Elastic Net e Mínimos Quadrados Parciais

Heitor Gabriel S. Monteiro

22/11/2021

Contents

1 Prelúdio

Nosso objetivo é exercitar UM algoritmo de redução de dimensão (**Mínimos Quadrados Parciais**) e outro de regularização e encolhimento de estimadores da regressão linear: o

(**Elastic Net**), que faz uma combinação linear entre a penalização Lasso ($\alpha = 1$) e Ridge ($\alpha = 0$) com o parâmetro `mixture`, na estrutura do Tidymodels. E λ sendo a importância da regularização, ou o grau de penalização, que é o parâmetro `penalty`. A estrutura do Elastic Net é:

$$\min_{\beta_0,\beta} \frac{1}{N} \sum_{i=1}^{N} w_i l(y_i, \beta_0 + \beta^T x_i) + \lambda \left[(1-\alpha)\|\beta\|_2^2/2 + \alpha\|\beta\|_1 \right],$$

Para importar os dados, vamos já retirar os `NA` usando `na.omit()`.

```
setwd('/home/heitor/Área de Trabalho/R Projects/Análise Macro/Labs/Lab 14')

library(tidyverse)    # padrão manipulação e visualização
library(tidymodels)   # padrão para ML
library(glmnet)       # Elastic Net
library(plsmod)       # Partial Least Squares
library(plotly)       # Gráficos interativos
library(GGally)       # Gráfico de Correlação
library(ISLR)         # Base de Dados Hitters
library(rmarkdown)    # Tabelas paginadas

set.seed(2022)        # Aleatoriedade fixa em número auspicioso

dds <- as.data.frame(Hitters) %>%
    as_tibble() %>% na.omit()
```

2 Estatísticas Descritivas

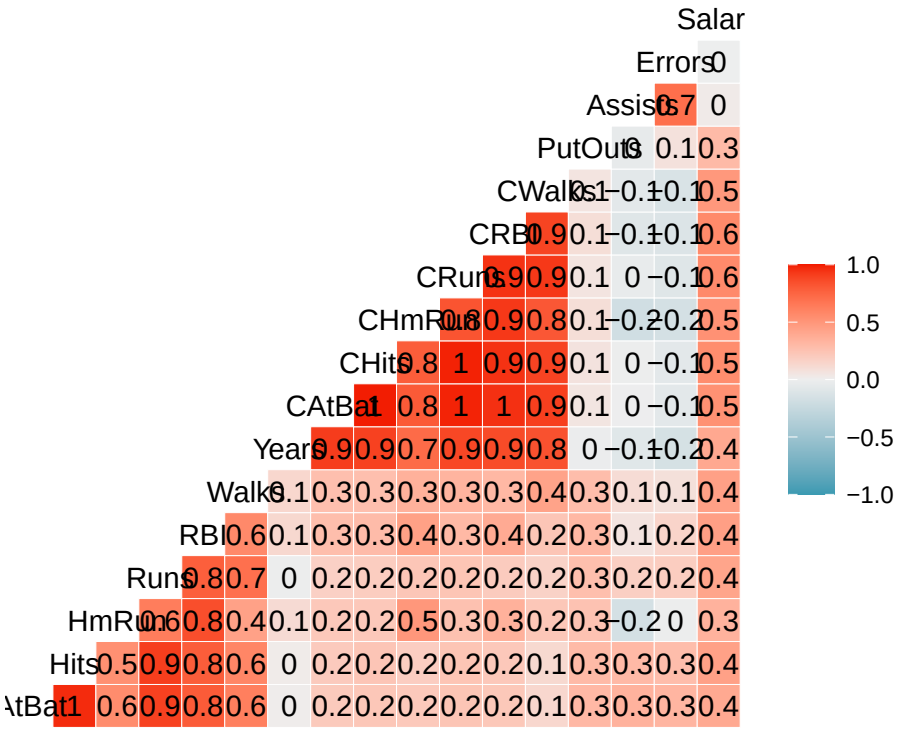
Reparemos que há muitas variáveis altamente correlacionadas e outras sem correlação. Para nossos fins de regredir `Salary`, realmente precisamos de um algoritmo de seleção de variáveis, ou redução delas.

```
dds %>% summary()
```

```
##      AtBat            Hits           HmRun            Runs
##  Min.   : 19.0   Min.   :  1.0   Min.   : 0.00   Min.   :  0.00
##  1st Qu.:282.5   1st Qu.: 71.5   1st Qu.: 5.00   1st Qu.: 33.50
##  Median :413.0   Median :103.0   Median : 9.00   Median : 52.00
##  Mean   :403.6   Mean   :107.8   Mean   :11.62   Mean   : 54.75
##  3rd Qu.:526.0   3rd Qu.:141.5   3rd Qu.:18.00   3rd Qu.: 73.00
##  Max.   :687.0   Max.   :238.0   Max.   :40.00   Max.   :130.00
##       RBI             Walks            Years            CAtBat
```

```
##  Min.   :  0.00   Min.   :  0.00   Min.   : 1.000   Min.   :   19.0
##  1st Qu.: 30.00   1st Qu.: 23.00   1st Qu.: 4.000   1st Qu.:  842.5
##  Median : 47.00   Median : 37.00   Median : 6.000   Median : 1931.0
##  Mean   : 51.49   Mean   : 41.11   Mean   : 7.312   Mean   : 2657.5
##  3rd Qu.: 71.00   3rd Qu.: 57.00   3rd Qu.:10.000   3rd Qu.: 3890.5
##  Max.   :121.00   Max.   :105.00   Max.   :24.000   Max.   :14053.0
##      CHits            CHmRun           CRuns             CRBI
##  Min.   :   4.0   Min.   :  0.00   Min.   :   2.0   Min.   :   3.0
##  1st Qu.: 212.0   1st Qu.: 15.00   1st Qu.: 105.5   1st Qu.:  95.0
##  Median : 516.0   Median : 40.00   Median : 250.0   Median : 230.0
##  Mean   : 722.2   Mean   : 69.24   Mean   : 361.2   Mean   : 330.4
##  3rd Qu.:1054.0   3rd Qu.: 92.50   3rd Qu.: 497.5   3rd Qu.: 424.5
##  Max.   :4256.0   Max.   :548.00   Max.   :2165.0   Max.   :1659.0
##      CWalks       League  Division    PutOuts          Assists
##  Min.   :   1.0   A:139   E:129    Min.   :   0.0   Min.   :  0.0
##  1st Qu.:  71.0   N:124   W:134    1st Qu.: 113.5   1st Qu.:  8.0
##  Median : 174.0                    Median : 224.0   Median : 45.0
##  Mean   : 260.3                    Mean   : 290.7   Mean   :118.8
##  3rd Qu.: 328.5                    3rd Qu.: 322.5   3rd Qu.:192.0
##  Max.   :1566.0                    Max.   :1377.0   Max.   :492.0
##      Errors           Salary       NewLeague
##  Min.   : 0.000   Min.   :  67.5   A:141
##  1st Qu.: 3.000   1st Qu.: 190.0   N:122
##  Median : 7.000   Median : 425.0
##  Mean   : 8.593   Mean   : 535.9
##  3rd Qu.:13.000   3rd Qu.: 750.0
##  Max.   :32.000   Max.   :2460.0
```

```
ggcorr(dds %>%
        select(!c('NewLeague', 'League', 'Division')),
       label = T)
```



3 Divisão: Treino & Teste

Vamos fazer a primeira divisão entre treino e teste. Aplicaremos a reamostragem somente nos dados de treino, conforme Kuhn & Johnson (2019).

```
slice_1   <- initial_split(dds)
train_dds <- training(slice_1)
test_dds  <- testing(slice_1)
```

4 Modelo

Formaremos agora dois modelos: o primeiro Elastic Net conforme a equação acima; o segundo será usando Mínimos Quadrados Parciais, com o número fixo de sete novos regressores, criados a partir da combinação linear dos regressores dos originais. Para detalhes, veja Lavine e Rayens (2019). Repare que os parâmetros de afinação `penalty` e `mixture` estão para ser testados no *cross validation*, para escolhermos o melhor.

```
algorit_elast <- linear_reg(
    penalty = tune::tune(),
    mixture = tune::tune()) %>%
    set_mode('regression') %>%
```

```
    set_engine("glmnet")
algorit_elast
```

```
## Linear Regression Model Specification (regression)
##
## Main Arguments:
##   penalty = tune::tune()
##   mixture = tune::tune()
##
## Computational engine: glmnet
```

```
algorit_pls <- pls(num_comp = 7) %>%
    set_mode("regression") %>%
    set_engine("mixOmics")
algorit_pls
```

```
## PLS Model Specification (regression)
##
## Main Arguments:
##   num_comp = 7
##
## Computational engine: mixOmics
```

5 Fórmula

Para preparar os dados para ser aplicado no modelo, aplicaremos o `step_normalize(all_numeric_pred` para normalizar todas as variáveis numéricas e `step_dummy(all_nominal_predictors())` para tranformar variáveis fator em *dummies*. A função `bake()` "cozinha" e nos mostra como ficarão os dados.

```
formula_geral <- recipe(Salary~.,
                        data = train_dds) %>%
    step_normalize(all_numeric_predictors()) %>%
    step_dummy(all_nominal_predictors()) %>%
    prep()

formula_geral %>%
    bake(new_data=NULL) %>%
    rmarkdown::paged_table(options = list(
        rows.print = 10,
        cols.print = 10))
```

6 Workflow

Definiremos dois procedimentos por termos dois modelos diferentes. Nele, vamos alimentar a estrutura dos modelos com a fórmula transformada. Repare que, como não há nada para ser *tunado* na nossa regressão usando *pls*, já aplico a divisão final `last_fit(slice_1)`. Retomaremos a `wrkflw_pls` depois que afinarmos o Elastic Net.

```
wrkflw_elast <- workflow() %>%
    add_model(algorit_elast) %>%
    add_recipe(formula_geral)

wrkflw_pls <- workflow() %>%
    add_model(algorit_pls) %>%
    add_recipe(formula_geral)

wrkflw_pls <- wrkflw_pls %>%
    last_fit(slice_1)
```

7 Validação

Definiremos o método de reamostragem para validação cruzada:

```
kfold_geral <- vfold_cv(train_dds,
                        v=10,
                        repeats = 2)
```

8 Ajustes e Treinamentos

8.1 Limites do `tune()`

Montaremos pares de parâmetros a serem afinados. Veja que `mixture` vai de 0 (Ridge) a 1 (Lasso):

```
grid_padr <- wrkflw_elast %>%
    parameters() %>%
    update(
        penalty = penalty(range = c(.25, .75)),
        mixture = mixture(range = c(0, 1))
    ) %>%
    grid_regular(levels = 5)
```

```
grid_padr %>%
    rmarkdown::paged_table(options = list(
        rows.print = 10,
        cols.print = 2))
```

8.2 Afinação dos `tune()`

Apesar de montar dois parâmetros de métrica, usaremos o `rmse`, que é mais sensível por pesar mais outliers.

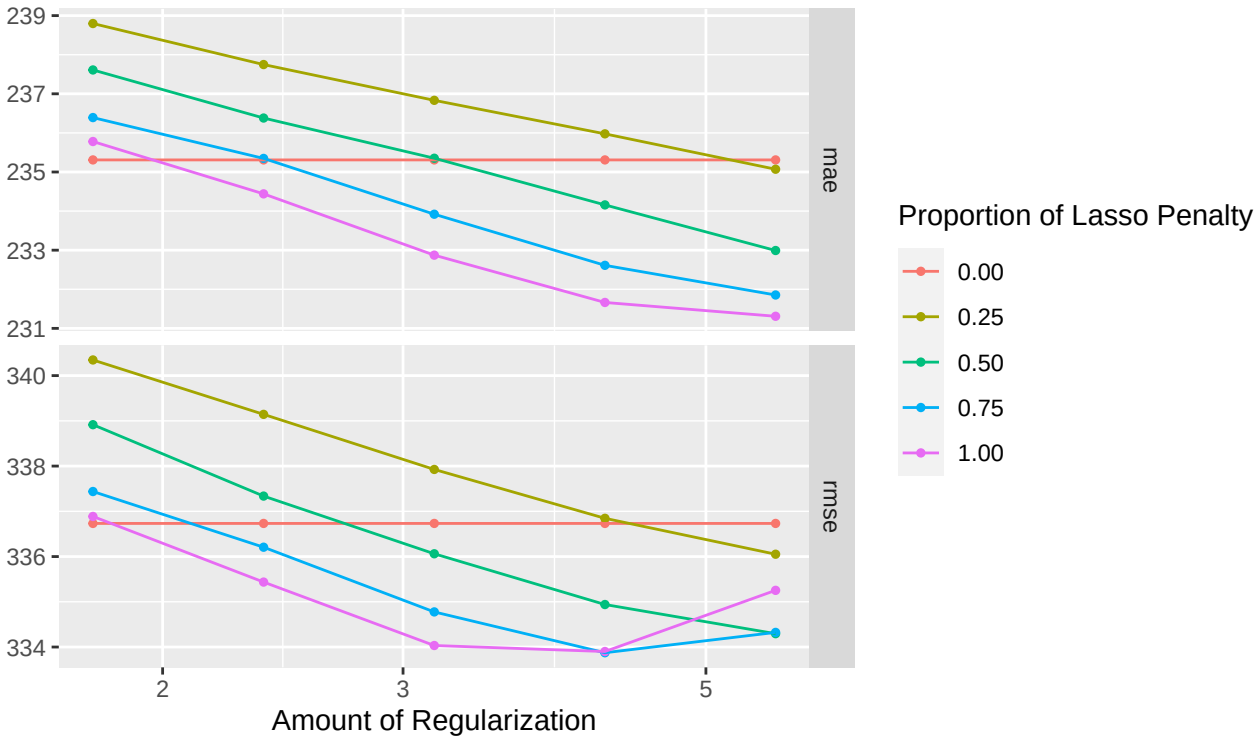
```
afinç_cv_elast <- wrkflw_elast %>%
    tune_grid(resamples = kfold_geral,
              grid      = grid_padr,
              control   = control_grid(save_pred = T),
              metrics   = metric_set(rmse, mae))

wrkflw_pls$.metrics
```

```
## [[1]]
## # A tibble: 2 x 4
##   .metric .estimator .estimate .config
##   <chr>   <chr>          <dbl> <chr>
## 1 rmse    standard     314.    Preprocessor1_Model1
## 2 rsq     standard       0.537 Preprocessor1_Model1
```

9 Seleção do melhor modelo

Reparemos que interessante: o Ridge (`mixture = 0`) sai de melhor e se torna o pior à medida que o peso dos preditores adicionais cresce.

```
afinç_cv_elast %>% ggplot2::autoplot()
```



Ainda sim, pelo `show_best(n=1, 'rmse')`, vemos que algoritmo dos mínimos quadrados parciais performa melhor que a da regularização e penalização:

```
afinç_cv_elast %>% show_best(n=1, 'rmse')
```

```
## # A tibble: 1 x 8
##   penalty mixture .metric .estimator  mean     n std_err .config
##     <dbl>   <dbl> <chr>   <chr>      <dbl> <int>   <dbl> <chr>
## 1    4.22    0.75 rmse    standard    334.    20    24.6 Preprocessor1_Model19
```

```
wrkflw_pls %>% show_best(n=1, 'rmse')
```

```
## # A tibble: 1 x 7
##   .workflow  .metric .estimator  mean     n std_err .config
##   <list>     <chr>   <chr>      <dbl> <int>   <dbl> <chr>
## 1 <workflow> rmse    standard    314.     1      NA Preprocessor1_Model1
```

```
melhor_tune <- select_best(afinç_cv_elast, 'rmse')
```

9.1 Finalmentes

Agora, extrairemos o melhor par de parâmetros do Elastic Net, montaremos o procedimento do `workflow()` e aplicaremos na amostra de teste. Lembremos que já fazemos isso para o *pls*, já que não havia nada a ser afinado.

```
algorit_fnl_elast <- algorit_elast %>%
    finalize_model(parameters = melhor_tune)

wrkflw_fnl_elast <- workflow() %>%
    add_model(algorit_fnl_elast) %>%
    add_recipe(formula_geral) %>%
    last_fit(slice_1)

wrkflw_fnl_elast$.predictions
```

```
## [[1]]
## # A tibble: 66 x 4
##    .pred  .row Salary .config
##    <dbl> <int>  <dbl> <chr>
##  1  748.     2   480  Preprocessor1_Model1
##  2  699.     9  1100  Preprocessor1_Model1
##  3  861.    10   517. Preprocessor1_Model1
##  4 1220.    21   777. Preprocessor1_Model1
##  5  701.    25   625  Preprocessor1_Model1
##  6  216.    27   110  Preprocessor1_Model1
##  7  727.    30   850  Preprocessor1_Model1
##  8  216.    36   248. Preprocessor1_Model1
##  9  742.    41   675  Preprocessor1_Model1
## 10  298.    43   340  Preprocessor1_Model1
## # ... with 56 more rows
```

Agora, para representação, faremos os dois plots das duas técnicas, comparando estimados e reais:

```
#ggplotly(
wrkflw_fnl_elast %>%
    collect_predictions() %>%
    ggplot(aes(x=.pred, y=Salary)) +
    geom_point() +
    geom_abline(intercept = 0,
                slope     = 1,
                color     = 'darkviolet',
                size      =.8)
```

```
#ggplotly(
    wrkflw_pls %>%
        collect_predictions() %>%
```

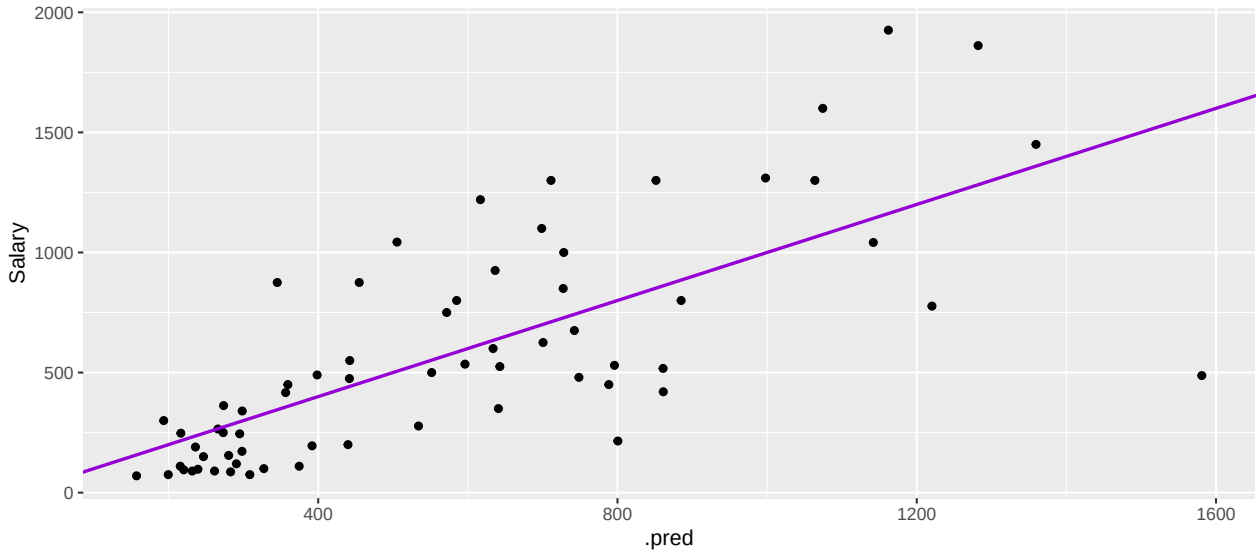


Figure 1: Modelo Elastic Net

```
ggplot(aes(x=.pred, y=Salary)) +
geom_point() +
geom_abline(intercept = 0,
            slope     = 1,
            color     = 'yellow3',
            size      =.8)
```

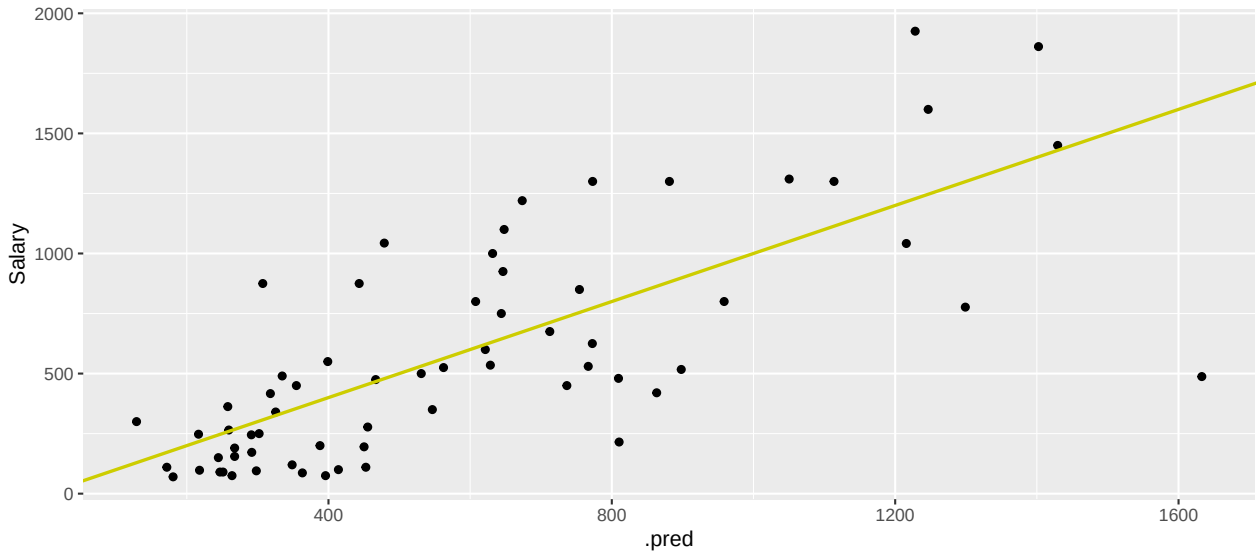


Figure 2: Modelo de Mínimos Quadrados Parciais