additionally, it is imperative to properly specify parameters in the model for RWL and appropriate variable selection is necessary to achieve optimal results with RWL. Finally, residual weighted learning is a hybrid of regression and classification as the model ultimately allows for treatment classification, but prior steps consider residuals from a regression fit of the outcome of interest[6].

References

[1] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[2] Michal Juraska and Maintainer Michal Juraska. Package ‘speff2trial’. *R software*, 2010.

[3] Suggests RColorBrewer and Maintainer Andy Liaw. Package ‘randomforest’. *University of California, Berkeley: Berkeley, CA, USA*, 2018.

[4] K. A. Linn B. Zhang M. Davidian S. T. Holloway, E. B. Laber and Maintainer Shannon T. Holloway A. A. Tsiatis. Package ‘dyntxregime’. *R software*, 2020.

[5] Baqun Zhang, Anastasios A Tsiatis, Eric B Laber, and Marie Davidian. A robust method for estimating optimal treatment regimes. *Biometrics*, 68(4):1010–1018, 2012.

[6] Xin Zhou, Nicole Mayer-Hamblett, Umer Khan, and Michael R Kosorok. Residual weighted learning for estimating individualized treatment rules. *Journal of the American Statistical Association*, 112(517):169–187, 2017.

```
#BIOS 740 Computing HW 1 Code
#Andrew Walther
#9/15/2021

##Data & Packages
set.seed(420)
#for ML method implementation
library(DynTxRegime)
library(randomForest)
#data manipulation/visualization
library(tidyverse)
#summary statistics
library(mosaic)
#cross-validation
library(caret)
#AIDS Clinical Trials Group Study 175 data (ACTG175)
library(speff2trial)
data(ACTG175)

#select observations in only arms 1,2 & eliminate unused factors as
directed (included 'arms')
dataCD496 <- ACTG175 %>% filter(!is.na(cd496), arms %in% 1:2) %>%
        select(-pidnum,-str2, -offtrt, -cd420, -r, -cd820, -cens,
               -days, -treat, -oprior, -z30, -zprior, -preanti)
###########################################################################
##Preprocessing & TEST/TRAIN Sets
#remove observations where 'cd496 = 0' (drops 1 observation)
dataCD496 <- dataCD496[dataCD496$cd496 != 0, ]
#add column for log transform of CD4 cell count (check if this is
necessary)
data_log_cd496 <- dataCD496 %>% mutate(log_cd496 = log(cd496))

#Create 10 folds of the dataset and add to TEST/TRAIN lists
folds <- createFolds(dataCD496$cd496, 10)
TEST_DATA <- list()
TRAIN_DATA <- list()
for(i in 1:10){
        TEST_DATA[[i]] <- dataCD496[folds[[i]],]
        TRAIN_DATA[[i]] <- dataCD496[-folds[[i]],]}
###########################################################################
##Exploratory Analysis / Feature Selection
#summary stats for CD490 & log(CD4)
favstats(dataCD496$cd496)
favstats(data_log_cd496$log_cd496)

#hist of CD4 Cell counts, possible right skew in the data
dataCD496 %>% ggplot() + geom_histogram(aes(x=cd496),binwidth = 50)
+labs(title = "Histogram of CD4 cell counts",x = "CD4 Cell Count (cells/
mm^3)")
#hist of log(CD4 cell), now there's left skew in the data (check for
normality)
data_log_cd496 %>% ggplot() + geom_histogram(aes(x=log(cd496)),binwidth =
0.1)
```

```r
+labs(title = "Histogram of log CD4 cell counts",x = "CD4 Cell Count
(cells/mm^3)")

#K-S test for normality - H0: data follows Normal distribution (p>0.05 so
confirm CD496 is normal)
ks.test(dataCD496$cd496, "pnorm", mean=mean(dataCD496$cd496),
sd=sd(dataCD496$cd496))

#Feature Selection: Train RPart Model to compute variable importance to
decide what to keep
rPartMod <- train(cd496 ~ ., data=dataCD496, method="rpart")
rpartImp <- varImp(rPartMod)
print(rpartImp)
##########################################################################
##Random Forest
#Initialize lists to hold mean predictions
RF_PREDS1 <- list()
RF_PREDS2 <- list()
#loop over 10 data folds with Random forest method
for(i in 1:10){
        #filter just observations in arm 1
        TRAIN_DATA1 <- TRAIN_DATA[[i]][TRAIN_DATA[[i]]$arms==1,]
        #filter just observations in arm 2
        TRAIN_DATA2 <- TRAIN_DATA[[i]][TRAIN_DATA[[i]]$arms==2,]
        #train RF model on just arm 1 observations
        rf.obj1 <- randomForest(cd496 ~ ., data = TRAIN_DATA1,
                                importance = TRUE, proximity = TRUE)
        #train RF model on just arm 2 observations
        rf.obj2 <- randomForest(cd496 ~ ., data = TRAIN_DATA2,
                                importance = TRUE, proximity = TRUE)
        #predict on RF models with full set of test data
        RF_PREDS1[[i]] <- mean(predict(rf.obj1, newdata = TEST_DATA[[i]]))
        RF_PREDS2[[i]] <- mean(predict(rf.obj2, newdata = TEST_DATA[[i]]))}
#compute mean & sd for each fold prediction mean (larger mean will serve as
the value estimate)
mean.pred1 <- mean(as.numeric(RF_PREDS1))
mean.pred2 <- mean(as.numeric(RF_PREDS2))
se.pred1 <- sd(as.numeric(RF_PREDS1))
se.pred2 <- sd(as.numeric(RF_PREDS2))
##########################################################################
##Augmented Inverse Probability Weighted Estimator
aipw.estimates <- list()
opt.Tx <- list()
for(i in 1:10){
        #specify propensity model (moPropen)
        moPropen <- buildModelObj(model = ~ 1, solver.method = 'glm',
                                  solver.args = list('family'='binomial'),
                                  predict.method = 'predict.glm',
                                  predict.args = list(type='response'))
        #specify outcome model (moMain)
        moMain <- buildModelObj(model = ~age+wtkg+hemo+homo+drugs+
                                        karnof+race+gender+strat+symptom+cd40+cd80
                                , solver.method = 'lm')
        #specify regimes
        data <- TRAIN_DATA[[i]]
        regime <- function(eta1, data) {
                tst <- {data$cd496 > eta1}
                rec <- rep(1, nrow(x=data))
                rec[!tst] <- 2
                return( rec )}
        #inverse probability weighted estimator object
        fit.AIPW <- optimalSeq(moPropen = moPropen, moMain = moMain,
                               regimes = regime, data = data,
                               response = data$cd496, txName = 'arms',
                               Domains = cbind(1,1062),
                               pop.size = 100, starting.values = 1)
        #value function estimates & test data treatment assignments
        aipw.estimates[[i]] <- as.numeric(estimator(fit.AIPW))
        opt.Tx[[i]]<- optTx(fit.AIPW, newdata = TEST_DATA[[i]])}
#compute mean & sd of estimated value functions (mean = 355.4116, se =
2.482627)
mean.aipw <- mean(as.numeric(aipw.estimates))
se.aipw <- sd(as.numeric(aipw.estimates))
#count patients assigned to trt 1 or 2 and compute majority proportion
trt1.total <- 0
trt2.total <- 0
for(i in 1:10){
        trt1 <- count(opt.Tx[[i]]$optimalTx == 1)
        trt2 <- count(opt.Tx[[i]]$optimalTx == 2)
        trt1.total <- trt1.total + trt1
        trt2.total <- trt2.total + trt2}
print(trt1.total)
print(trt2.total)
print(trt2.total/(trt1.total+trt2.total))
##########################################################################
##Residual Weighted Learning
#Initialize lists to hold value estimates & optimal treatment outcomes
rwl.estimates <- list()
opt.Tx <- list()
for(i in 1:10){
        #specify propensity model w/ all parameters (moPropen)
        moPropen <- buildModelObj(model = ~age+wtkg+hemo+homo+drugs+
                                          karnof+race+gender+strat+symptom+cd40+cd80,
                                  solver.method = 'glm',
                                  solver.args = list('family'='binomial'),
                                  predict.method = 'predict.glm',
                                  predict.args = list(type='response'))
        #specify outcome model w/ all parameters (moMain)
        moMain <- buildModelObj(model = ~age+wtkg+hemo+homo+drugs+
                                        karnof+race+gender+strat+symptom+
                                        cd40+cd80, solver.method = 'lm')
        #residual weighted learning object (cvfolds = 1 for efficiency &
include all covariates)
        fit.rwl <- rwl(moPropen = moPropen, moMain = moMain, data =
TRAIN_DATA[[i]],
                       reward = TRAIN_DATA[[i]]$cd496, txName = 'arms',
                       regime = ~age+wtkg+hemo+homo+drugs+
                               karnof+race+gender+strat+symptom+cd40+cd80,
cvFolds = 1,
                       responseType = 'continuous')
        #value function estimates
        rwl.estimates[[i]] <- as.numeric(estimator(fit.rwl))
        #optimal treatment assignments from test data
        opt.Tx[[i]]<- optTx(fit.rwl, newdata = TEST_DATA[[i]])}
#compute mean & sd of estimated value functions (mean = 369.8081, se =
12.60558)
mean.rwl <- mean(as.numeric(rwl.estimates))
se.rwl <- sd(as.numeric(rwl.estimates))
#count patients assigned to trt 1 or 2 and compute majority proportion
trt1.total <- 0
trt2.total <- 0
for(i in 1:10){
        trt1 <- count(opt.Tx[[i]]$optimalTx == 1)
        trt2 <- count(opt.Tx[[i]]$optimalTx == 2)
        trt1.total <- trt1.total + trt1
        trt2.total <- trt2.total + trt2}
print(trt1.total)
print(trt2.total)
print(trt2.total/(trt1.total+trt2.total))
```