

# spamizer

*Marc Sànchez, Francesc Xavier Bullich, Gil Gassó*

*5/8/2019*

```

// TODO : Posar el projecte al github.

# x és el nom del fitxer que volem carregar
loadFormattedData <- function(x){

  tmp = read.csv(x)
  names(tmp) <- c("id", "phi", "k", "tp", "tn", "fp", "fn", "nham", "nspam")

  #Calculem els tcr dels valors
  # BASE : (NSPAM) / (50 * NHAM + NSPAM)
  base <- tmp$nspam / (50 * tmp$nham + tmp$nspam)
  # WERR: (50 * FP + FN)/(50 * NHAM + NSPAM) + 0.000001 -> per que no sigui 0
  werr <- (50 * tmp$fp + tmp$fn) / (50 * tmp$nham + tmp$nspam) + 0.000001
  # TCR : BASE / WERR
  tcr <- base/werr

  # Calculem l'accuracy
  accuracy <- (tmp$nspam + tmp$nham - tmp$fp - tmp$fn)/(tmp$nspam + tmp$nham) * 100

  # Generar una matriu que permeti representar els resultats en funció de k i phi
  values <- data.frame(accuracy, tcr)
  names(values) <- c("accuracy", "tcr")
  head(values)

  tmp <- cbind(tmp, values)

  # Ordenem els valors
  tmp <- tmp[order(-tmp$tcr), ]

  return(tmp)
}

```

## Naive Bayes

En aquest apartat s'especifica com s'adapta el mètode de naive bayes al filtratge de correu.

### Naive Bayes.

// TODO : S'ha de parlar de tot el que es fa a dins del mètode que tenim implementat al codi.

### Assumpcions.

// TODO : Comentar tot el que es dona per sentat al utilitzar aquest mètode, com per exemple que la màquina està ben entrenada ...

### Punts forts i febles del mètode de Naive Bayes.

// TODO :

## Applicació.

En aquest apartat s'es

### Tecnologies escollides.

Comentar també la comparació de l'ús de base de dades en memòria.

### Manual de l'aplicació.

```
# usage: spamizer
# -c <arg> Usage : -c <spamDir> <hamDir> [-n <int>]
# Receives 2 parameters, A directory with spam mails and a
# directory with ham mails. A calculation for values phi and k
# will be done using a selection for the mails set. By default
# the selection will be random based on k-fold cross-validation
# and the heuristic method used to calculate phi and k values
# will be random
# -d Flag that indicates that data must be loaded from local
# database, this database is allocated inside project dir named
# db made by csv files
# -h Set training mails as ham, adding this argument -s must not be
# present
# -n <arg> The number of iterations for -c mode execution.
# -p Set the persistance of the memory database to a local database
# -s Set training mails as spam, adding this argument -h must not
# be present
# -t <arg> Directories where training mails in txt are stored, this or
# database argument must be present you can set a maximum of 2
# directories in this several order : -t <spamDir> <hamDir>. If
# only one dir is set the parameter -h or -s must be included
# -v <arg> Directory where validation mails in txt are stored. This
# procedure will validate mail inside validationDir with
# database loaded by default or stored inside memory. [-h / -s]
# -v <validationDir> .
```

## Utilització.

### Implementació.

#### Lectura de fitxers.

#### Mètode de selecció.

##### Adaptació del mètode K-fold cross-validation.

En comptes de realitzar la divisió ...

Filtre i abstracció del filtratge.

Stanford Core NLP.

Custom Filter.

Entrenament.

Validació.

// TODO : Explicar la nostre adaptació del mètode hill climbing utilitzat.

Compute (Application, adaptació del mètode Hill Climbing).

Fase Experimental.

// TODO : ... Pensar l'estructura encara.

Evaluació dels FP i dels FN en funció de K i PHI

// TODO :

Càcul del TCR (Total Cost Ratio)

Amb el Total cost ratio podem extreure un valor que pondera amb més força el valor de les aparicions dels falços positius. El que es busca amb el Tcr és el valor màxim possibles. Per fer-ho hem realitzat varis execucions i hem preparat una sèrie de conclusions per intentar esbrinar les funcions phi i k que millor s'acosten al nostre problema mitjançant el càlcul del tcr.

Veiem com es pot generar la columna TCR

```
results = read.csv("/Users/marc Sanchez/Projects/spamizer/analisis/20000m-500n-SF.csv")
#Calculem els tcr dels valors
# BASE : (NSPAM) / (50 * NHAM + NSPAM)
base <- results$NSPAM / (50 * results$NHAM + results$NSPAM)
# WERR: (50 * FP + FN)/(50 * NHAM + NSPAM) + 0.000001 -> per que no sigui 0
werr <- (50 * results$FP + results$FN) / (50 * results$NHAM + results$NSPAM) + 0.000001
# TCR : BASE / WERR
tcr <- base/werr

# Generar una matriu que permeti representar els resultats en funció de k i phi
values <- data.frame(results$PHI, results$K, tcr)
names(values) <- c("phi", "k", "tcr")
head(values)

##          phi         k      tcr
## 1 2.103004 0.8211889 29.954564
## 2 1.140895 1.5680716 11.313981
## 3 3.032053 0.7119040 10.994228
```

```

## 4 3.627198 0.1008787 10.737433
## 5 1.033468 0.2739749 9.265549
## 6 1.647332 0.9765125 9.136871

# Ordenem els valors
values <- values[order(-values$tcr), ]
head(values, 20)

##          phi         k      tcr
## 1298 2.697174 0.22600527 33.67084
## 1120 1.236293 2.34200744 30.15416
## 1  2.103004 0.82118893 29.95456
## 1058 1.879062 0.02760826 29.90953
## 1491 3.137478 0.23219458 29.04305
## 1355 2.798966 0.09099896 27.00852
## 576  1.924919 0.38421715 26.20279
## 930  4.382425 2.86167496 25.13894
## 1332 2.300543 0.34994918 24.90058
## 1205 2.277744 2.09654442 23.51241
## 1184 4.532676 2.01806091 21.53804
## 1267 3.261305 0.99216020 20.27118
## 1179 4.653059 2.43772195 18.95931
## 1172 3.035856 1.56213014 18.69684
## 585  3.352996 1.72896730 17.93125
## 1242 4.096789 2.80409060 15.30624
## 582  2.452848 0.24901731 13.80774
## 988  4.544143 0.48137259 13.14401
## 651  2.694429 0.84360478 11.99221
## 1424 1.741707 2.31180243 11.79649

```

## Anàlisi de la PHI i la K.

Carreguem les diferents simulacions en un dataframe per poder processar les dades, utilitzem la funció loadFormattedData declarada a l'aratat de funcions del document. Aquesta funció ens afegeix les columnes calculades per l'accuracy i el tcr.

```

b1 = loadFormattedData("/Users/marc Sanchez/Projects/spamizer/analisys/m20000-n9500-SF-P-16-k-03.csv")
b2 = loadFormattedData("/Users/marc Sanchez/Projects/spamizer/analisys/m20000-n10000-P-15-K-03.csv")
b3 = loadFormattedData("/Users/marc Sanchez/Projects/spamizer/analisys/20000m-500n-SF.csv")
b4 = loadFormattedData("/Users/marc Sanchez/Projects/spamizer/analisys/2000m-1000n-phi-1.7-2.3-k-0-0.5.csv")

v <- rbind(b1, b2, b3, b4)
v <- v[order(-v$tcr), ]
head(v)

##          id      phi         k      tp      tn      fp      fn      nham      nspam      accuracy      tcr
## 66031 8637 1.778913 0.5925286 695 625 0 11 695 636 99.17355 57.63278
## 26510 2299 1.362548 0.6460839 688 644 0 15 688 659 98.88641 43.83089
## 14114 1064 2.179125 0.2739705 607 542 0 13 607 555 98.88124 42.59106
## 38981 5932 1.734514 1.0386465 694 666 0 16 694 682 98.83721 42.53095
## 20121 4046 2.723349 0.4087232 836 740 0 18 836 758 98.87077 42.01178
## 59141 7948 4.616297 0.4507945 1075 948 0 25 1075 973 98.77930 38.83499

```

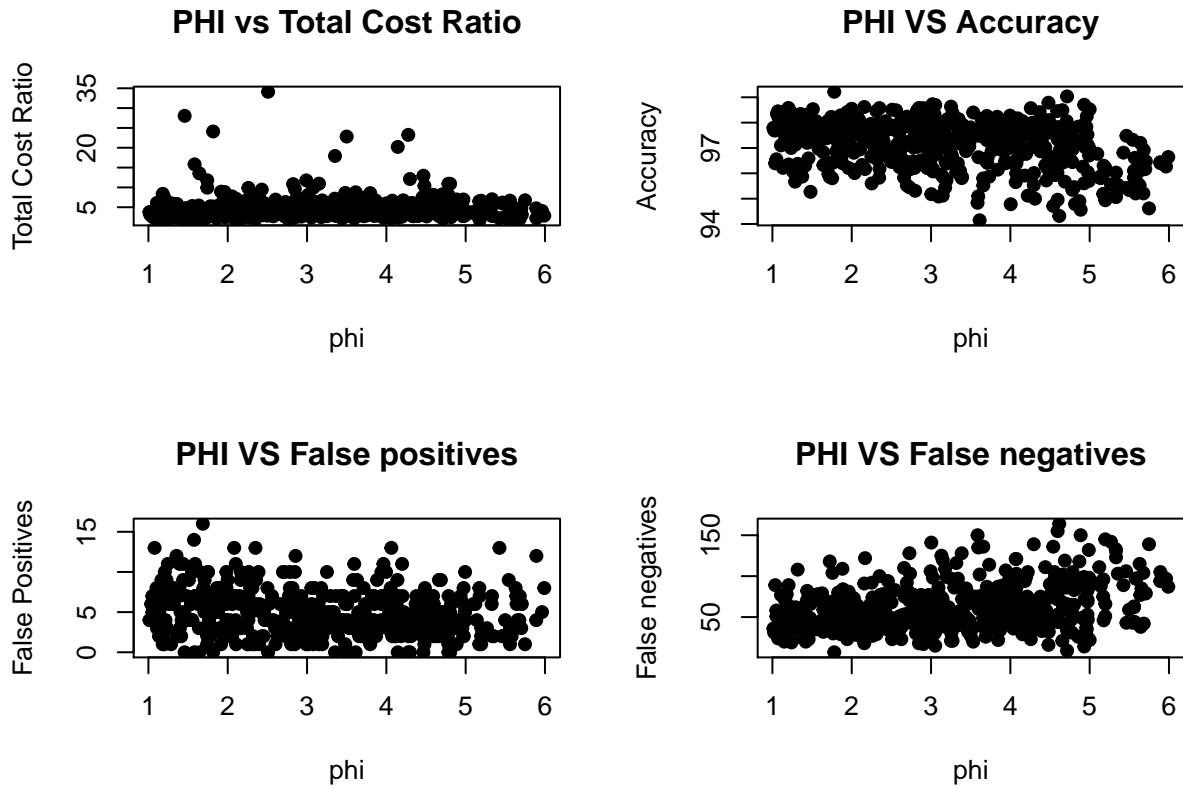
## PHI

Si tenim en compte el què representa el valors de phi, el que ens trobem és que la phi és el factor d'increment de la probabilitat per que un correu sigui considerat SPAM. És a dir un valor de phi = 2, provoca que per que un correu sigui considerat spam la seva probabilitat ha de ser 2 cops superior a la probabilitat de que sigui ham. Un valor de phi = 1 fa que no hi hagi increment obligatori per a la comparació.

El valor mínim que té sentit assignar-li a phi és 1 i el màxim el podríem limitar a 5 com a molt o inclús a 6 si el que volem és no tenir cap correu que sigui Ham i que el consideri com Spam. Aquest paràmetre se'l pot considerar més influent que el valor de k ja que el valor de phi està directament lligat al nombre de falsos positius i de falsos negatius. En canvi el valor de k representa un coeficient molt baix a aplicar a totes les paraules.

Veiem els següents diagrames de dispersió donada una mostra de 500 punts sobre el total de les execucions.

```
m <- v[sample(nrow(v), size = 500), ]  
  
par(mfrow=c(2,2))  
plot(m$phi, m$tcr, main="PHI vs Total Cost Ratio",  
     xlab="phi", ylab="Total Cost Ratio", pch=19)  
  
plot(m$phi, m$accuracy, main="PHI VS Accuracy",  
     xlab="phi", ylab="Accuracy", pch=19)  
  
plot(m$phi, m$fp, main="PHI VS False positives",  
     xlab="phi", ylab="False Positives", pch=19)  
  
plot(m$phi, m$fn, main="PHI VS False negatives",  
     xlab="phi", ylab="False negatives", pch=19)
```



```
par(mfrow=c(1,1))
```

En l'anterior grid podem veure diferents comparacions del comportament de la variable phi sobre una mostra de 500 elements dins del conjunt total de les execucions. Dels gràfics anteriors podem extreure certes conclusions a vista tenint en compte que durant les execucions no s'ha fixat en cap moment ni un ordre de lectura de correus, ni un valor per k ni un valor per phi i els correus per validar eren seleccionats aleatoriament. De totes maneres disposem d'un número molt elevat i amb molta varietat de resultats.

- Es pot veure que la mitjana del tcr queda entre 1 i 6.
- Es pot veure com a més valor de phi, més disminueix el nombre de fp (lentament).
- Es pot veure com a més valor de phi més augmenta el nombre de fn (més pronunciat).
- Es pot veure com l'accuracy es entre el 97 i 99 però que si el valor de phi augmenta llavors l'accuracy baixa 4 punts.

## K

Quan apliquem el suavitzat hem de tenir en compte què passa si donats el bag of words de ham i el de spam una paraula no existeix. En la nostre fòrmul aquest valor ens podria proporcionar multiplicacions per 0 i farà que si una paraula no existeix aquesta paraula ja ens determini si un correu és ham o és spam.

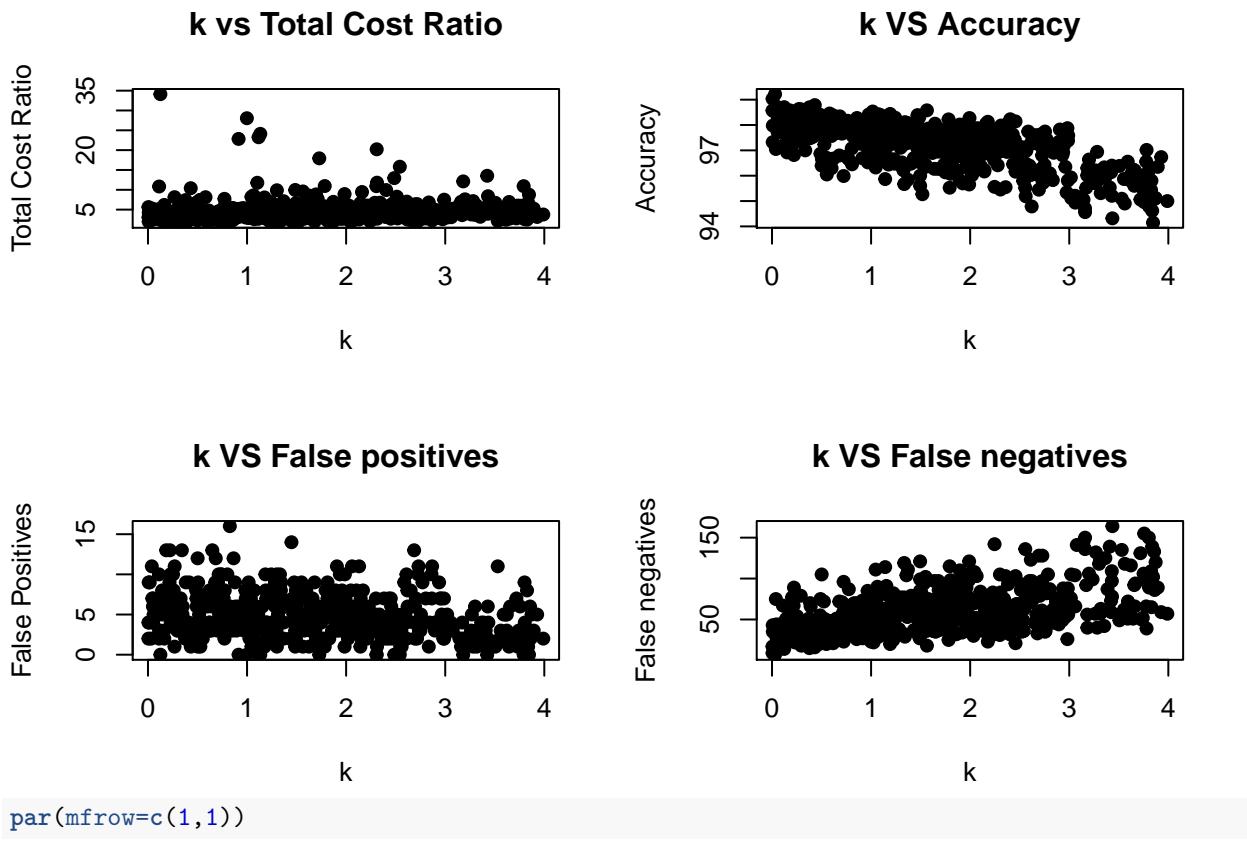
Per tant la k estipula el valor que se li assigna a una paraula quan aquesta no és present. Aquest valor no pot ser 0 però pot ser proper a zero. Si fos zero es provocaria el mateix cas que l'esmentat anteriorment. Tanmateix no té sentit aplicar un valor molt gran a la k ja que si ho fem aquest valor provocaria que les paraules que no existeixen fossin puntuades molt altes i se li treuria valor de càlcul a les aparicions.

```
par(mfrow=c(2,2))
plot(m$k, m$tcr, main="k vs Total Cost Ratio",
     xlab="k", ylab="Total Cost Ratio", pch=19)

plot(m$k, m$accuracy, main="k VS Accuracy",
     xlab="k", ylab="Accuracy", pch=19)

plot(m$k, m$fp, main="k VS False positives",
     xlab="k", ylab="False Positives", pch=19)

plot(m$k, m$fn, main="k VS False negatives",
     xlab="k", ylab="False negatives", pch=19)
```



Utiitzant el mateix supòsit que en la variable phi observem doncs :

- Amb els valors de k per el ter passa quelcom molt similar als valors de phi.
- Amb els valors de k més petits l'accuracy ha augmenta, a mesura que es fa créixer el valor de k més disminueix l'accuracy.
- Veiem que no impacte molt aquest valor en el dels falços positius.
- Per altra banda veiem que té una relació directe amb el comportament dels falços negatius.

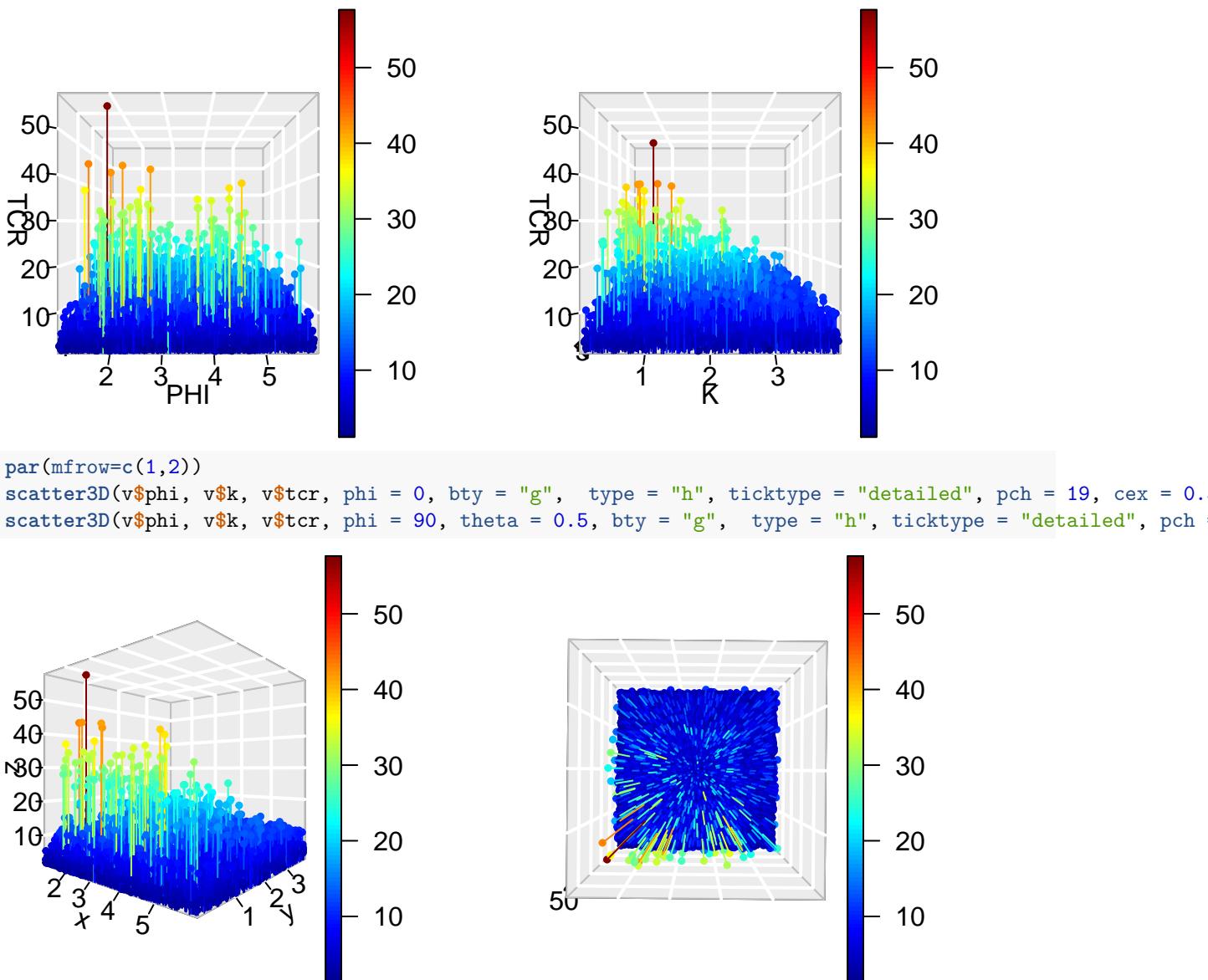
Limitarem els valors de k en un rang de (0 - 1].

### Conclusions conjuntes entre phi i k

No té sentit mirar les formes dels valors de phi i k de manera independent per què son valors generats aleatoriament, els seus histogrames es presenten de la següent manera.

El que sí que té sentit és observar si les variables es poden descriure conjuntament amb el nombre de FP o FN i finalment si es poden comprovar mitjançant el total cost ratio. La variable phi està directament lligada amb els valors FP i FN per definició.

```
par(mfrow=c(1,2))
scatter3D(v$phi, v$k, v$tcr, phi = 0, theta=0, bty = "g", type = "h", ticktype = "detailed", pch = 19,
scatter3D(v$phi, v$k, v$tcr, phi = 0, theta=90, bty = "g", type = "h", ticktype = "detailed", pch = 19)
```



Segons el coeficient de correlació lineal de Pearson la variable phi i el tcr sembla que concentrin la seva relació lineal en una línia recta i constant en un valor d'entre 1 i 5 per el tcr. Podem parlar més o menys que la variable phi li passa el mateix.

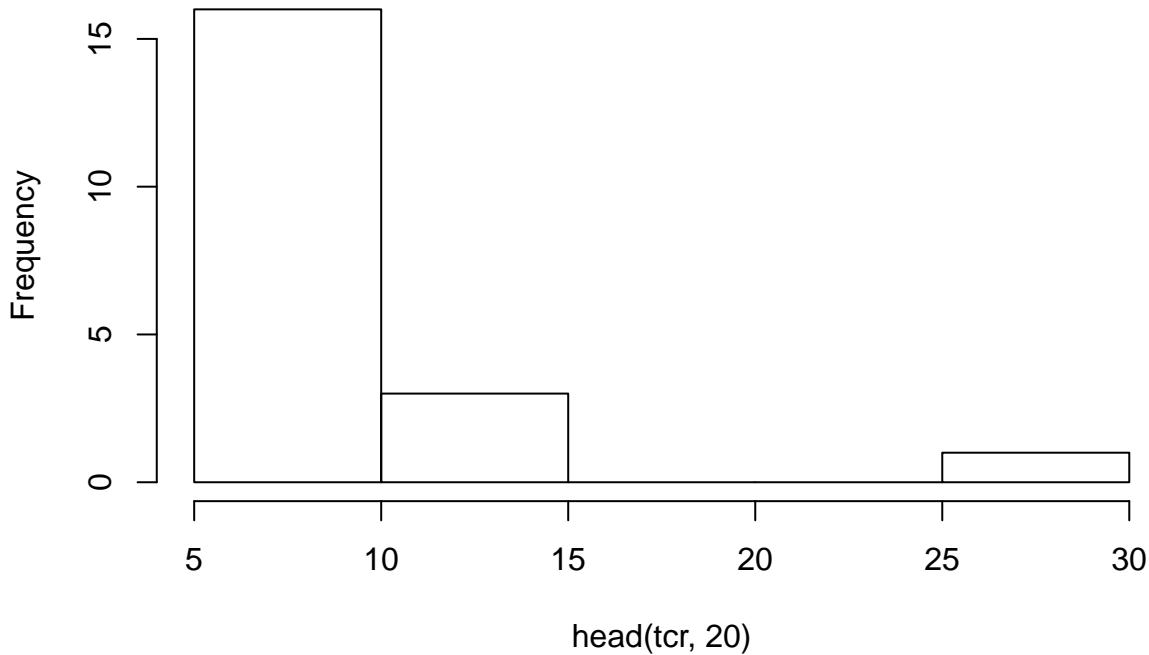
No hem d'oblidar però que el valor del tcr s'ha tret directament de l'execució amb les variables phi i k i per tant ens serveix per veure si hi ha algun patró pel que la funció k o la funció phi de manera independent entre elles poden fer que creixi el valor del tcr, tanmateix això no serà possible degut a que el valor del tcr s'estreu tant de la variable phi tant com de la variable k i **S'hauria de fixar o bé la phi o bé la k per poder extreure una conclusió sobre el tema.**

### Cerca d'un model per la variable TCR

Podem mirar quina forma té la variable TCR segons les execucions generades. Veiem que el total cost ratio concentra per valors aleatòris el seu pes entre els valors 2 - 7.

```
# Histograma de tcr
hist(head(tcr, 20))
```

## Histogram of head(tcr, 20)



```
# Mitjana per el valor del tcr per phi i per k amb rang entre [1,5] i (0-3).
median(tcr)
```

```
## [1] 3.527646
```

El que es pretén és realitzar un estudi de quan les variables phi i k considerades com a constants en l'execució del programa es comporten de manera adient per el filtratge. En una primera iteració amb prop de 1500 execucions utilitzant sense lemmatitzar amb un nombre total de 20000 correus i generant una discriminació d'entre un 5% i un 15% per a la validació ens trobem el següent gràfic.

Veiem que amb uns valors aproximats d'entre 0 i 1 per k i entre 1.7 i 3.5 de phi, s'hi concentren els que tenen el tcr més alt. Per tant generem una altre iteració d'uns 1000 valors restringint aquest rang per phi i k en el quadrant on apareixen més aparicions dels valors de phi i de k concretament phi entre [1.7,2.3] i k entre (0,0.30] i tornant a visualitzar el resultat.

Veiem que la mitjana ha augmentat respecte l'anterior però poc, altres indicadors que podem fer servir son per exemple el valor màxim trobat o fins i tot la mitjana dels 50 valors més alts.

A continuació comparem els valors per les dos rangs i observerem els histogrames dels 50 valors millors per les dues distribucions de resultats per veure si ens han aparegut valors més bons restringit el rang.

Tenint en compte que les dades limitades son de 500 elements i les dades que tenen el rang més ampli son de 1500 i veient com es mantenen els valors més alts mirem de concretar més els resultats i si es pot ajustar més el rang dels valors generats per phi i k mitjançant els gràfics següents :

```
#scatter3D(resultsP1723K005$PHI, resultsP1723K005$K, tcrl, phi = 0, bty = "g", type = "h", ticktype =
#scatter3D(resultsP1723K005$PHI, resultsP1723K005$K, tcrl, phi = 90, theta = 0.5, bty = "g", type = "h")
```

### Recol·lecció dels millors valors

Per ara sabem que ajustant el rang una mostra de 500 valors es comporta de manera similar que una mostra de 1500 valors ambdós generats aleatoriament tant per phi com per k, això ens fa pensar que aquest ajustament

s'està comportant millor que el rang més ampli i que en part pot ser que haguem trobat indicis d'un màxim local de la relació de les dues variables.

Per seguir recollirem els millors valors de les dues execucions anteriors i mirarem de centrar-los en un sol dataset per treballar-lo. Considerarem que els millors valors per nosaltres son a partir del tcr 25.

```
#bestvals <- rbind(values[values$tcr > 25, ], valueslimited[valueslimited$tcr > 25, ])
#bestvals <- bestvals[order(-bestvals$tcr), ]
```

Ara presentem l'histograma de les aparicions per phi i per k, intentant cercar encara aquest màxim local que creiem que existeix en aquesta franja.

```
#hist(bestvals$k)
#hist(bestvals$phi)
```

Veient els resultats obtinguts i observant les aparicions i els valors de phi i k sobre el dataset bestvals s'ha decidit llençar una 3a tanda focalitzant els rangs per k i phi als valors d'entre [0.20,0.30] per k i de [1.8,2.8] per phi.

### Focalització del valor de k.

Carreguem els resultats.

Realitzem el mateix procediment que amb els valors anteriors. Cerquem el TCR i recollim els millors valors dins del dataframe de bestvalues.

A partir d'aquest moment amb totes les dades més bones que tenim fins ara, és a dir amb els màxims amb els que treballem per phi i per k, si fem la mitjana de k i mirem el millor valor per tcr podem veure :

Que la mitjana dels k millors valors per k és el mateix valor que el màxim tcr trobat. Per tant fixem el valor de k en : 0.236267.

### Exemple de funció K

En el següent gràfic la grandària dels punts estipula quant de gran és l'error no desitjat, és a dir, quan un correu considerat **HAM es filtra com SPAM**. Als eixos hi podem veure els valors de phi i k utilitzats per a la validació. El percentatge de correus utilitzats sobre els 200 correus totals és d'entre 5% i 15% i la selecció d'aquest valor és aleatòria.

```
#head(results)

# De moment la millor opció.

#scatter3D(valores$phi, valores$k, valores$tcr, phi = 0, bty = "g", type = "h", ticktype = "detailed",
#d = ggplot(valores,aes(phi, k, fill=tcr)) + ggtitle("Plot of 100 values") + xlab("PHI") + ylab("K") d

#firstValues <- head(valores, 30)

#ggplot(data = valores, aes(x = phi, y = k)) + geom_tile(aes(fill = tcr))

#grid.arrange(p1,p2,ncol=2)
#heatmap(data.matrix(valores))

#radius <- sqrt(valores$tcr/pi)
```

```
#symbols(valores$phi, valores$k, circles = radius, inches = 0.1, fg = "white", bg = "red", main = "Size  
#plot(valores$tcr~sort(valores$k), type="l")  
#line(valores$tcr~sort(valores$phi), col="red")
```

## Referències

- R graphics