Uso pacote massive

Alcides C. Neto & Yasmin R. Fernandes

23 de março de 2016

Primeiramente será instalado e requisitado o pacote massive, isso é feito como segue-se abaixo:

Instalação

```
## Instalando o devtools para efetuar a instalação de pacotes no git -----------

install.packages("devtools")

## Instalando pacote massive ---------------------------------------------------

devtools::install_git("https://gitlab.c3sl.ufpr.br/di12/massive.git")
```

Requisição

```
## Requisitando o pacote massive -----------------------------------------------

library(massive)
library(xtable)

## Definir opção do xtable -----------------------------------------------------

options(xtable.comment = FALSE)
```

Leitura e seleção dos dados

```
## Requisitando banco de dados -------------------------------------------------

data(wine)

## Selecionar parte dos dados para análise -------------------------------------

dados <- wine[1:500,]
```

Aplicando função massive

A idéia é testar todas as combinações de modelos possíveis com as variáveis explicativas disponíveis na base.

A função massive pega uma subamostra do conjunto de dados como teste, na qual será ajustado os modelos. A outra parte da base fica como treino, ou seja, são aplicados os modelos já ajustados, obtidos pela base de teste, aos dados de treino.

O retorno da função é uma lista com 4 itens obtidos atravéz da base teste, sendo eles:

- **coef**: lista dos coeficiêntes de cada modelo.
- **model**: lista as variáveis que entram em cada modelo.
- **aic**: lista dos coeficiêntes AIC de cada modelo.
- **pacertos**: porcentagem de modelos com diferença entre teste e treino menor que 0.5.

```
## Aplicando para ambos grr ------------------------------------------------------

resultado1 <- massive(dados, seed = 20137525)
resultado2 <- massive(dados, seed = 20137523)
```

```
## Obtendo data.frame de comparação ----------------------------------------------

compara1 <- data.frame("AIC" = resultado1$aic, "Proporção" = resultado1$pacertos)

xtable(head(compara1))
```

	AIC	Proporção
1	99.10	45.00
2	119.46	39.00
3	99.46	45.00
4	120.54	39.00
5	100.80	44.00
6	120.44	38.00

```
compara2 <- data.frame("AIC" = resultado2$aic, "Proporção" = resultado2$pacertos)

xtable(head(compara2))
```

	AIC	Proporção
1	124.77	45.00
2	135.15	39.00
3	126.15	45.00
4	133.13	41.00
5	125.98	45.00
6	134.98	40.00

```
## Menor AIC para o compara1 -----------------------------------------------------

linhaAIC1 <- compara1[,1] == min(compara1[,1]) # Linha melhor AIC.
xtable(compara1[linhaAIC1, ])
```

	AIC	Proporção
1	99.10	45.00

```
linhaProp1 <- compara1[,2] == max(compara1[,2])
xtable(compara1[linhaProp1, ])
```

	AIC	Proporção
1	99.10	45.00
3	99.46	45.00
19	100.41	45.00
23	102.32	45.00

```
## Menor AIC para o compara2 ------------------------------------------------

linhaAIC2 <- compara2[,1] == min(compara2[,1]) # Linha melhor AIC.
xtable(compara2[linhaAIC2, ])
```

	AIC	Proporção
17	117.62	40.00

```
linhaProp2 <- compara2[,2] == max(compara2[,2])
xtable(compara2[linhaProp2, ])
```

	AIC	Proporção
37	127.70	46.00

Guardando tabelas de comparação

```
write.table(compara1, file = "grr20137525.txt", row.names = FALSE, sep = "\t")
write.table(compara2, file = "grr20137523.txt", row.names = FALSE, sep = "\t")
```

Função massive comentada

```
massive <- function(base, preditor = 'ultimo', percent = 10, seed = 489850){
    # Define semente:
    set.seed(seed)
    # Se o preditor é diferente de "ultimo":
    if(preditor != 'ultimo'){
        # Se o número de variáveis é diferente do preditor:
        if(ncol(base) != preditor){
            # Realica o preditor na ultima coluna:
            base[ncol(base) + 1] <- base[,preditor]
            # Exclui o preditor repetido (diferente ultimo):
            base <- base[,-preditor]
      }
    }
    # Renomeia coluna do preditor:
    colnames(base)[ncol(base)] <- 'predict'
    # Número de linhas da base completa:
    total <- nrow(base)
    # Trunca a porcentagem para n inteiro:
    totaltest <- floor(total * (percent / 100))
```

```r
    # Seleciona indice elementos teste:
    testrow <- sample(1:total, totaltest)
    # Constroi matriz de teste:
    test <- base[testrow,]
    # Controi matriz de treino:
    train <- base[-testrow,]
    # Número de variáveis explicativas no teste:
    var <- ncol(test) - 1
    # Número de modelos possíveis:
    binvar <- 50 # 2^var - 1
    # Definindo vetor de coeficientes:
    coef <- vector("list", 4)
    # Renomeando vetor de coeficientes:
    names(coef) <- c("coef", "model", "aic", "pacertos")
    # Inicializando vetor de coeficientes:
    coef$coef <- coef$model <- coef$aic <- coef$pacertos <- c(rep(0, binvar))
    for(i in 1:binvar){ # Para cada modelo possível.
        # Variáveis que entram no modelo.
        b <- c(rev(as.numeric(intToBits(i))[1:var]) == 1, TRUE)
        # Ajusta o modelo com as variáveis selecionadas por b:
        rlm <- lm(predict ~. ,data= test[,c(names(test)[c(b)])])
        coef$coef[i] <- list(rlm$coefficients) # Coeficiente dos modelos.
        coef$model[i] <- list(b) # Variáveis que entram no modelo.
        coef$aic[i] <- AIC(rlm) # Cálculo do AIC.
        # train.p recebe a treino selecionada pelas variáveis do modelo:
        train.p <- train[,unlist(coef$model[i])]
        # Inicializando variável preditor:
        preditor <- c(rep(0,dim(train.p)[1]))
        beta <- rlm$coefficients # Guarda os betas.
        # j de 1 até quantidade observações do train.p:
        for(j in 1:dim(train.p)[1]){
            # Calculando predict com vetor train.p:
            preditor[j] <- beta[1] + sum(train.p[j,-dim(train.p)[2]] * beta[-1])
        }
        # data base com preditores teste e treino.
        p.df <- data.frame(cbind(train.p[dim(train.p)[2]],preditor))
        # Modulo da diferença entre preditores teste e treino.
        res <- abs(p.df[,1] - p.df[,2])
        #  Soma as diferenças menores que 0.5:
        contagem <- sum(res < 0.5)
        # Calcula Proporção de acertos:
        coef$pacertos[i] <- round(contagem / total, digits = 2) * 100
    }
    return(coef = coef) # retorna a matriz coef.
}
```