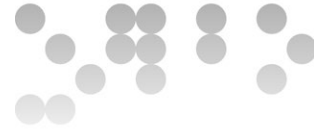


Tabla de contenido

Exercicis.....	2
Exercici 1	2
Normalització per la diferència	2
Escalat decimal	4
Normalització basada en la desviació estàndard: estandardització de valors.....	5
Exercici 2.....	6
Mètodes de selecció d'atributs	6
Mètodes d'extracció d'atributs	8
Exercici 3.....	10
Reemplaçar els valors desconeguts per una constant.....	11
Reemplaçar el valor desconegut amb la mitjana	13
Reemplaçar el valor desconegut amb el valor més freqüent.....	13
Reemplaçar amb un valor aleatori de la distribució de la variable	14
Exercici 4.....	15
Carrega i exàmen preliminar del conjunt de dades	15
Exploració i tractament de valors desconeguts	16
Discretització d'atributs	16
Creació d'un atribut nou	16
Exploració visual de les dades (EDA).....	16
Bibliografia	16



Exercicis

Exercici 1

Explica el concepto de transformació de valors i dona tres exemple on es vegi la seva utilitat.

Per **transformació de valors** entenem modificacions dins el tipus de valors que poden prendre tots o alguns del atributs.

Les operacions més habituals són la **normalització** i la **discretització** de dades. Existeixen varies tècniques per a la transformació de dades, passem a examinar tres de les més importants. Considerarem X com al valor original del atribut i X^* com al valor del atribut normalitzat.

Normalització per la diferència

La normalització per la diferència tracta de compensar l'efecte de la distància del valor que tractem respecte al màxim dels valors observats. Es a dir,

$$X^* = \frac{X - \min(X)}{\text{rango}(X)} = \frac{X - \min(X)}{\max(x) - \min(x)}$$

Exemple de normalització per la diferència



Per a il·lustrar aquesta tècnica utilitzarem el conjunt de dades `cars`¹ del següent llibre *"Data Mining and Predictive Analytics"*²:

```
## Carregem les dades
library(readr)
cars <- read_csv("data/cars.txt")
```

Al següent fragment presentem un resum estadístic de la variable `weightlbs`:

```
## Resum estadístic variable `weight`
summary(cars$weightlbs)

##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      1613   2246   2835   3005   3664   4997
```

Troblem els valors màxim i mínim de la variable `weightlbs`:

```
## Trobem els valors mínim i màxim de la variable `weight`
valor_min <- min(cars$weightlbs)
valor_max <- max(cars$weightlbs)
```

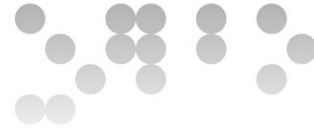
Finalment, trobem el valor normalitzat de la variable `weightlbs`:

```
## Valor normalitzat mitjançant la tècnica de la diferència
valors_norm_dif <- (cars$weightlbs - valor_min)/(valor_max - valor_min)
head(valors_norm_dif)

## [1] 0.76713948 0.09219858 0.54255319 0.63475177 0.12943262 0.67582742
```

¹ Conjunt de dades disponibles en <http://www.dataminingconsultant.com>

² Daniel T. Larouse, Chantal D. Larouse: Data Mining and Predictive Analytics. USA, John Wiley & Sons, 2015, ISBN 978-1-118-11619-7



Escalat decimal

La tècnica del **escalat decimal** ens garanteix que tots els valors normalitzats estiguin entre -1 i 1.

$$X_{decimal}^* = \frac{X}{10^d}$$

on d representa el nombre de dígitos del valor de la dada amb el valor absolut més gran.

Exemple de normalització decimal

Per a explicar aquesta tècnica seguirem treballant amb el conjunt de dades `cars` de l'exemple anterior:

```
## Calculem el valor de decimals  
max(abs(cars$weightlbs))  
## [1] 4997
```

Un cop calculat el nombre de dígitos (d), ens trobem amb condicions de transformar els valors de la variable `weight`:

```
## Valors transformats amb la tècnica decimal  
valors_norm_dec <- cars$weightlbs/10^4  
head(valors_norm_dec)  
## [1] 0.4209 0.1925 0.3449 0.3761 0.2051 0.3900
```



Normalització basada en la desviació estàndard: estandardització de valors

El mètode d'estandardització de valors assegura que s'obtenen valors dins el rang escollit que tenen la propietat que la seva mitjana és zero i la seva desviació estàndard val 1.

Es a dir, l'estandardització consisteix en la diferència entre el valor de l'atribut i la seva mitjana, dividint aquesta diferència per la desviació estàndard dels valors de l'atribut. Es a dir:

$$Z - score = \frac{X - mean(X)}{SD(X)}$$

Exemple de normalització per estandardització

En primer lloc, ens caldrà calcular la mitjana i la desviació estàndard:

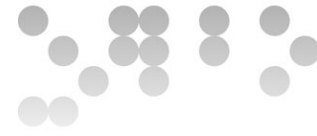
```
## Calculem la mitjana
m <- mean(cars$weightlbs)

## Calculem la desviació estàndard
s <- sd(cars$weightlbs)
```

Finalment, apliquem la transformació mitjançant la formula presentada anteriorment:

```
# Estandardització de valors
valors_norm_z <- (cars$weightlbs - m)/s
head(valors_norm_z)

## [1] 1.4115004 -1.2672210 0.5201570 0.8860769 -1.1194457 1.0490989
```



Exercici 2

Explica el concepte de reducció de nombre d'atributs i dona un exemple on es vegi la seva utilitat. De quines tècniques disposem per a comprovar que no estem perdent qualitat en aquests procés?

La reducció del nombre d'atributs consisteix a trobar un subconjunt dels atributs originals que permeti d'obtenir models de la mateixa qualitat que els que s'obtidrien utilitzant tots els atributs. Aquest problema s'anomena **problema de la selecció òptima d'atributs**.

Aquest problema pot tindre diferents enfocaments, com per exemple: escollir els millors atributs a partir d'un anàlisi preliminar, eliminar atributs redundants o que aporten poca informació, o reduir la dimensionalitat de les dades generant nous atributs a partir dels existents. En tots aquests casos, la finalitat es reduir el cost computacional per a la creació de models.

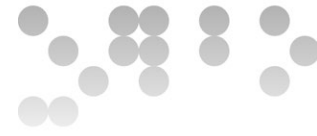
Existeixen els següents mètodes per a tractar amb el problema de la selecció òptima d'atributs:

- Mètodes de selecció d'atributs
- Mètodes de reducció del nombre d'atributs

Tot seguit explicarem en que consisteixen cada un dels mètodes i presentarem algunes de les tècniques més importants per a la reducció d'atributs.

Mètodes de selecció d'atributs

La **selecció d'atributs** consisteix a escollir únicament atributs que són realment rellevants per a resoldre el problema, descartant aquells que no ens aporten informació rellevant per a resoldre el problema.



Depenent de si la selecció de característiques fa ús o no de la informació del mètode de classificació posterior, podem definir la següent taxonomia:

- Els **algoritmes filtre** (*filter*), on els atributs o conjunt d'atributs son evaluats de forma independent respecte del mètode de classificació que s'utilitzarà amb posterioritat.
- Els **algoritmes empotrats** (*wrapper*), on el mètode de selecció de característiques utilitza el classificador que usarà amb posterioritat.

A continuació passem a estudiar els diferents mètodes de selecció de característiques i els algoritmes utilitzats.

En primer lloc, explicarem breument els mètodes per a la selecció d'atributs individuals, coneguts com a **algoritmes univariants**:

- Selecció de màxima rellevància (*maximum relevance selection*), que utilitza el coeficient de correlació entre cada atribut.
- Selecció basada en la informació mútua, mesura la informació mútua entre variables aleatòries que modelen cada característica i les etiquetes de classificació.
- Mètodes basats en tests estadístics, apliquen tests estadístics de hipòtesi sobre les dades, com per exemple el **t-static** o el **chi-square**.

En segon lloc, trobem els mètodes de selecció de subconjunts d'atributs, coneguts com a **algoritmes multivariants**:

- Recerca exhaustiva (*exhaustive search*), consisteix en definir un espai de recerca i avaluar, mitjançant un funció de cost, totes les possibles combinacions. Només es aplicable a problemes de dimensionalitat reduïda.
- Selecció pas a pas (*stepwise selection*), consisteix en iterar per un algoritme en el que cada pas o be afegeix al conjunt d'atributs aquell



atribut que augmenta el rendiment global del conjunt, o bé el elimina aquell atribut que fa que el rendiment empitjori.

- Ramificació i poda (*branch and bound*), consisteix en aplicar la tècnica de recerca **branch and bound**.

Mètodes d'extracció d'atributs

La extracció d'atributs es tracta de calcular nous atributs a partir d'existents, amb l'objectiu de que els nous atributs resumeixin millor la informació que contenen, capturant la naturalesa de la estructura subjacent en les dades.

Anàlisi de Components Principals (PCA)

L'anàlisi de components principals (*Principal Component Analysis*, PCA) ens ajuda a solucionar problemes de reducció de dimensionalitat i extracció de característiques en les nostres dades de manera automàtica. El PCA es un algoritme molt conegut en l'àmbit de l'anàlisi de dades, i té moltes aplicacions diferents. Informalment, es pot definir com la tècnica que intenta aconseguir una representació d'un conjunt de dades a un espai de dimensionalitat més reduïda, minimitzant l'error quadràtic.

Exemple de Anàlisi de Components Principal (PCA)

Per a il·lustrar aquest exemple farem ús del *dataset* `houses`^[3]:

```
# Realitzem la lectura de les dades
library(readr)
houses <- read_delim("data/houses.csv", ";",
  escape_double = FALSE, col_names = FALSE,
  trim_ws = TRUE, skip = 1)
```

A continuació preparem les dades per a realitzar l'anàlisi:



```
# Donem nom als atributs
names(houses) <- c("MVAL", "MINC", "HAGE", "ROOMS", "BEDRMS", "POPN",
"HHLDS", "LAT", "LONG")
```

A continuació normalitzem les dades amb el mètode d'estandardització de valors que hem tractat en el [exercici 1](#):

```
# Estandarditzem les variables

houses$MVAL_Z <- (houses$MVAL - mean(houses$MVAL))/(sd(houses$MVAL))
houses$MINC_Z <- (houses$MINC - mean(houses$MINC))/(sd(houses$MINC))
houses$HAGE_Z <- (houses$HAGE - mean(houses$HAGE))/(sd(houses$HAGE))
houses$ROOMS_Z <- (houses$ROOMS - mean(houses$ROOMS))/(sd(houses$ROOMS))
houses$BEDRMS_Z <- (houses$BEDRMS - mean(houses$BEDRMS))/(sd(houses$BEDRMS))
houses$POPN_Z <- (houses$POPN - mean(houses$POPN))/(sd(houses$POPN))
houses$HHLDS_Z <- (houses$HHLDS - mean(houses$HHLDS))/(sd(houses$HHLDS))
houses$LAT_Z <- (houses$LAT - mean(houses$LAT))/(sd(houses$LAT))
houses$LONG_Z <- (houses$LONG - mean(houses$LONG))/(sd(houses$LONG))
```

Seleccionarem una mostra aleatòria del 90% del conjunt de dades:

```
# Seleccionem aleatoriament el 90% de les dades per al joc de proves
dist_unif <- runif(dim(houses)[1], min = 0, max = 1)
test_houses <- houses[which(dist_unif < .1), ]
train_houses <- houses[which(dist_unif <= .1), ]
```

Per a realitzar el PCA utilitzarem el paquet `psych`³:

```
# Instal·lem el paquet
# install.packages("psych",dependencies=TRUE)
# Carreguem el paquet en la sessió
library(psych)

# Anàlisi de Components Principal (PCA)
pca_analysis <- principal(train_houses[, c(10:17)],
                           nfactors = 8,
```

³ Podem consultar la documentació en <https://cran.r-project.org/web/packages/psych/index.html>



```
rotate = "none",
scores = TRUE)
```

A continuació es mostren els resultats de l'ànalisi PCA:

```
# Resultas PCA
# Valors propis (eigen)
pca_analysis$values

## [1] 3.90159314 1.73176284 1.01379849 0.87055944 0.28562899 0.12512193
## [7] 0.05699476 0.01454041

# Mostrem la matriu amb les variàncies
pca_analysis$loadings

##
## Loadings:
##          PC1      PC2      PC3      PC4      PC5      PC6      PC7      PC8
## MVAL_Z      0.904      0.225 -0.344
## MINC_Z      0.912  0.201      0.343
## HAGE_Z    -0.429      -0.438  0.775  0.133
## ROOMS_Z    0.963      -0.166 -0.161
## BEDRMS_Z   0.975      -0.116
## POPN_Z     0.934 -0.122      0.124  0.280
## HHLDS_Z     0.976      0.117      0.135
## LAT_Z      -0.218  0.874  0.428
##
##          PC1      PC2      PC3      PC4      PC5      PC6      PC7      PC8
## SS loadings  3.902  1.732  1.014  0.871  0.286  0.125  0.057  0.015
## Proportion Var 0.488  0.216  0.127  0.109  0.036  0.016  0.007  0.002
## Cumulative Var 0.488  0.704  0.831  0.940  0.975  0.991  0.998  1.000
```

Exercici 3

De quines tècniques disposem per resoldre el problema de la possible falta de valors d'un atribut? Dona almenys un exemple de cada tècnica.

Un dels problemes més habituals en el tractament previ de les dades és l'absència de valors per a un atribut determinat.



Algunes de les opcions disponibles són les següents:

1. Reemplaçar els valors desconeguts per una constant, especificada per l'analista.
2. Reemplaçar el valor desconegut amb la mitjana (per a variables numèriques) o la moda (per a variables categòriques).
3. Reemplaçar els valors desconeguts amb un valor generat aleatòriament de la distribució de la variable.

Reemplaçar els valors desconeguts per una constant

En els següents exemples utilitzarem el *dataset* `cars` que ja hem utilitzat en apartats anteriors:

```
# Importem les dades
library(readr)
cars <- read_csv("data/cars.txt")
```

Realitzem un primer contacte amb el joc de dades, visualitzan la seva estructura i els 6 primers registres:

```
# Realitzem un examen preliminar del conjunt de dades
str(cars)
head(cars)
```

Per tal de simplificar l'exemple i millorar la llegibilitat del document només treballarem amb quatre variables:

```
# Seleccionem les variables mpg, cubicinches, hp i brand
my_cars <- cars[, c(1, 3, 4, 8)]
head(my_cars)

## # A tibble: 6 x 4
##   mpg cubicinches   hp brand
##   <dbl>      <int> <int> <chr>
## 1  14.0         350  165  US.
## 2  31.9          89   71 Europe.
```



```
## 3  17.0      302   140 US.
## 4  15.0      400   150 US.
## 5  30.5       98    63 US.
## 6  23.0      350   125 US.
```

Per tal de demostrar les diferents tècniques farem que el *dataframe* `my_cars` tingui valors desconeguts:

```
# Fem certs valors desconeguts
my_cars[2, 2] <- NA
my_cars[4, 4] <- NA
head(my_cars)

## # A tibble: 6 x 4
##   mpg cubicinches   hp brand
##   <dbl>      <int> <int> <chr>
## 1  14.0        350   165 US.
## 2  31.9         NA    71 Europe.
## 3  17.0        302   140 US.
## 4  15.0        400   150 <NA>
## 5  30.5         98    63 US.
## 6  23.0        350   125 US.
```

Tot seguit, es mostra com reemplaçar els valors desconeguts amb constants:

```
# Amb l'ajuda de un test lògic descobrim els valors desconeguts
missing_values_cubicinches <- is.na(my_cars$cubicinches)
missing_values_brand <- is.na(my_cars$brand)

# Reemplacem els valors desconeguts amb constants
my_cars$cubicinches[missing_values_cubicinches] <- 0
my_cars$brand[missing_values_brand] <- "Valor desconegut"
head(my_cars)

## # A tibble: 6 x 4
##   mpg cubicinches   hp brand
##   <dbl>      <dbl> <int> <chr>
## 1  14.0        350   165 US.
## 2  31.9         0    71 Europe.
## 3  17.0        302   140 US.
## 4  15.0        400   150 Valor desconegut
## 5  30.5        98.0    63 US.
## 6  23.0        350   125 US.
```



Reemplaçar el valor desconegut amb la mitjana

A continuació es mostra un exemple de com reemplaçar valors desconeguts amb la mitjana:

```
# Reemplacem els valors desconeguts amb la mitjana
my_cars$cubicinches[missing_values_cubicinches] <- mean(na.omit(my_cars$c
cubicinches))
head(my_cars)

## # A tibble: 6 x 4
##   mpg cubicinches   hp brand
##   <dbl>      <dbl> <int> <chr>
## 1  14.0         350   165  US.
## 2  31.9         202    71 Europe.
## 3  17.0         302   140  US.
## 4  15.0         400   150 <NA>
## 5  30.5          98.0    63  US.
## 6  23.0         350   125  US.
```

Reemplaçar el valor desconegut amb el valor més freqüent

A diferència de altres mesures estadístiques, R no proporciona una funció predefinida per al càlcul de la moda. Es per això, que crearem una funció per a calcular el valor més freqüent en un conjunt de dades. Aquesta funció pren com a argument un vector i retorna el valor més freqüent:

```
# Funció per al càlcul de la moda
getmode <- function(v) {
  uniqv <- unique(v)
  uniqv[which.max(tabulate(match(v, uniqv)))]
}

# Trobem el valor més freqüent
moda <- getmode(my_cars$brand)
moda

## [1] "US."
```



```
# Reemplacem els valors desconeguts amb la moda
my_cars$brand[missing_values_brand] <- moda
head(my_cars)
```

```
## # A tibble: 6 x 4
##   mpg cubicinches   hp brand
##   <dbl>      <dbl> <int> <chr>
## 1  14.0        350   165 US.
## 2  31.9        202    71 Europe.
## 3  17.0        302   140 US.
## 4  15.0        400   150 US.
## 5  30.5         98.0    63 US.
## 6  23.0        350   125 US.
```

Reemplaçar amb un valor aleatori de la distribució de la variable

Per últim, a continuació es mostra un exemple de com reeemplaçar amb un valor aleatori de la distribució de la variable:

```
# Generem observacions aleatòries
random_cubinchess_obs <- sample(na.omit(my_cars$cubicinches), 1)
random_brand_obs <- sample(na.omit(my_cars$brand), 1)
random_cubinchess_obs

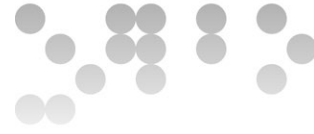
## [1] 360

random_brand_obs

## [1] "US."

# Reemplacem els valors desconeguts amb les observacions aleatòries
my_cars$cubicinches[missing_values_cubicinches] <- random_cubinchess_obs
my_cars$brand[missing_values_brand] <- random_brand_obs
head(my_cars)
```

```
## # A tibble: 6 x 4
##   mpg cubicinches   hp brand
##   <dbl>      <dbl> <int> <chr>
## 1  14.0        350   165 US.
## 2  31.9        360    71 Europe.
## 3  17.0        302   140 US.
## 4  15.0        400   150 US.
```



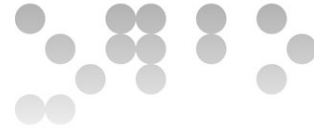
```
## 5  30.5      98.0    63 US.  
## 6  23.0     350    125 US.
```

Exercici 4

A partir del joc de dades disponible en el següent enllaç: <http://archive.ics.uci.edu/ml/datasets/Adult> realitza un estudi similar al que se ha realitzat amb el joc de dades “Titanic”. Explica el procés que has seguit, quin coneixement has extret, quins objectius t’havies fixat i quins passos i tècniques has emprat.

Carrega i exàmen preliminar del conjunt de dades

```
# Carrega del joc de dades  
datosAdult <- read.csv('http://archive.ics.uci.edu/ml/machine-learning-databases/adult/adult.data', stringsAsFactors = FALSE, header = FALSE)  
  
# Noms dels atributs  
names(datosAdult) <- c("age", "workclass", "fnlwgt", "education", "education-num", "marital-status", "occupation", "relationship", "race", "sex", "capital-gain", "capital-loss", "hour-per-week", "native-country", "income")
```



Exploració i tractament de valors desconeguts

Discretització d' atributs

Creació d' un atribut nou

Exploració visual de les dades (EDA)

Bibliografia

[1] Daniel T. Larouse, Chantal D. Larouse: Data Mining and Predictive Analytics. USA, John Wiley & Sons, 2015, ISBN 978-1-118-11619-7

[2] Jordi Gironés Roig, Jordi Casas Roma, Julià Minguillón Alfonso, Ramon Caihuelas Quiles : Minería de Datos: Modelos y Algoritmos. Barcelona, Editorial UOC, 2017, ISBN: 978-84-9116-904-8.

[3] Jiawe Han, Michellie Chamber & Jian Pei: Data mining : concepts and techniques. 3^o Edition. USA, Editorial Elsevier, 2012, ISBN 978-0-12-381479-1