Custom Xgboost reg : squared-error

kipédène COULIBALY

19/04/2023

I) Analytical formula

The analytical formula of Mean Squared Error (MSE) is :

$$MSE = \frac{1}{n}\sum_{i=1}^{N}(Y_i - \hat{Y}_i)^2$$

- **Objective function** :

$$\begin{aligned} f(pred, label) &= \frac{1}{2}(pred - label)^2 \\ Grad &= (pred - label) \\ Hess &= 1 \end{aligned}$$

NB: Grad and Hess are vectors.

- **Evaluation metrics** :

Here we use two evaluation metrics: first, the Root Mean Square Error (RMSE) which is simply the square root of the MSE :

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{N}(Y_i - \hat{Y}_i)^2}$$

Then for a robust verification we use Mean Absolute Error (MAE) without implementing it (you should be able to do it without any problem) :

$$MAE = \frac{1}{n}\sum_{i=1}^{N}|Y_i - \hat{Y}_i|$$

II) Implementation with R

```
library(ISLR)
library(xgboost)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.4.2     v purrr   0.3.4
## v tibble  3.1.8     v dplyr   1.0.8
## v tidyr   1.2.1     v stringr 1.5.0
## v readr   2.1.3     v forcats 0.5.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## x dplyr::slice()  masks xgboost::slice()
```

```
library(Metrics)

# Data #
df = ISLR::Hitters %>% select(Salary, AtBat, Hits, HmRun, Runs, RBI, Walks,
                              Years, CAtBat, CHits, CHmRun, CRuns, CRBI, CWalks,
                              PutOuts, Assists, Errors)
df = df[complete.cases(df),]
train = df[1:150,]
test = df[151:nrow(df),]

# XGBoost Matrix
dtrain <- xgb.DMatrix(data = as.matrix(train[,-1]),label = as.matrix(train[,1]))
dtest <- xgb.DMatrix(data = as.matrix(test[,-1]),label = as.matrix(test[,1]))
watchlist <- list(eval = dtest)

# Custom objective function (squared error)
myobjective <- function(preds, dtrain) {
  labels <- getinfo(dtrain, "label")
  grad <- (preds - labels)
  hess <- rep(1, length(labels))
  return(list(grad = grad, hess = hess))
}

# Custom Metric
evalerror <- function(preds, dtrain) {
  labels <- getinfo(dtrain, "label")
  err <- (preds - labels)^2
  return(list(metric = "MyError", value = sqrt(mean(err))))
}

# Custom Model
param1 <- list(booster = 'gbtree', learning_rate = 0.1, objective = myobjective,
               eval_metric = evalerror, set.seed = 2020)

xgb1 <- xgb.train(params = param1, data = dtrain, nrounds = 500, watchlist,
                  maximize = FALSE, early_stopping_rounds = 5)
```

```
## [21:26:40] WARNING: amalgamation/../src/learner.cc:627:
## Parameters: { "set_seed" } might not be used.
##
##   This could be a false alarm, with some parameters getting used by language bindings but
##   then being mistakenly passed down to XGBoost core, or some parameter actually being used
##   but getting flagged wrongly here. Please open an issue if you find any such cases.
##
##
## [1]  eval-MyError:598.144739
## Will train until eval_MyError hasn't improved in 5 rounds.
##
## [2]  eval-MyError:562.479449
## [3]  eval-MyError:529.981101
## [4]  eval-MyError:501.730120
## [5]  eval-MyError:479.081301
## [6]  eval-MyError:459.354028
## [7]  eval-MyError:442.053856
## [8]  eval-MyError:428.946020
## [9]  eval-MyError:415.313647
## [10] eval-MyError:405.469398
## [11] eval-MyError:397.764096
## [12] eval-MyError:389.997507
## [13] eval-MyError:384.085178
## [14] eval-MyError:377.797138
## [15] eval-MyError:373.483600
## [16] eval-MyError:369.491675
## [17] eval-MyError:366.362771
## [18] eval-MyError:364.648653
## [19] eval-MyError:362.902533
## [20] eval-MyError:362.202943
## [21] eval-MyError:361.646832
## [22] eval-MyError:361.243759
## [23] eval-MyError:361.841428
## [24] eval-MyError:362.642165
## [25] eval-MyError:363.644628
## [26] eval-MyError:364.847153
## [27] eval-MyError:366.247771
## Stopping. Best iteration:
## [22] eval-MyError:361.243759
```

```r
pred1 = predict(xgb1, dtest)
mae1 = mae(test$Salary, pred1)

## Normal Model
param2 <- list(booster = 'gbtree', learning_rate = 0.1,
               objective = "reg:squarederror", set.seed = 2020)

xgb2 <- xgb.train(params = param2, data = dtrain, nrounds = 500, watchlist,
                  maximize = FALSE, early_stopping_rounds = 5)
```

```
## [21:26:40] WARNING: amalgamation/../src/learner.cc:627:
## Parameters: { "set_seed" } might not be used.
##
##   This could be a false alarm, with some parameters getting used by language bindings but
##   then being mistakenly passed down to XGBoost core, or some parameter actually being used
##   but getting flagged wrongly here. Please open an issue if you find any such cases.
##
##
## [1]  eval-rmse:598.144740
## Will train until eval_rmse hasn't improved in 5 rounds.
##
## [2]  eval-rmse:562.479436
## [3]  eval-rmse:529.981105
## [4]  eval-rmse:501.730122
## [5]  eval-rmse:479.081305
## [6]  eval-rmse:459.354033
## [7]  eval-rmse:442.053856
## [8]  eval-rmse:428.946021
## [9]  eval-rmse:415.313644
## [10] eval-rmse:405.469399
## [11] eval-rmse:397.764097
## [12] eval-rmse:389.997514
## [13] eval-rmse:384.085181
## [14] eval-rmse:377.797144
## [15] eval-rmse:373.483605
## [16] eval-rmse:369.491672
## [17] eval-rmse:366.362777
## [18] eval-rmse:364.648650
## [19] eval-rmse:362.902536
## [20] eval-rmse:362.202943
## [21] eval-rmse:361.646823
## [22] eval-rmse:361.243754
## [23] eval-rmse:361.841421
## [24] eval-rmse:362.546827
## [25] eval-rmse:362.973748
## [26] eval-rmse:363.185824
## [27] eval-rmse:363.490337
## Stopping. Best iteration:
## [22] eval-rmse:361.243754
```

```r
pred2 = predict(xgb2, dtest)
mae2 = mae(test$Salary, pred2)

# comparaison
print(list(xgb1$evaluation_log$eval_MyError[xgb1$best_iteration],
           xgb2$evaluation_log$eval_rmse[xgb2$best_iteration]))
```

```
## [[1]]
## [1] 361.2438
##
## [[2]]
## [1] 361.2438
```

```r
print(list(mae1, mae2))
```

```
## [[1]]
## [1] 203.3253
##
## [[2]]
## [1] 203.3253
```