Scénario: *Text mining*, exploration statistique d'un corpus de courriels

Résumé

Exploration statistique d'un volume de données textuelles important afin de définir les caractéristiques des pourriels (spam). Cette analyse nécessite la mise en œuvre successive et donc l'apprentissage approfondi des principales méthodes de statistique mutidimensionnelle : ACP, AFC, MDS, classification non supervisée. Le scénario d'analyse proposé vise à la représentation la plus explicite des données étudiées et teste également une approché spécifique pour la fouille de texte et les matrices très creuses, la factorisation d'une matrice non-négative (NMF).

1 Introduction

1.1 Objectif

Cette étude est un exemple d'*analyse textuelle* d'un corpus de documents, ici des courriels. Une telle analyse est classiquement basée sur la fréquence d'une sélection de mots. L'objectif est de mieux appréhender la structure particulière de ces données avant d'aborder un autre travail de fouille de données ou *d*ata mining. L'objectif de discrimination ou classification supervisée sera ensuite de construire un *d*étecteur de pourriels (spams) personnalisé c'est-à-dire adapté au contenu spécifique de la boîte d'un internaute. Il s'agit en fait d'un modèle susceptible de prévoir la *q*ualité d'un message reçu en fonction de son contenu. Le déroulement de cette étude est évidemment marqué par le type particulier des données mais celle-ci peut facilement se transposer à d'autres types de données textuelles ou analyse du contenu : livres, pages web, discours politiques, réponses ouvertes dans des questionnaires... les exemples sont nombreux en sciences humaines, marketing lorsqu'il s'agit d'estimer des scores, par exemple, de satisfaction de clientèle. Les données se caractérisent généralement par des matrices très creuses c'est-à-dire comportant beaucoup de 0s.

1.2 Données

George, ingénieur chez HP dans le département *Computer Science* a recueilli un échantillon de messages électroniques dans chacun desquels il a évalué le nombre d'occurrences d'une sélection de mots et caractères. Les variables considérées sont, dans une première partie, des rapports : nombre d'occurrences d'un mot spécifique sur le nombre total de mots ou nombre d'occurrences d'un caractère sur le nombre de caractères du message avant d'être, dans une deuxième partie, des indicatrices ou facteurs : présence / absence de mots ou ensemble de caractères. Il a également considéré trois variables prenant en compte la casse (majuscule / minuscule) des caractères et une dernière qualitative binaire indiquant le classement qu'il a fait de chaque message : `spam` ou `Nsp`. Les variables d'occurrences sont décrites dans le tableau 1, celles associées à la casse dans le tableau 2. Ces données sont publiques, elles servent régulièrement de *b*enchmark pour la comparaison de méthodes d'apprentissage machine :

Frank, A. ; Asuncion, A. (2010). UCI Machine Learning Repository. Irvine, CA : University of California, School of Information and Computer Science. `http://archive.ics.uci.edu/ml`

Ce sont donc finalement 58 variables qui sont observées sur 4601 messages dont 1813 pourriels (spams). La variable binaire `Spam` est présente à titre illustratif, elle est toujours considérée en supplémentaire dans ce travail exploratoire préliminaire.

Le tableau 1 liste 54 variables exprimant soit :

- le rapport du nombre d'occurrence d'un mot (resp. de caractères) sur le nombre total de mots (de caractères) dans un message,
- soit la présence ou non de ce mot (resp. caractère) dans le message,
- des numéros (85...) qui sont ceux de bureau, téléphone, code postal de George.

La tableau 2 liste 4 variables dont celle dénombrant le nombre de lettres majuscules.

2 Préparation des données

TABLE 1 – *Les colonnes contiennent successivement le libellé de la variable, le mot ou ensemble de caractères concernés, le libellé des modalités Présence / Absence utilisées après recodage.*

Variable	Mot ou Carac.	Modalités P/A	Variable	Mot ou Carac.	Modalités
make	make	make / Nmk	X650	650	650 / N65
address	address	addr / Nad	lab	lab	lab / Nlb
all	all	all / Nal	labs	labs	labs / Nls
X3d	3d	3d / N3d	telnet	telnet	teln / Ntl
our	our	our / Nou	X857	857	857 / N87
over	over	over / Nov	data	data	data / Nda
remove	remove	remo / Nrm	X415	415	415 / N41
internet	internet	inte / Nin	X85	85	85 / N85
order	order	orde / Nor	technology	technology	tech / Ntc
mail	mail	mail / Nma	X1999	1999	1999/ N19
receive	receive	rece / Nrc	parts	parts	part / Npr
will	will	will / Nwi	pm	pm	pm / Npm
people	people	peop / Npp	direct	direct	dire / Ndr
report	report	repo / Nrp	cs	cs	cs / Ncs
addresses	addresses	adds / Nas	meeting	meeting	meet/Nmt
free	free	free / Nfr	original	original	orig / or
business	business	busi / Nbs	project	project	proj / Npj
email	email	emai / Nem	re	re	re / Nre
you	you	you / Nyo	edu	edu	edu / Ned
credit	credit	cred / Ncr	table	table	tabl / Ntb
your	your	your / Nyr	conference	conferenc	e conf / Ncf
font	order	font / Nft	CsemiCol	;	Cscl / NCs
X000	000	000 / N00	Cpar	(	Cpar / NCp
money	money	mone/ Nmn	Ccroch	[	Ccro / NCc
hp	hp	hp / Nhp	Cexclam	!	Cexc / NCe
hpl	hpl	hpl / Nhl	Cdollar	$	Cdol / NCd
george	george	geor / Nge	Cdiese	#	Cdie / NCi

TABLE 2 – *Liste de 4 variables, de leur libellé et des modalités après recodage.*

Code variable	Libellé	Modalités
Spam	Type de message pourriel ou non	Spam / Nsp
CapLM	Nombre moyen de capitales par mot	Mm1 / Mm2 / Mm3
CapLsup	Nombre max de capitales par mot	Ms1 / Ms2 / Ms3
CapLtot	Nombre totale de lettres capitales	Mt1 / Mt2 / Mt3

2.1 Logiciel

Le logiciel qui semble le plus pratique à utiliser pour cet enchaînement de traitements est R complété par la librairie FactoMineR développée à Agro-Campus de Rennes : Husson, Josse et Lê `http://factominer.free.fr` et celle NMF de Gaujoux et Seoighe (2010)[1].

2.2 Lire les données

Les données sont disponibles dans le fichier `spam.dat` du répertoire "data". Les lire avec les commandes suivantes :

```
spam=read.table("spam.dat",header=TRUE)
spam[,1]=as.factor(spam[,1])
```

2.3 Description élémentaire

Etudier les distributions des variables, celles-ci sont toutes dissymétriques et comportent beaucoup de "0" (matrice de données creuses. L'analyse en tant que variables quantitatives nécessite des transformations ; justifier.

```
Lspam=data.frame("spam"=spam[,1],log(1+spam[,2:58]))
```

3 Approche "quantitative"

3.1 Calcul de l'ACP

Ce sont d'abord les variables quantitatives qui sont étudiées pour tenter de caractériser les spams. Comparer et commenter les résultats obtenus.

```
library(FactoMineR)
res.pca=PCA(spam,scale.unit = FALSE,quali.sup=1)
res.pca1=PCA(spam,scale.unit = TRUE,quali.sup=1)
res.pca=PCA(Lspam,scale.unit = FALSE, quali.sup=1)
res.pca=PCA(Lspam,scale.unit = TRUE,quali.sup=1)
```

Que dire de la transformation par logarithme ? de la réduction ? Quelle analyse est préférable ? Combien d'axes ?

```
barplot(res.pca$eig[,1],main="Eigenvalues",
          names.arg=1:nrow(res.pca$eig))
plot(res.pca,choix="ind",habillage=1,
          label="quali.sup",cex=0.5)
plot(res.pca,choix="var")
dimdesc(res.pca,axes=c(1,2))
```

Les graphiques sont difficiles à lire mais, à l'aide du type de résultats de la dernière commande, tâcher une première interprétation des axes retenus. Sur ces graphiques, pouvez vous déjà caractériser les deux types de messages, courriels ou pourriels ?

3.2 Classification des variables

Ces résultat n'est pas satisfaisants, notamment à cause du nombre de variables. Une classification de celles-ci pourrait aider à l'interprétation.

```
dist.var<-as.dist(1-cor(Lspam[2:58])**2)
clas.var<-hclust(dist.var,method="ward")
plot(clas.var)
plot(clas.var$height[56:40])
```

Commenter les choix adoptés. D'autres sont possibles et doivent être essayés ; donnent-ils des résultats plus pertinents ? Combien de classes et pourquoi ? Retrouve-t-on des éléments d'interprétation des variables dans l'ACP ?

```
rS = cor(Lspam[2:58])
dS2=sqrt(1-rS**2)
dN=dimnames(Lspam[2:58])[[2]]
mdspam= cmdscale(dS2, k=2)
plot(mdspam, type="n", xlab="", ylab="",main="")
text(mdspam,dN)
abline(v=0,h=0)
```

Représentation finalement très similaire à celle de l'ACP ! Pourquoi ?

Représentation de 4 classes de variables dans les coordonnées du MDS

```
classes <- cutree(clas.var,k=4)
sort(classes)
names(classes[classes==2]) #variables de la classe 2
coul = classes
plot(mdspam, type="n", xlab="Dimension 1",
                ylab="Dimension 2", main="CAH euclid")
text(mdspam,dN,col=coul)
```

Retrouve-t-on des éléments d'interprétation des variables dans l'ACP ?

4 Approche "qualitative"

Changement de stratégie, en considérant les aspects qualitatifs des variables : présence / absence, d'un mot ou caractère plutôt que les comptages. De même les variables concernant le nombre de lettres majuscules sont recodées en trois classes.

4.1 Recodage

C'est souvent la partie la plus fastidieuse du travail : recoder en classe, c'est-à-dire transformer en facteur, chaque variable. Le point important est de donner à chaque modalité un identificateur suffisamment explicite pour que les sorties graphiques soient lisibles et ce d'autant plus qu'il y a de variables à traiter. Les commandes suivantes on été utilisées pour l'obtention du fichier `spamq.dat`.

```
# recodage en présence / absence des variables
spamq=data.frame(matrix(as.factor(as.matrix
                          (spam[,2:55]>0)),ncol=54))
# Renommage des niveaux des facteurs (juste un exemle)
# Chaque variable est renommée
# présence du mot "make" et "Nmk" absence de celui-ci
make=factor(spamq[,"make"], c(TRUE, FALSE),
                        labels=c("make", "Nmk"))
# ... toutes les variables
```

```
# Comptages de majuscules en 3 classes
CapLMq=cut(spam[,"CapLM"],breaks=quantile
      (spam[,"CapLM"], probs = seq(0, 1, 1/3)),
      labels = c("Mm1","Mm2","Mm3"),
      include.lowest = TRUE)
```

Enfin, toutes variables sont regoupées dans la même base et sauvegardées. Il suffit de relire le fichier :

```
spam.quali=read.table("spamq.dat")
summary(spam.quali)
```

4.2 AFCM

La variable supplémentaire `spamf` est cette fois en fin de fichier (58). deux autres variables redondantes sont également en variable supplémentaire. Après le calcul de l'AFCM, une kyrielle de graphiques peuvent être construits. Certains sont proposés mais d'autres sont sans doute plus pertinents pour arriver à représenter au mieux la structure des données et expliquer la "nature" des spans.

```
afc=MCA(spam.quali,quali.sup=c(32,34,58))
plot.MCA(afc,invisible=c("ind"),col.var="blue")
# avec un zoom
plot.MCA(afc,invisible=c("ind"),col.var="blue",
                    xlim=c(-1,1),ylim=c(-1,1))
# les messages en couleur
plot(afc$ind$coord,type="p",pch=".",cex=2,
           col=as.integer(spam.quali[,58]),
           xlim=c(-1,1),ylim=c(-1,1))
```

La dispersion des message est plus encourageante qu'avec l'ACP mais que dire d'une possible discrimination linéaire entre pourriels et courriels ?

4.3 Classification des modalités

Comme les modalités sont très nombreuses, une classification de celles-ci va aider à l'interprétation.

```
dist.mod=dist(afc$var$coord, method="euclidean")
hclusmod=hclust(dist.mod,method="ward")
plot(hclusmod)
plot(hclusmod$height[112:100])
hclasmod = cutree(hclusmod,k=4)
clas.mod=kmeans(afc$var$coord, 4)
kclasmod=clas.mod$cluster
# comparaison des classes entre CAH et k-means
table(hclasmod,kclasmod)
```

Que dire de la stabilité des classes ?

```
plot.MCA(afc,invisible=c("ind"),
           col.var=as.integer(clas.mod$cluster))
plot(afc$ind$coord,type="p",pch=".",cex=2,
           col=as.integer(spam.quali[,58]))
```

Lister les modalités des variables par classe. Quelles co-occurrences de quels mots/caractères caractérisent principalement les spams. Quels sont les messages indifférentiables ? Que suggérer à Georges pour améliorer son détecteur de pourriel ? Comment éviter que vos messages ne soient "manger" par les anti-spams ?

5 Approche par NMF

Les données quantitatives sont reconsidérées mais en intégrant le caractère essentièlement "creux" de la matrice des données. Cette situation couramment répandue a suscité une nouvelle forme d'analyse dite Non Negativ Matrix Factorization (NMF) dont le principe est de rechercher deux matrices de faible

rang r de telle sorte que leur produit approche au mieux les valeurs observées. Contrairement à L'ACP où les facteurs sont recherchés orthogonaux 2 à 2, cette méthode impose la contrainte de non négativité des matrices pour construire les facteurs de la décomposition. Ces facteurs ne permettent plus de représentation comme en ACP ou en MDS mais au moins une classification non supervisée tant des lignes que des objets lignes et colonnes de la matrice.

Cette approche est testée sur les données de spam pour en comparer les résultats obtenus.

5.1 Factorisation non négative

La librairie `NMF` (Gaujoux et Seoighe, 2010)[1] de R propose plusieurs versions de l'algorithme de factorisation, principalement *Multiplicative update algorithms* et *Alternate least Square (ALS)*, adaptées à deux fonctions perte possibles : divergence de Kullback-Leibler (KL) ou moindres carrés. Attention, les choix : fonction objectif, algorithme, du rang des matrices, influencent fortement les résultats obtenus qui se résument principalement à ces classifications construites sur les facteurs de la décomposition.

La librairie `NMF` (Gaujoux et Seoighe, 2010)[1] a été réalisée et publiée avec pour premier objectif le traitement des données génomiques dans l'environnement de Bioconductor, donc de la Bioinformatique. C'est sans doute ce qui explique l'utilisation systématique de la classification ascendante hiérarchique et des représentations graphiques par *heatmap*.

Chargement de la librairie et identification des algorithmes disponibles. Plusieurs initialisation sont possibles ; seule celle aléatoire est utilisée.

```
library(NMF)
nmfAlgorithm()
nmfAlgorithm("brunet")
nmfAlgorithm("lee")
nmfAlgorithm("snmf/l")
nmfAlgorithm("snmf/r")
```

Identifier la fonction perte, seuls les deux derniers algorithmes sont issus de l'ALS.

Les données quantitatives initiales sont reprises en compte. Attention, certes les données sont bien creuses mais mes variables s'expriment dans des unités et donc avec des variances très différentes. Une forme de normalisation est nécessaire à un niveau ou un autre.

```
creux=as.matrix(spam[,1:57])
classe=spam[,58]
creux=cbind(log(1+creux[,1:54]),log(creux[,55:57])/2)
boxplot(creux)
# souci pour la suite :
sum(apply(creux,1,sum)==0)
# 3 messages sont devenus tout à 0
# suppression
ident=apply(creux,1,sum)!=0
creux=creux[ident,]
classe=classe[ident]
```

Comparer les méthodes en exécutant pour chacune d'entre elles 10 factorisations de rang 5. Les exécutions sont répétées car la convergence locale dépend de l'initialisation.

```
res.multi.method=nmf(creux, 5,nrun=10,
  list("brunet","lee","snmf/l","snmf/r"),
        seed = 111, .options ="t")
compare(res.multi.method)
consensusmap(res.multi.method,hclustfun="ward")
```

Plusieurs critères de comparaison sont proposés. Lequel choisir ? Pourquoi ?

Choix du rang des matrices de la décomposition.

```
estim.r=nmf(creux,2:7,method="snmf/l",
   nrun=10,seed=111)
plot(estim.r)
consensusmap(estim.r)
```

Utiliser les résultats précédents pour déterminer un rang "optimal".

Une fois méthode et rang déterminés, itérer plusieurs fois l'exécution pour en déterminer une "meilleure".

```
nmf.spam=nmf(creux,5,method="snmf/l",
```

```
   nrun=30,seed=111)
```

Extraction des résulats numériques.

```
summary(nmf.spam)
s=featureScore(nmf.spam)
summary(s)
s=extractFeatures(nmf.spam)
str(s)
# les matrices de facteurs
w=basis(nmf.spam)
h=coef(nmf.spam)
```

Production des graphiques associés aux matrices `w` et `h` de la factorisation. La classe des messages (spam ou pas spam) est ajoutée dans la représentation des lignes.

```
basismap(nmf.spam,annRow=classe,hclustfun="ward")
coefmap(nmf.spam,hclustfun="ward")
```

Remarquer la plus ou moins bonne séparation des pourriels et courriels dans les classes de la CAH.

5.2 Classification des variables

Comme c'est logique, il semble bien que le dendrogramme produit dans les cartes précédentes soient directement issues des classifications ascendantes hiérarchiques calculées à partir des distances euclidiennes entre les lignes de `w` et les colonnes de `h`.

```
dist.mod=dist(t(h), method="euclidean")
hclusmod.h=hclust(dist.mod,method="ward")
plot(hclusmod)
plot(hclusmod$height[56:46])
```

Les variables quantitatives étant conservées en l'état contrairement à la section précédente (AFCM), les résultats sont nécessairement différents. Ce n'est pas d'usage dans la communauté bioinformatique mais la classification des variables est représentable dans les coordonnées d'un MDS, ou, ce serait équivalent, dans les composantes d'une ACP des "facteurs" de la NMF qui ne sont pas orthogonaux deux à deux.

```
mdspam= cmdscale(dist.mod, k=2)
dN=dimnames(h)[[2]]
plot(mdspam, type="n", xlab="", ylab="",main="")
text(mdspam,dN)
abline(v=0,h=0)
```

Les mots ou lettres à plus forte occurrence prennent encore trop d'importance, les facteurs sont donc réduits.

```
dist.mod=dist(scale(t(h)), method="eucl")
mdspam= cmdscale(dist.mod, k=2)
hclusmod.h=hclust(dist.mod,method="ward")
plot(hclusmod.h)
plot(hclusmod.h$height[56:46])
hclasmod = cutree(hclusmod.h,k=4)
plot(mdspam, type="n", xlab="", ylab="",main="")
text(mdspam,dN,col=hclasmod)
abline(v=0,h=0)
```

Il n'est pas possible comme en ACP ou AFCM de mettre en relation les deux représentations des lignes et colonnes, individus et variables de la matrice factorisée. Cela peut être fait de façon détournée à l'aide d'une *heatmap* qui peut intégrer deux classifications obtenues par ailleurs.

```
# classificaiton des messages à partir de w
dist.mod=dist(scale(w), method="euclidean")
hclusmod.w=hclust(dist.mod,method="ward")
plot(hclusmod.w)
# intégration des deux classifications
aheatmap(creux,Rowv=hclusmod.w,
   Colv=hclusmod.h,annRow=classe,
   annCol=as.factor(hclasmod))
```

Deux classifications et deux représentations se trouvent en concurrence.

- La première utilise les composantes de l'AFCM des modalités présence / absence fournies pas la SVD du tableau disjonctif complet normalisé,

- La deuxième utilise la matrice H des coefficients de la factorisation en matrices non négatives.

Difficile de savoir laquelle est la plus pertinente ou informative. La réponse n'en peut être apportée qu'au cas par cas et par un commanditaire de l'étude spécialiste du domaine, Biologie, marketing, maintenance,... concerné.

Références

[1] Renaud Gaujoux et Cathal Seoighe, *A flexible R package for nonnegative matrix factorization*, BMC Bioinformatics **11** (2010), n° 1, 367, http://www.biomedcentral.com/1471-2105/11/367.