USASK Data Science BootcampT7:Multi-Task Prediction Using Stacking Algorithms (MTPS)Practice Questions

TA: Kyle GardinerTutor: Xiaowen Cao

June 14, 2023. (1:00pm-2:30pm)

Continous Outcomes

(1) **Load `HIV` data from `MTPS` package**

```
library(MTPS)
data(HIV)
```

(2) **Converting `XX` and `YY` to matrices. Using only the first 3 outcomes from `YY` (`ABC`, `3TC`,and `AZT`).**

```
xmat<-as.matrix(XX)
ymat<-as.matrix(YY[,1:3])
```

(3) **Splitting the data 80/20 (NOTE: make sure to use `set.seed(123)`)**

```
set.seed(123)
nobs <- nrow(xmat)
training.id <- sample(seq_len(nobs), size = 0.8 * nobs)
y.train.continous <- ymat[training.id, ]
y.test.continous  <- ymat[-training.id, ]
x.train.continous <- xmat[training.id, ]
x.test.continous  <- xmat[-training.id, ]
```

(4) **Fitting MTPS model. Using the Residual stacking method (`cv=FALSE`,`residual=TRUE`). Using `glmnet1` in `method.step1` and `rpart1` in `method.step2`.**

```
fit.rs.continous <- MTPS(xmat = x.train.continous, ymat = y.train.continous, family = "gaussian",
                         cv = FALSE, residual = TRUE,
                         method.step1 = glmnet1,
                         method.step2 = rpart1)
```

(5) **Predicting values with `predict()`**

```
pred.rs.continous <- predict(fit.rs.continous, x.test.continous)
```

Binary Outcomes

(1) **Load `HIV` data from `MTPS` package**

```
library(MTPS)
data(HIV)
```

(2) **Converting `XX` and `YY` to matrices. Using only the last 2 outcomes from `YY` (`D4Tand DDI`).**

```
xmat<-as.matrix(XX)
ymat<-as.matrix(YY[,c(4,5)])
```

(3) **Converting outcome variables into binary (0 or 1)**

```
cutoffs<-c(1.5,1.5)
ymat.bin<-ymat
xmat.bin <- xmat
for(ii in 1:2) ymat.bin[,ii] <- (10^ymat[,ii] < cutoffs[ii])
```

(4) **Splitting the data 80/20 (NOTE: make sure to use `set.seed(123)`)**

```
set.seed(123)
nobs <- nrow(xmat.bin)
training.id <- sample(seq_len(nobs), size = 0.8 * nobs)
y.train.binary <- ymat.bin[training.id, ]
y.test.binary  <- ymat.bin[-training.id, ]
x.train.binary <- xmat.bin[training.id, ]
x.test.binary  <- xmat.bin[-training.id, ]
```

(5) **Fitting MTPS model. Using the Residual stacking method (`cv=FALSE,residual=TRUE`). Using `rpart1` in `method.step1` and `lm1` in `method.step2`. Specifying `resid.type="deviance"`**

```
fit.rs.binary <- MTPS(xmat = x.train.binary, ymat = y.train.binary, family = "binomial",
                              cv = FALSE, residual = TRUE,
                              method.step1 = rpart1,
                              method.step2 = lm1,
                              resid.type = "deviance")
```

(6) **Predicting values with `predict()`**

```
pred.rs.binary <- predict(fit.rs.binary, x.test.binary)
```

(7) **Print out confusion matrix for the 2 binary outcomes using the threshold of 0.5 for the predicted probabilities.**

```
for (jj in 1 : ncol(y.test.binary)) {
  print(colnames(y.test.binary)[jj])
  print(table((pred.rs.binary[,jj] > 0.5)*1, y.test.binary[,jj]))
}
```

```
## [1] "D4T"
##
##       0   1
##   0  89  21
##   1  23 117
## [1] "DDI"
##
##       0   1
##   0 116  29
##   1   9  96
```

Mixed Outcomes

(1) **Load `HIV` data from `MTPS` package**

```
library(MTPS)
data(HIV)
```

(2) **Converting `XX` and `YY` to matrices. Now using all 5 outcomes from `YY` (`ABC`, `3TC`, `AZT`, `D4T`, and `DDI`).**

```
xmat<-as.matrix(XX)
ymat<-as.matrix(YY)
```

(3) **Converting the first 3 outcomes to binary and leaving the last 2 outcomes as continous**

```
cutoffs <- c(2,3,3)
ymat.mix<-ymat
xmat.mix <- xmat
for(ii in 1:3) ymat.mix[,ii] <- (10^ymat[,ii] < cutoffs[ii])
```

(4) **Splitting the data 80/20 (NOTE: make sure to use `set.seed(123)`)**

```
set.seed(123)
nobs <- nrow(xmat.mix)
training.id <- sample(seq_len(nobs), size = 0.8 * nobs)
y.train.mix <- ymat.mix[training.id, ]
y.test.mix  <- ymat.mix[-training.id, ]
x.train.mix <- xmat.mix[training.id, ]
x.test.mix  <- xmat.mix[-training.id, ]
```

(5) **Fitting MTPS model. Using the Residual stacking method (`cv=FALSE`,`residual=TRUE`). Using `glmnet.lasso` in `method.step1` and `rpart1` in `method.step2`.**

```
fit.rs.mix <- MTPS(xmat = x.train.mix, ymat = y.train.mix,
                   family = c("binomial","binomial","binomial","gaussian","gaussian"),
                   cv = FALSE, residual = TRUE,
                   method.step1 = glmnet.ridge,
                   method.step2 = lm1)
```

(6) **Using `predict()` to get predicted values and probabilites.**

```
pred.mix.rs <- predict(fit.rs.mix, x.test.mix)
```