

# spamizer

*Marc Sànchez, Francesc Xavier Bullich, Gil Gassó*

*5/8/2019*

// TODO : Posar el projecte al github.

# 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 :

## Aplicació.

En aquest apartat s'es

## Tecnologies escollides.

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

## Manual de l'aplicació.

## 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 :

**Càlcul del TCR ()**

```
#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", "trcv")
head(values)

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

**Estudi de les variables PHI i K.**

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.

## Anàlisi de la PHI i la K.

### PHI

Si tenim en compte el què representa els 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 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.

### K

Quan apliquem el suavitzat hem de tenir en compte que donats el bag of words de ham i el de spam, què passa si la paraula no existeix? doncs que el valor de les multiplicacions serà 0 i farà que si una paraula no existeix aquesta paraula 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 que les aparicions no computessin tant.

Limitarem els valors de k en un rang de (0 - 3].

### 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$trcv, phi = 0, bty = "g", type = "h", ticktype = "detailed")

#d = ggplot(valores,aes(phi, k, fill=trcv)) + ggtitle("Plot of 100 values") + xlab("PHI") + ylab("K")

#firstValues <- head(valores, 30)

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

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

#radius <- sqrt(valores$trcv/pi)
#symbols(valores$phi, valores$k, circles = radius, inches = 0.1, fg = "white", bg = "red", main = "Size")

#plot(valores$trcv~sort(valores$k), type="l")
#line(valores$trcv~sort(valores$phi), col="red")
```

## Referències

- R graphics