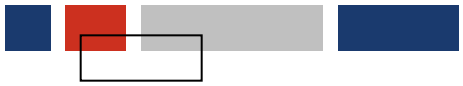
INSTITUTO NACIONAL DE ESTATÍSTICA
STATISTICS PORTUGAL

» Introdução à Ciência dos Dados

Pedro Campos
Conceição Ferreira
João Lopes

DMSI / ME



12 de fevereiro de 2019

Índice

- Modelos
- Modelos preditivos
- Redes Neuronais Artificiais
- Algoritmos Genéticos
- Exemplo práctico

Modelos

» Modelos exploratórios

- modelos não-supervisionados (i.e. não têm variável-resposta);
- criam associações e agrupamentos.

Modelos

» Modelos preditivos

- modelos supervisionados (têm variável-resposta);
- criam relações entre variáveis que permitem fazer previsões:

    variável-resposta qualitativa: problema de classificação

    variável-resposta quantitativa: problema de regressão

Modelos

» Modelos exploratórios

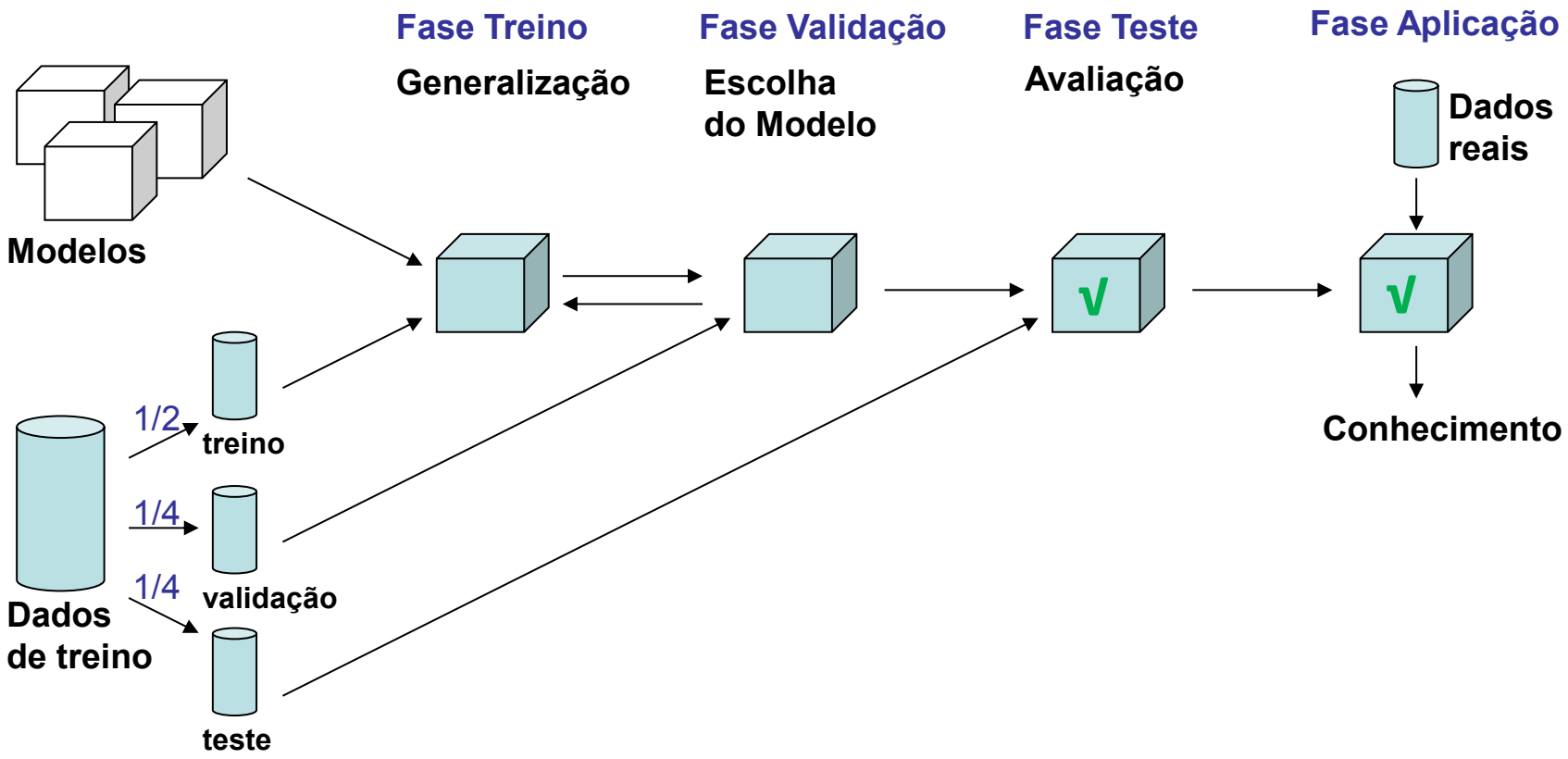
- Regras de Associação
- Agrupamento (*Clustering*)

» Modelos preditivos

- Procura (e.g. Árvores de Decisão)
- Distâncias (e.g. *k-Nearest Neighbour*)
- Probabilísticos (e.g. Redes *Bayesianas*)
- Otimização (e.g. Redes Neuronais, *Support Vector Machine*)

Modelos preditivos

» Visão Geral



Modelos preditivos

» Conceitos

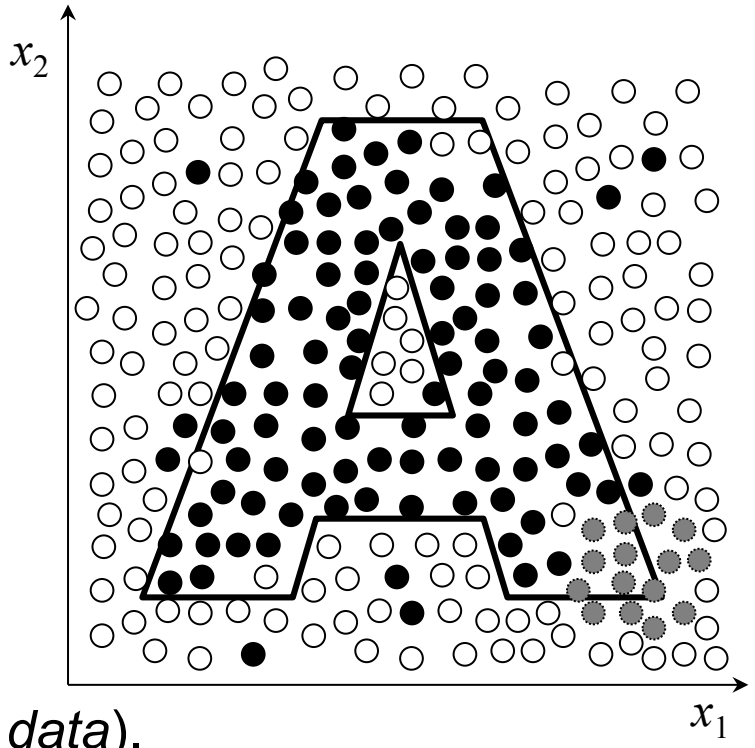
- variável-resposta e variáveis-preditivas;
- problemas no conjunto de treino (e.g. ruído, dados em falta, …);
- *curse of dimensionality*;
- sub-ajustamento vs. sobre-ajustamento;
- critérios de paragem de um algoritmo;
- máximos locais e máximo global.

Modelos preditivos

» Conceitos: conjunto de treino

e.g.

- variável-resposta (qualitativa)
- 2 variáveis-preditivas (x_1 e x_2)



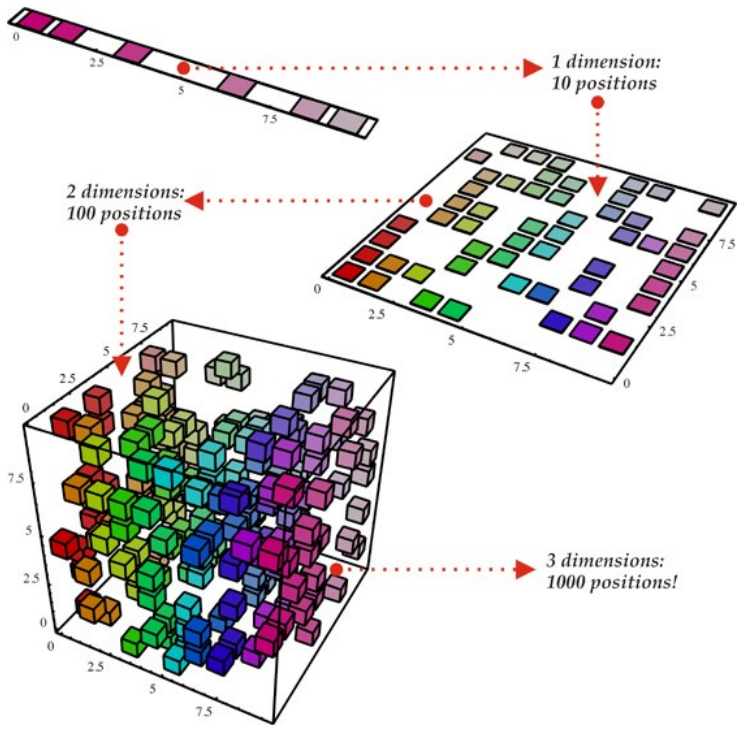
- ruído nos dados (*noisy data*);
- dados em falta (*missing data*);
- exemplos fronteiriços (*borderline*);
- dados desequilibrados (*imbalanced data*).

Modelos preditivos

» Conceitos: *curse of dimensionality*

+ variáveis => + informação

=> + espaço a explorar



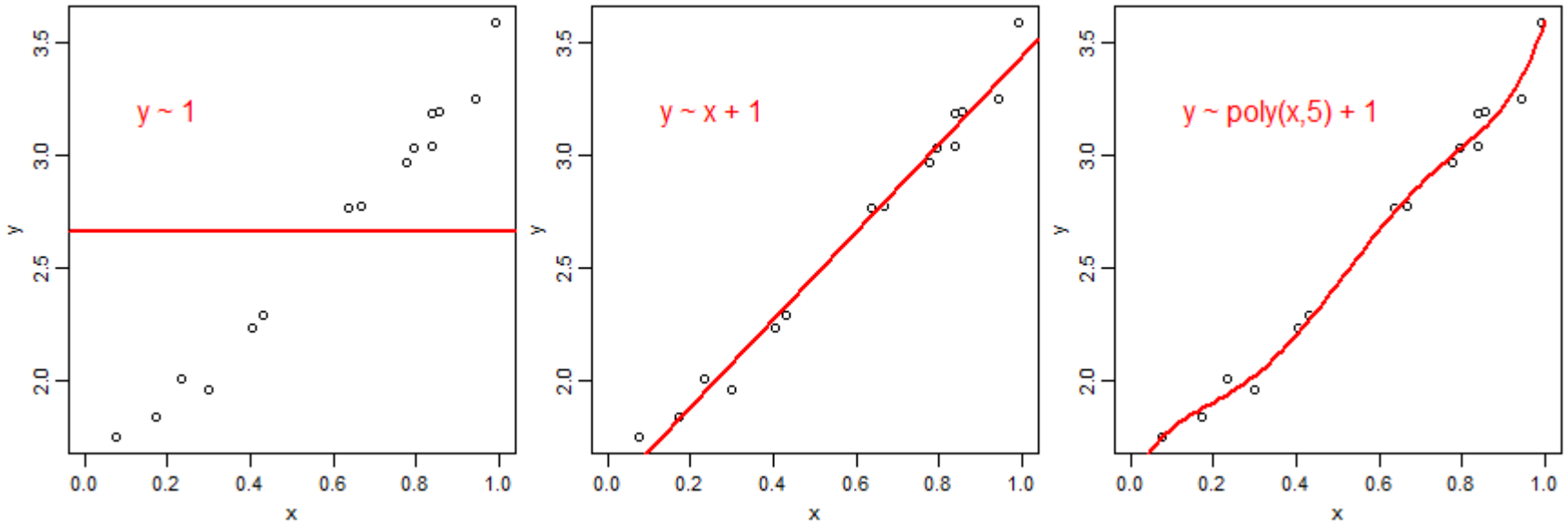
Y. Bengio (2008) “CurseDimensionality.jpg”

Modelos preditivos

» Conceitos: sub- e sobre-ajustamento

- sub-ajustamento: extrai pouco conhecimento;

- sobre-ajustamento: não constrói generalizações.



Modelos preditivos

» Conceitos: critérios de paragem

- Modelos preditivos são tipicamente iterativos;
- Cada iteração ajusta o modelo ao conjunto de treino;
- Critérios de paragem do ajustamento:

    i) número de iterações máximo atingido;

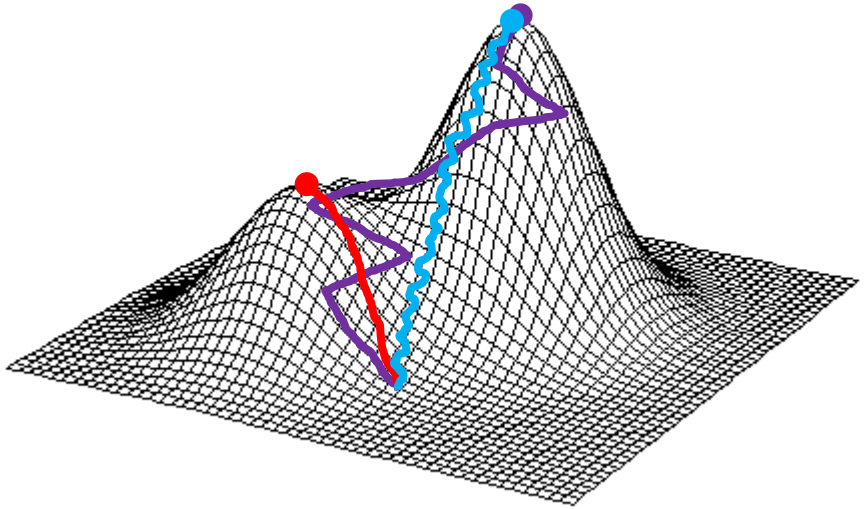
    ii) erro reduz-se abaixo de limiar de erro;

    iii) modelo não é alterado.

Modelos preditivos

» Conceitos: máximos locais e globais

- *hill climbing* (algoritmo *greedy*);
- *stochastic hill climbing;*
- *simulated annealing*.



adaptado de Headlessplatter (2007) “Local_maximum.png”

Redes Neuronais Artificiais

» Redes Neuronais Artificiais

- Algoritmos de optimização inspirados nas redes neuronais do cérebro, onde redes densas de neurónios realizam aprendizagens complexas;

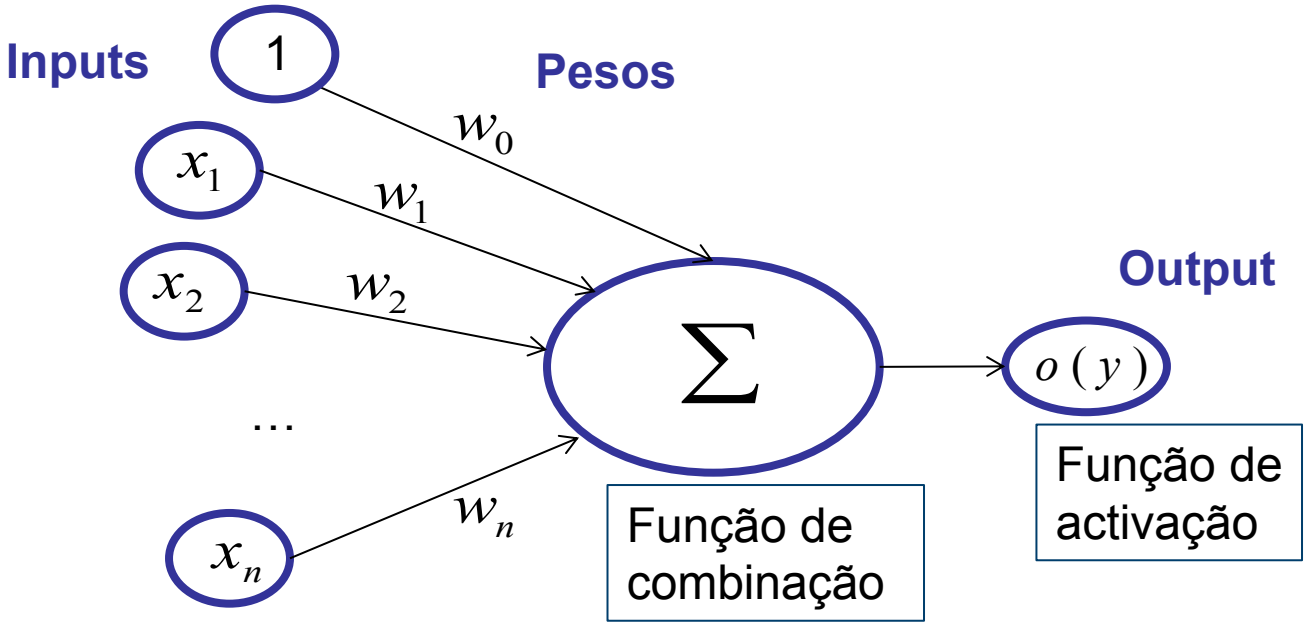
- Os neurónios recebem um input de estímulos que se combinam na função de combinação e produzem, através da função de activação, um output de resposta;

- Modelos simples: um neurónio (i.e. *single-layer perceptron*);

- Modelos complexos: multi-camadas (i.e. *multi-layer perceptron*).

Redes Neuronais Artificiais

» *Single-layer perceptron*



Redes Neuronais Artificiais

» *Single-layer perceptron*

Função de combinação

$$y = w_0 + w_1 x_1 + w_2 x_2 + \ldots + w_n x_n$$

$$= \sum_{i=0}^{n} w_i x_i$$

Função de activação

$$o(y) = \begin{cases} 1 \ sse \ \ y > 0 \\ 0 \ \ c.c. \end{cases}$$

$$o(y) = \frac{1}{1 + e^{-y}}$$

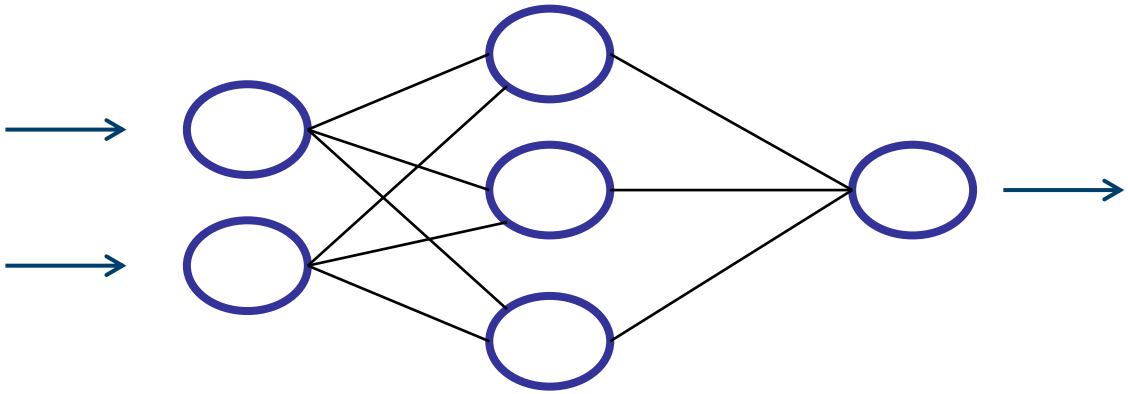
$$o(y) = y$$

- Classificador Linear
- Regressão Múltipla Linear

Redes Neuronais Artificiais

» *Multi-layer Perceptron*

- Camadas dispostas paralelamente: camada de entrada, camadas intermédias (i.e. camadas escondidas) e camada de saída.



Redes Neuronais Artificiais

» Vantagens e Desvantagens

- Grande capacidade de generalizar, e tolerante a ruído e a dados em falta;

- Ajusta modelos complexos sem tratamento analítico.

- Parâmetros do modelo de difícil interpretação (*blackbox*);

- Resultados dependem da arquitectura da rede (i.e. pouco robustos);

- Número de neurónios pode levar a sobre-ajustamento do modelo.

Redes Neuronais Artificiais

» No **R**

```
library("nnet")
library("caret")
library("NeuralNetTools")

trainset <- read.csv("../data/ripley-set.csv",header=TRUE)
trainset[,"label"] <- factor(trainset[,"label"],levels=c("0","1"))

#Create testing set by hold out
n <- nrow(trainset)
n_test <- floor(n/3)
sampindex <- sample(1:n,size=n_test,replace=FALSE)
testset <- trainset[sampindex,]
trainset <- trainset[-sampindex,]
```

Redes Neuronais Artificiais

```
#Select settings for artificial neural network
nn_decay <- 0.3                          #weight decay
nn_maxit <- 500                          #maximum number of iterations
grid1 <- expand.grid(size=1:5,decay=nn_decay)
train1 <- train(label ~.,                #model design
                trainset,                #training set
                maxit=nn_maxit,          #maximum number of iterations
                tuneGrid=grid1,          #parameter space to explore
                metric="Accuracy",       #metric to evaluate models
                method="nnet",trace=FALSE)
nn_size <- train1$bestTune[1,1]          #number of nodes in hidden layer

#Perform Neural Network fit
mod <- nnet(label ~.,                    #model design
            trainset,                    #training set
            size=nn_size,decay=nn_decay,maxit=nn_maxit)
print(mod); summary(mod)
```

Redes Neuronais Artificiais

```
#Plot model
plotnet(mod,cex_val=0.7,circle_cex=3)
olden(mod,bar_plot=TRUE)

#Test model
res <- predict(mod,                    #model
               testset,                #testing set
               type="class")           #can be "raw" or "class"

#Confusion Matrix
conf_mat <- table(testset[,c("label")],res)
conf_mat

#Error rate
err_rt <- 100*(sum(conf_mat) - sum(diag(conf_mat)))/sum(conf_mat)
err_rt
```

Redes Neuronais Artificiais

E\O	0	1
0	31	8
1	5	39

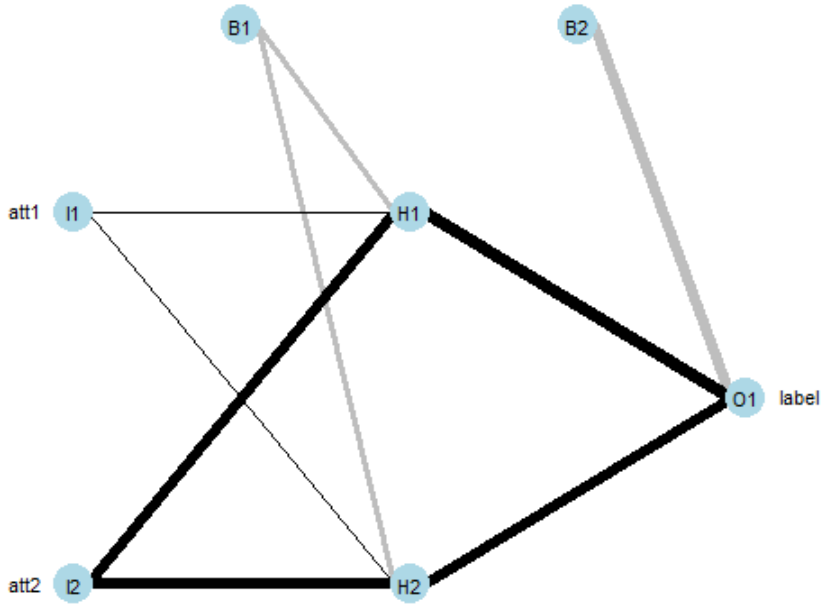
n = 250

n_train = 167

n_test = 83

error rate = 15.7%

labels	Importance (Olden, 2004)
attr1	3.586
attr2	25.451



Algoritmos Genéticos

» Enquadramento

- Algoritmos de optimização inspirados na evolução natural de sistemas biológicos (popularizados por John Holland nos anos 70);

- Utilizam população de soluções (i.e. agentes), cada uma constituída por sequências de variáveis (i.e. cromossomas). Cada solução tem valores diferentes para cada variável (i.e. genes);

- As populações de soluções são avaliadas por função de *fitness*, e geram novas gerações de soluções;

- A população evolui ao longo das gerações através de mecanismos evolutivos (seleção, recombinação, mutação, …).

Algoritmos Genéticos

» Mecanismos evolutivos

Seleção: as “melhores” soluções passam para a geração seguinte;

Recombinação: troca de informação entre soluções;

Mutação: a transferência de informação entre gerações é imperfeita;

outros (e.g. Migração): entrada de soluções novas na população.

Algoritmos Genéticos

» Vantagens e Desvantagens

- Algoritmo é facilmente interpretável;

- Processo de optimização robusto a ruído e a máximos locais;

- Facilmente paralelizável, (i.e. redução de tempo de computação).

- Rapidez de convergência depende da escolha dos parâmetros do modelo e dos critérios de paragem;

- A função de *fitness* requer tratamento analítico;

- Número de variáveis utilizado influencia grandemente a intensidade computacional (i.e. *curse of dimensionality*).

Algoritmos Genéticos

» Exemplo*

- Um explorador vai para a selva;
- Mochila com capacidade de 20Kg;
- Conjunto de itens caracterizados por pontos de sobrevivência e peso;
- Qual o melhor conjunto de itens?

ITEM	SURV. PTS.	WEIGHT
pocketknife	10.00	1.00
beans	20.00	5.00
potatoes	15.00	10.00
unions	2.00	1.00
sleeping bag	30.00	7.00
rope	10.00	5.00
compass	30.00	1.00

**Knapsack problem* é um problema de otimização clássico do século XIX. Adaptado de Marek Obitko. http://www.r-bloggers.com/genetic-algorithms-a-simple-r-example/

Algoritmos Genéticos

» Exemplo*

- O número possível de instâncias é $2^7=128$;
- Maximizar pontos de sobrevivência, mas mantendo peso < 20Kg.

Exemplos de instâncias:

pkt-knife	beans	potatoes	unions	sleep bag	rope	compass	Fitness
0	1	0	1	1	0	0	52
1	0	0	0	1	1	1	80
0	0	1	0	1	1	0	0

**Knapsack problem* é um problema de otimização clássico do século XIX. Adaptado de Marek Obitko. http://www.r-bloggers.com/genetic-algorithms-a-simple-r-example/

Algoritmos Genéticos

» No **R**

```
library("GA")

#Define data
n <- c("pocketknife","beans","potatoes","unions","sleepingbag","rope","compass")
p <- c(10,20,15,2,30,10,30)            #profits
w <- c(1,5,10,1,7,5,1)                 #weights
W <- 20                                #knapsack capacity

#Define fitness function
knapsack <- function(x){
    if(sum(x*w) > W)
        return(0)
    else
          return(sum(x*p))
}
```

Algoritmos Genéticos

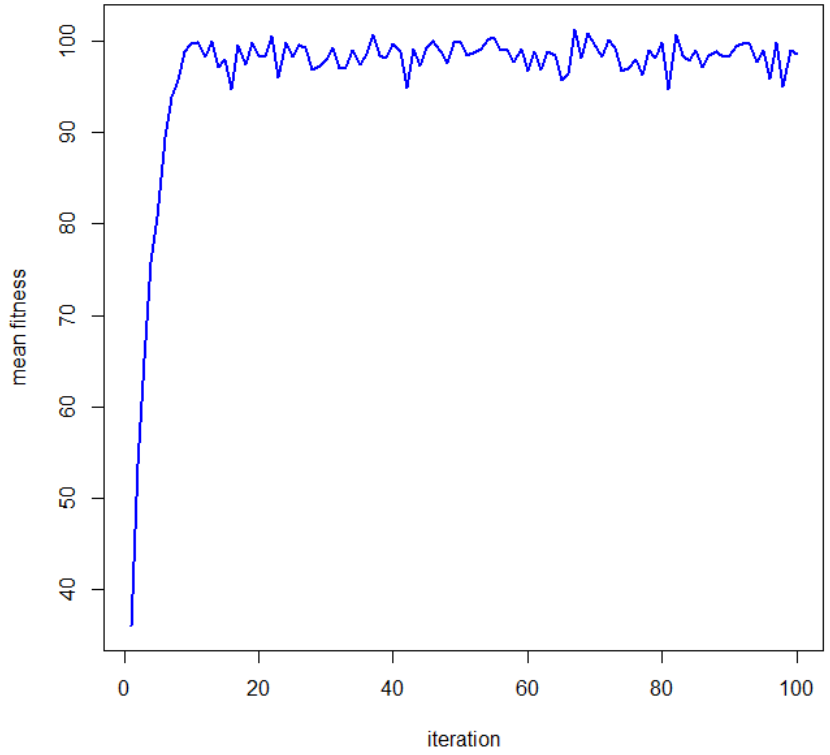
```
#Run SGA
SGA <- ga(type="binary",
fitness=knapsack,                    #fitness function
  nBits=length(n),                   #chromosome length
  popSize=100,                       #population size
  pcrossover=0.8,                    #crossover rate
  pmutation=0.1,                     #mutation rate
  elitism=5,                         #number of best individuals sure to be selected
  maxiter=100,                       #number of generations
  names=n,                           #name of "genes"
  seed=101)

res <- SGA@solution
print(res)                           #best solution
sum(res)                             #total number of selected items
sum(res*p)                           #total profit of selected items
sum(res*w)                           #total weight of selected items
```

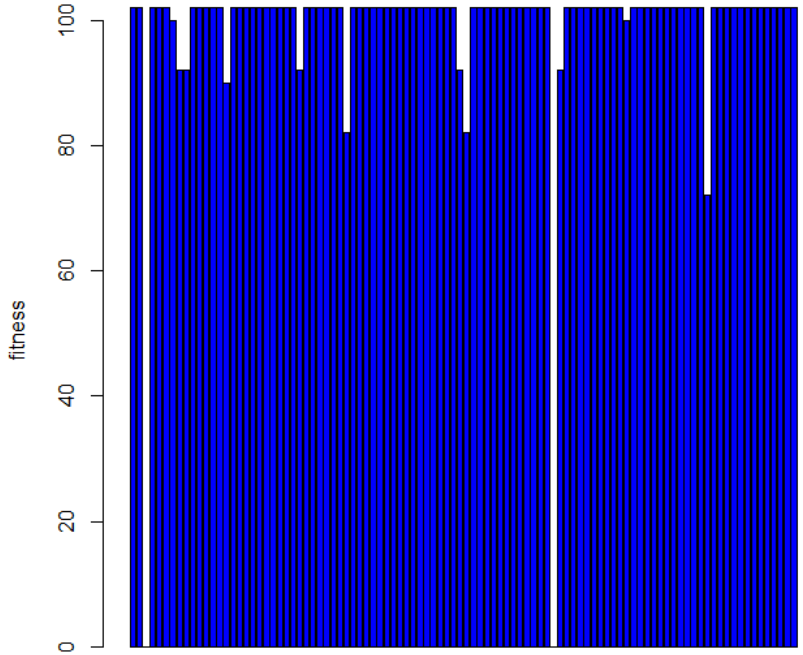
Algoritmos Genéticos

```
#Plot model
plot(SGA@summary[,"mean"],type="l",ylab="mean fitness",xlab="iteration",lwd=2)
barplot(SGA@fitness,ylab="fitness",xlab="",col="blue")
```

Algoritmos Genéticos

pkt-knife	beans	potatoes	unions	sleep-bag	rope	compass	Fitness
1	1	0	1	1	1	1	102





Exemplo práctico

» EU BD Hackathon 2017

- **Objectivo:** ajudar a definir políticas sobre o mercado laboral
- **Organização:** Eurostat and CEDEFOP
- **Projecto:**

  - Caracterização do Mercado Laboral.

  - Estabelecer associações entre indicadores relevantes (e.g. Mobilidade Laboral na UE) e características do mercado laboral.

https://github.com/jsollari/EUhackathon2017

https://ec.europa.eu/eurostat/web/products-statistical-working-papers/-/KS-TC-18-002

Exemplo práctico

» Dados: caracaterísticas do Mercado Laboral

- “reg_dem” – informação demográfica
- “earn” – estrutura de ganhos
- “educ_uoe_fin” – gastos públicos em educação
- “ilc” – rendimento e condições de vida
- “employ” – informação sobre o emprego
- “nama10” – contas nacionais
- “educ_part” – informação sobre educação

7 datasets, **17** main variables

Exemplo práctico

» Dados: caracaterísticas do Mercado Laboral

- "reg_dem" by **age** (**NUTS2**)
- "earn" by **occupation** and **economic activity**
- "educ_uoe_fin"
- "ilc" (**NUTS2**)
- "employ" by **age**, **education level**, **economic activity** (**NUTS2**)
- "nama10" (**NUTS2**)
- "educ_part" (**NUTS2**)

7 datasets, **17** main variables, **76** variables

Exemplo práctico

» Dados: Mobilidade Laboral na UE

- "lfso_14leeow" – informação sobre a força laboral

1 dataset, **1** main variable

subjects: **25 (NUTS0)**

Exemplo práctico

» No **R**

```
#1. FUNCTIONS
source("misc_v2.3.r")

#2. READ DATA
f1 <- "../data/lmktattract.csv"
f2 <- "../data/lmktmobil.csv"
x_all <- read.table(file=f1,header=TRUE,sep=",",dec=".",row.names=1)
y_all <- read.table(file=f2,header=TRUE,sep=",",dec=".",row.names=1)

# 3. ANALYSE DATA
#remove attributes with missing data
ina <- apply(x_all,2,function(x){any(is.na(x))})
x <- x_all[,!ina]
```

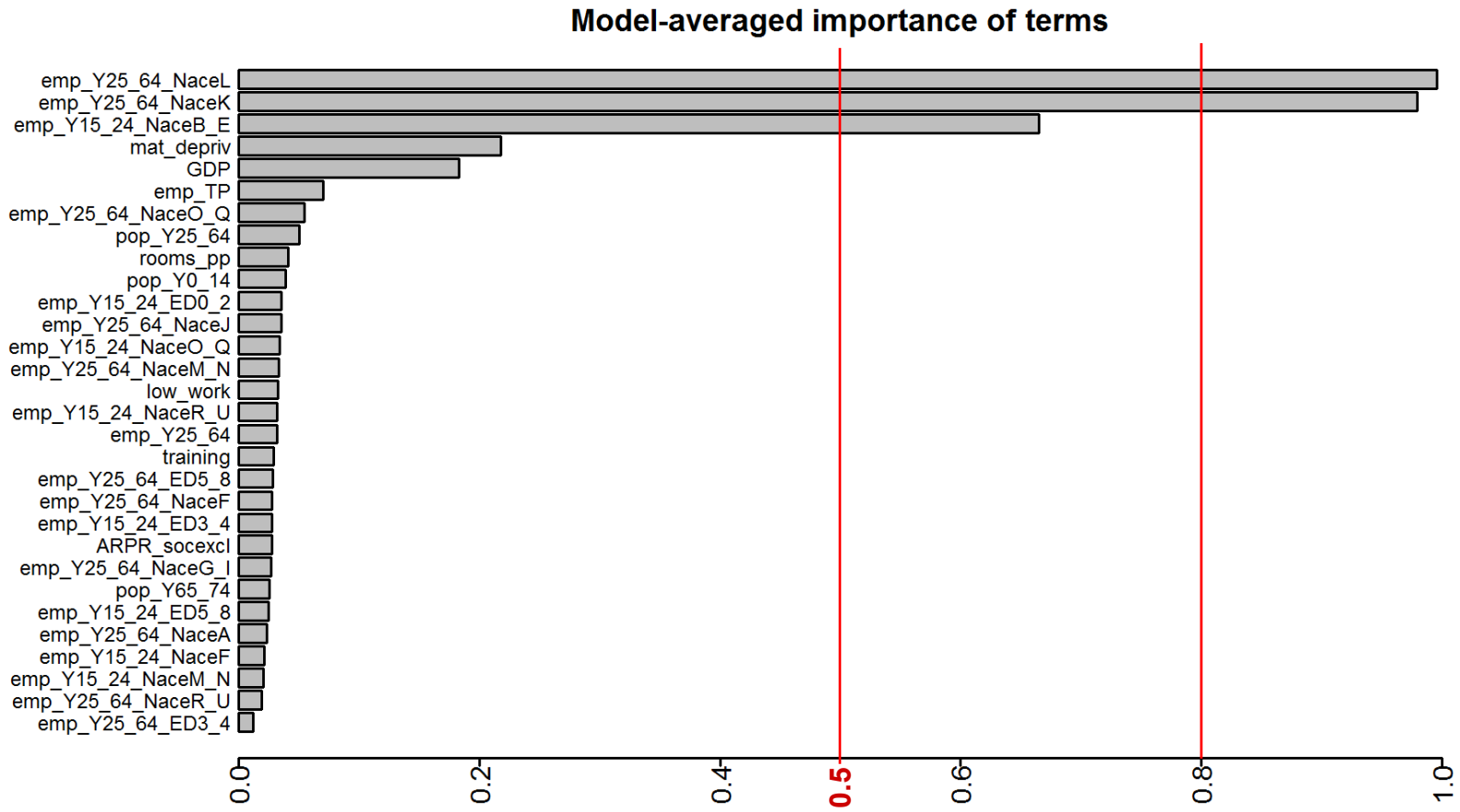
Exemplo práctico

```
#remove data entries with missing data
y <- y_all[,"lmktm_Total",drop=FALSE]
ina <- is.na(y[,1])
x <- x[!ina,]
y <- y[!ina,,drop=FALSE]

#reduce number of attributes
f1 <- "../results/lmktattract_2/datred.log"
x <- reduce_predictors_v2(x,y[,1],
  thr1=0.90,          #upper threshold for correlation between predictors
  method="pearson",   #type of correlation
  thr2=NULL,          #lower threshold for correlation between predictors and response
  thr3=0.00,          #lower threshold for coefficient of variation
  thr4=Inf,           #upper threshold for Variance Inflation Factor (VIF)
  maxsize=30,         #maximum number of predictors
  f1=f1)
```

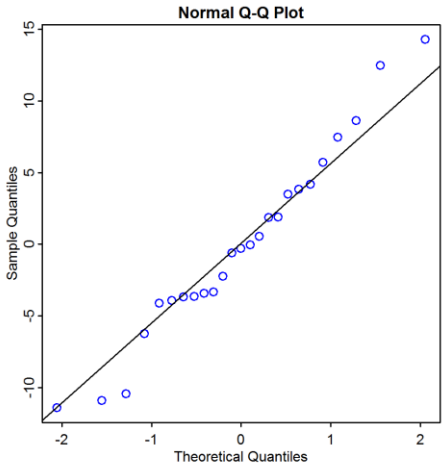
Exemplo práctico

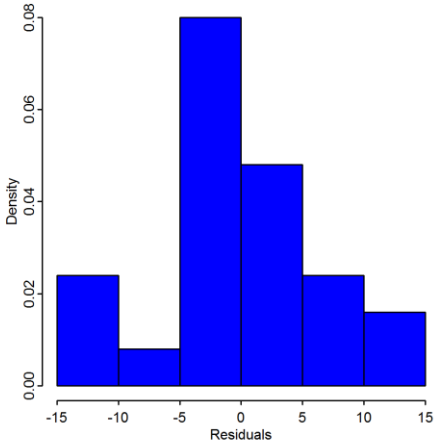
```
#perform GA
f1 <- "../results/lmktattract_2/modsel"
mod1 <- mod_select_lm_v2(x,y,
  maxs1=10,          #maximum number of attributes
  popsize = 100,     #population size
  mutrate = 0.001,   #per locus (i.e. per term) mutation rate, [0,1]
  sexrate = 0.1,     #sexual reproduction rate, [0,1]
  imm = 0.0,         #immigration rate, [0,1]
  deltaM=1e-6,       #Stop Rule: change in mean IC
  deltaB=1e-6,       #Stop Rule: change in best IC
  conseq = 5,        #Stop Rule: times with no improvement
  nreps = 4,         #number of repeats
  f1=f1)

#fit best model
f1 <- "../results/lmktattract_2/fit"
fit_lm(mod1$formula,mod1$data,f1=f1,main="LMkt Mobility")
```

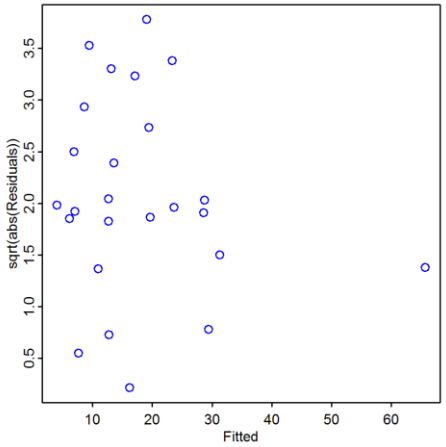
Exemplo práctico

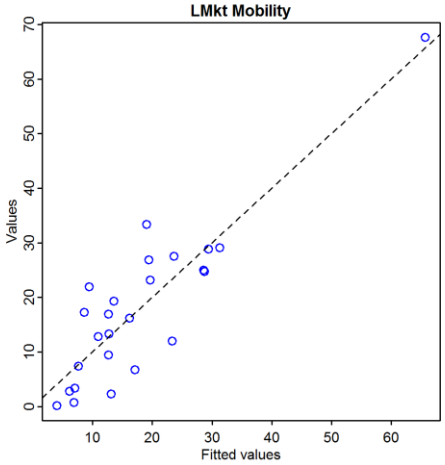


Exemplo práctico









labels	value	st. err.	t value	Pr(>|t|)
(Intercept)	1.82	6.196	0.294	0.772
emp_Y15-24_NaceB-E	-0.38	0.198	-1.929	0.067
emp_Y25-64_NaceK	4.61	0.711	6.474	<0.001
emp_Y25-64_NaceL	10.04	2.832	3.474	<0.001

Adjusted-R^2 = 0.76; F(3, 21) = 25.85; p-value < 0.001.

Exemplo práctico

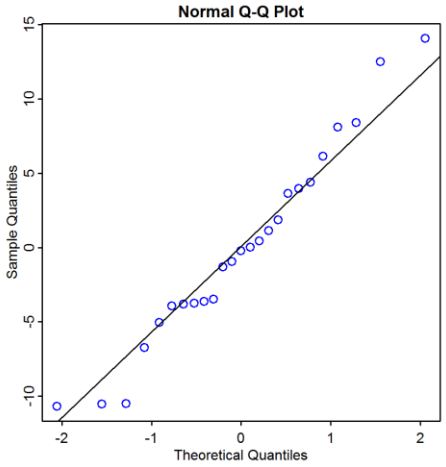
» No **R**

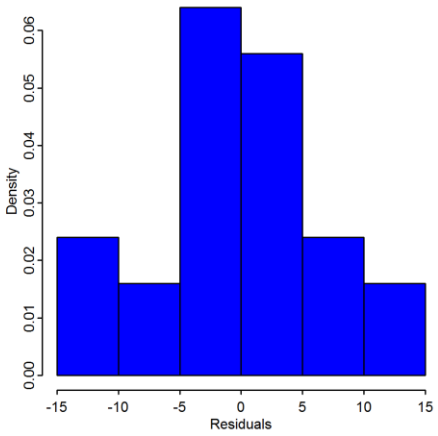
```
#1. FUNCTIONS
source("misc_v2.3.r")

#2. ANALYSE DATA
decay <- 1e-4                          #parameter for weight decay
maxit <- 1000                          #maximum number of iterations
abstol <- 1e-6                         #absolute fit criterion
reltol <- 1e-12                        #relative fit criterion

#fit best model (nn size = 1)
size <- 1                              #number of units in the hidden layer
f1 <- "../results/lmktattract_3/fit_nnet1"
fit_nnet_v2(mod1$formula,mod1$data,nn_size=size,nn_decay=decay,nn_maxit=maxit,
  nn_abstol=abstol,nn_reltol=reltol,f1=f1,main="LMkt Mobility")
```

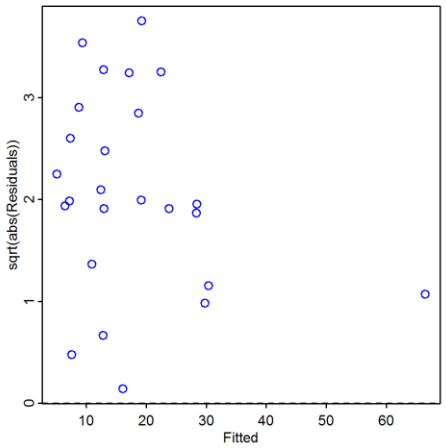
Exemplo práctico

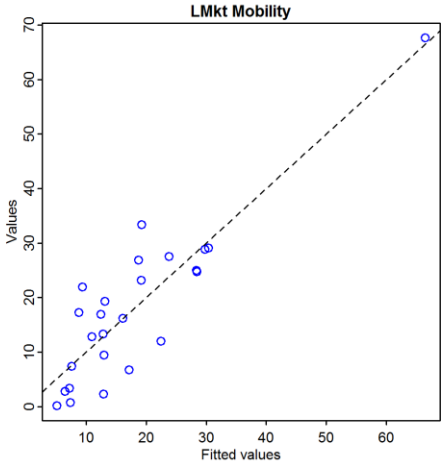
```
#fit best model (nn size = 3)
size <- 3                                  #number of units in the hidden layer
f1 <- "../results/lmktattract_3/fit_nnet3"
fit_nnet_v2(mod1$formula,mod1$data,nn_size=size,nn_decay=decay,nn_maxit=maxit,
  nn_abstol=abstol,nn_reltol=reltol,f1=f1,main="LMkt Mobility")

#fit best model (nn size = 10)
size <- 10                                 #number of units in the hidden layer
f1 <- "../results/lmktattract_3/fit_nnet10"
fit_nnet_v2(mod1$formula,mod1$data,nn_size=size,nn_decay=decay,nn_maxit=maxit,
  nn_abstol=abstol,nn_reltol=reltol,f1=f1,main="LMkt Mobility")
```

Exemplo práctico

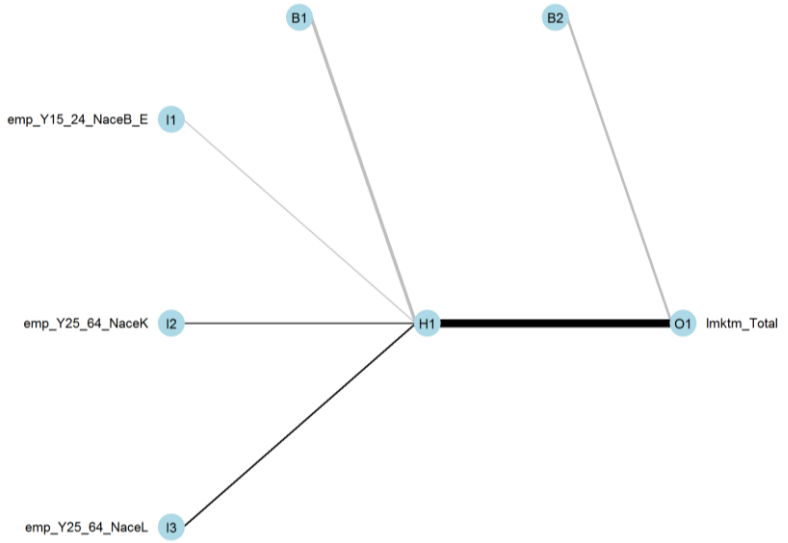




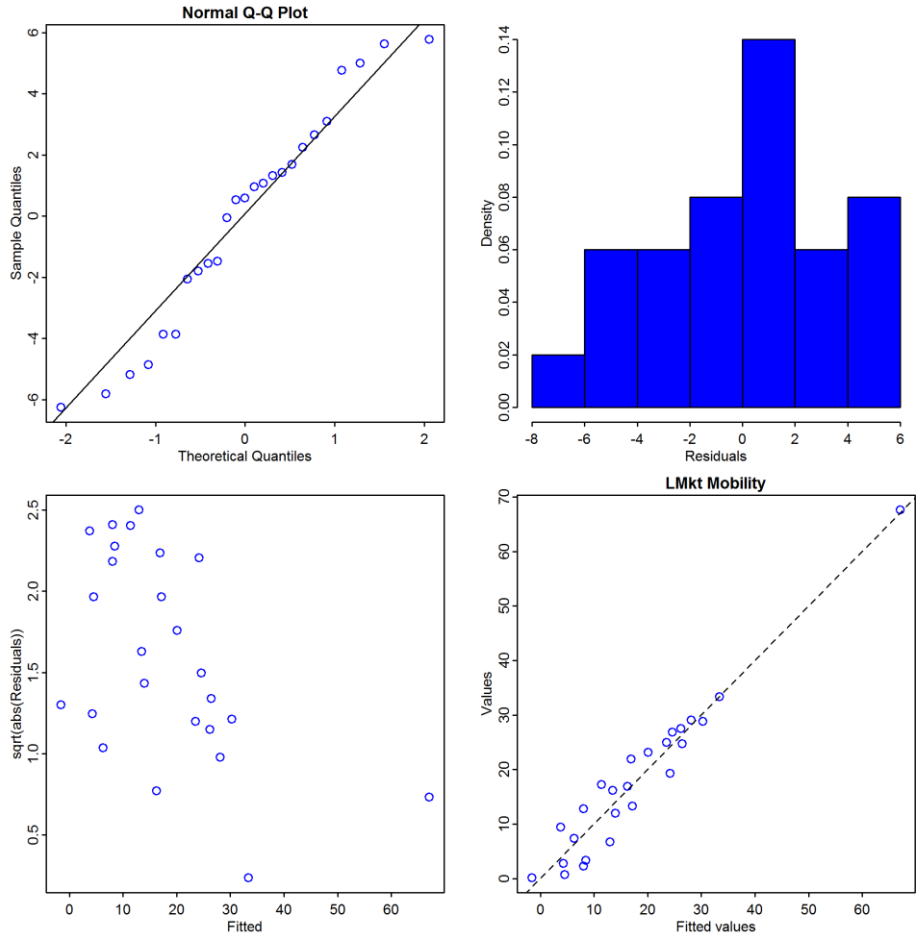




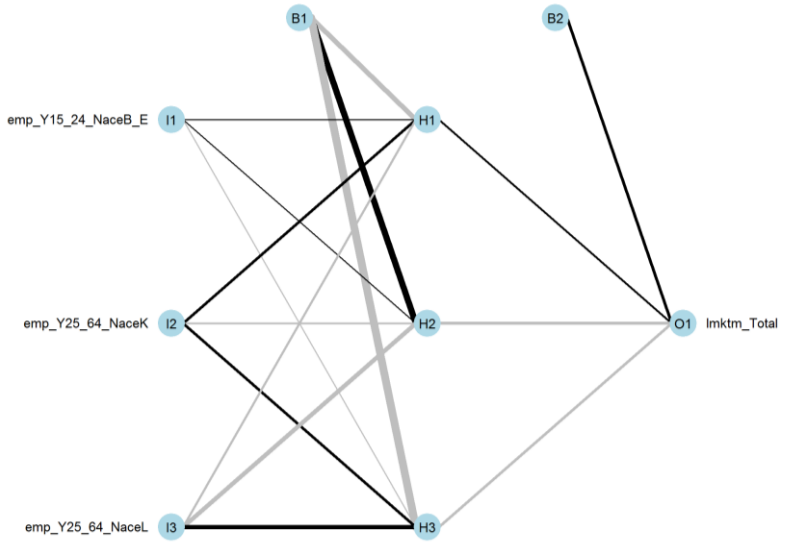
labels	Importance (Olsen, 2004)
emp_Y15-24_NaceB-E	-0.027
emp_Y25-64_NaceK	0.292
emp_Y25-64_NaceL	0.693



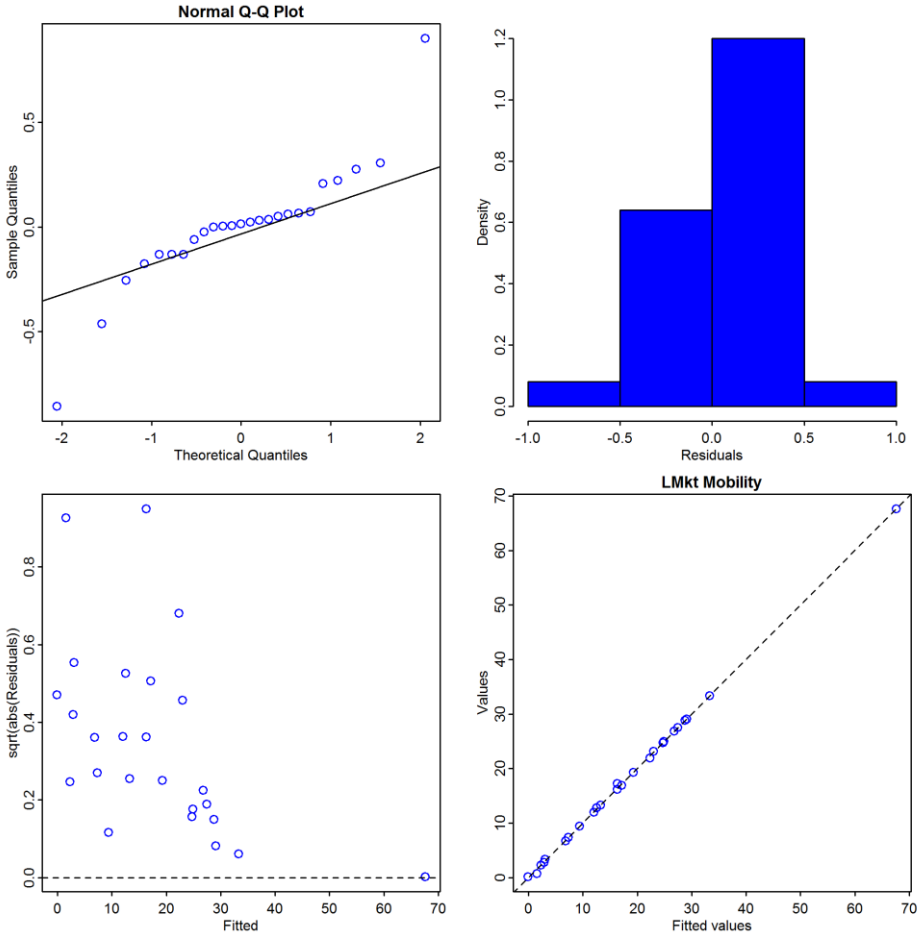
Exemplo práctico



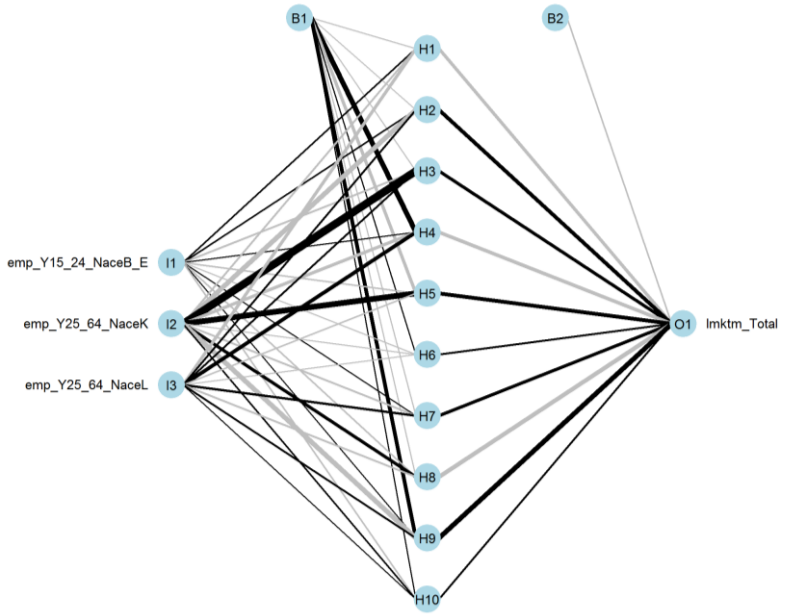
labels	Importance (Olsen, 2004)
emp_Y15-24_NaceB-E	0.060
emp_Y25-64_NaceK	-0.030
emp_Y25-64_NaceL	0.315



Exemplo práctico



labels	Importance (Olsen, 2004)
emp_Y15-24_NaceB-E	-0.651
emp_Y25-64_NaceK	0.511
emp_Y25-64_NaceL	2.483



Bibliografia

» Azzalini A & Scarpa B (2012) Data Analysis and Data Mining - An Introduction. Oxford University Press, New York.

» Due KL & Swamy MNS (2016) Search and Optimization by Metaheuristics - Techniques and Algorithms Inspired by Nature. Springer, Switzerland.

» Gama J, Carvalho APL, Faceli K, Lorena AC e Oliveira M (2012) Extração de Conhecimento de Dados. *Data Mining*. Edições Sílabo. Lisboa.

» Larose DT & Larose CD (2014) Discovering Knowledge in Data – An Introduction to Data Mining. John Wiley & Sons, New Jersey.

» Torgo L, (2017) Data Mining with R – Learning with Case Studies. Taylor & Francis Group, New York.