Scénario: de la SVD à l'ACP

Résumé

Décomposition en valeurs singulières (SVD) d'une matrice rectangulaire comme introduction à l'Analyse en Composantes Principales ; illustration sur des données fictives avec R puis avec le package FactoMineR.

Avertissement

- Les différents travaux et analyses proposés tout au long de ces documents sont largement explicités. Les commandes en R ou SAS toutes fournies. L'important n'est pas de trouver la bonne syntaxe des commandes ni de finir au plus vite mais de réfléchir sur les méthodes, leurs conditions d'applications, les résultats obtenus. L'apprentissage de ces logiciels et de leur programmation est un autre cours.
- Il est possible de directement copier les commandes de l'affichage de ce texte vers une fenêtre d'édition mais **attention**
  - charger le fichier pdf et l'ouvrir dans acrobat car la visualisation dans un navigateur perturbe beaucoup l'opération "copier/coller".
  - certains caractères, "modifiés" par les normes d'affichage réservent des surprises, notamment le caractère " ' " (quote) n'a pas la même fonction que le caractère " ' " (apostrophe). Il est vivement conseillé d'ouvrir une fenêtre d'édition de texte (xemacs, kile, notepad...), pas de traitement de texte (ni word, ni open office), qui contiendra les différentes commandes a faire exécuter.
- Il est aussi possible de prendre le temps d'entrer les commandes au clavier, cela laisse le temps de réfléchir !

1 SVD et recherches d'éléments propres

Cette section se propose d'illustrer les principaux résultats d'algèbre linéaire utiles en exploration statistique multidimensionnelle. Ceci concerne les valeurs et vecteurs propres de matrices symétriques en lien avec la décomposition en valeur singulière ou SVD d'une matrice rectangulaire $n \times p$ pour en faire une approximation par une matrice de mêmes dimensions mais de rang inférieur.

1.1 SVD avec métrique usuelle

Décomposition en valeurs singulières d'une matrice rectangulaire relativement à des métriques classiques définies par la matrice identité.

```
# Génération d'un matrice n x p
X=matrix(runif(100),20,5)
# SVD
res=svd(X)
# Valeurs singulières
res$d
# Vérifier l'orthonormalité des vecteurs
t(res$u)%*%res$u
t(res$v)%*%res$v
# Vérifier la reconstruction de X
# à l'erreur machine près
X-res$u%*%diag(res$d)%*%t(res$v)
```

Comparer ci-dessous les valeurs propres des matrices $\mathbf{X}'\mathbf{X}$ et $\mathbf{X}\mathbf{X}'$. Que dire des dimensions, du rang de ces matrices, de la multiplicité de la valeur propre nulle ?

Comparer avec les valeurs propres et les valeurs singulières de $\mathbf{X}$.

```
# Valeurs et vecteurs propres
dec1=eigen(t(X)%*%X)
dec2=eigen(X%*%t(X))
dec1$values
dec2$values
sqrt(dec1$values)
U=dec2$vectors
V=dec1$vectors
# Orthonormalité des vecteurs
t(V)%*%V
t(U)%*%U
```

Vérifier la bonne cohérence des dimensions de ces vecteurs puis comparer les vecteurs singuliers à droite et à gauche de $\mathbf{X}$ avec les vecteurs propres de ces

matrices. D'où viennent les différences ?

```
V-res$v
U[,1:5]-res$u
```

Vérifier que les premiers termes de la SVD sont la meilleure approximation de $\mathbf{X}$ par une matrice de rang inférieure.

```
# approximation de rang 4
Xhat=res$u[,1:4]%*%diag(res$d[1:4])%*%t(res$v[,1:4])
# calcul de la norme ||X-Xhat||^2
sum((X-Xhat)**2)
```

Comparer cette norme avec les carrés des valeurs singulières, que vaudrait cette norme avec une approximation de rang 3 ?

Retrouver la matrice des vecteurs singuliers à droite à partir de celle des vecteurs singuliers à gauche et réciproquement.

```
# Vecteurs singuliers à gauche:
Ug=res$u
# vecteurs singuliers à droites
# calculés à partir de U
Vd=t(X)%*%Ug
# Vérifier l'orthogonalité
t(Vd)%*%Vd
# mais pas l'orthonormalité
# Que vaut la diagonale ?
diag(t(Vd)%*%Vd)-res$d**2
# Il faut normaliser
Vd=Vd%*%diag(1/res$d)
t(Vd)%*%Vd
# Vérifier
Vd-res$v
```

Retrouver de façon analogue les vecteurs singuliers `Ug` à partir de ceux `Vd` en prémultipliant par la matrice `X`.

1.2 SVD généralisée

Les calculs précédents concernent des matrices (applications linéaires) dans des espaces euclidiens munis de métriques "usuelles" définies par les matrices identités. Pour calculer une SVD généralisée par rapport à des métriques quelconques, celà nécessite d'introduire des changements de métriques car la SVCD généralisée n'est pas prévue dans les logiciels.

Comme en ACP, on considère une métrique dans l'espace vectoriel $\mathbb{R}^p (p = 5)$ définie par une matrice $\mathbf{M}$ symétrique définie positive et une métrique de matrice diagonale $\mathbf{D}$ sur l'expace vectoriel $\mathbb{R}^n (n = 20)$. La SVD de $\mathbf{X}$ relativement à $\mathbf{M}$ et $\mathbf{D}$ est calculée à partir de celle classique de $\mathbf{D}^{1/2}\mathbf{X}\mathbf{M}^{1/2}$ relativement aux métriques de matrice identité ; $\mathbf{M}^{1/2}$ est la racine carrée de $\mathbf{M}$.

En effet, si

$$\mathbf{D}^{1/2}\mathbf{X}\mathbf{M}^{1/2} = \mathbf{U}\mathbf{\Lambda}\mathbf{V}' \quad \text{avec} \quad \mathbf{U}'\mathbf{U} = \mathbf{I} \quad \text{et} \quad \mathbf{V}'\mathbf{V} = \mathbf{I}$$

alors en prémultipliant à gauche et multipliant à droite :

$$\mathbf{X} = \mathbf{D}^{-1/2}\mathbf{U}\mathbf{\Lambda}\mathbf{V}'\mathbf{M}^{-1/2} = \left(\mathbf{D}^{-1/2}\mathbf{U}\right)\mathbf{\Lambda}\left(\mathbf{M}^{-1/2}\mathbf{V}\right)'$$

et $\mathbf{D}^{-1/2}\mathbf{U}$ définit les vecteurs $\mathbf{D}$-orthonormés singuliers à gauche et $\mathbf{M}^{-1/2}\mathbf{V}$ ceux à droite de la SVD$(\mathbf{X}, \mathbf{M}, \mathbf{D})$ car ils vérifient bien :

$$\left(\mathbf{D}^{-1/2}\mathbf{U}\right)'\mathbf{D}\left(\mathbf{D}^{-1/2}\mathbf{U}\right) = \mathbf{I} \quad \text{et} \quad \left(\mathbf{M}^{-1/2}\mathbf{V}\right)'\mathbf{M}\left(\mathbf{M}^{-1/2}\mathbf{V}\right) = \mathbf{I}.$$

Construction des différentes matrices.

```
# Matrice M symétrique définie positive
H=matrix(rnorm(25),5,5)
M=t(H)%*%H
# Décompostion de M
s=eigen(M);l=s$values;v=s$vectors
l; v
# Racine de M
Mr=v%*%diag(sqrt(l))%*%t(v)
# Inverse de racine de M
```

```
Mi=v%*%diag(1/sqrt(l))%*%t(v)
# vérifications
Mr%*%Mi;M-Mr%*%Mr
# Construction de D
w=runif(20); w=w/sum(w)
D=diag(w)
Dr=diag(sqrt(w)); Di=diag(1/sqrt(w))
```

Construction de la SVD$(\mathbf{X}, \mathbf{M}, \mathbf{D})$ et vérifications.

```
# SVD de l'image de X (changement de métrique)
s=svd(Dr%*%X%*%Mr)
U=s$u;L=s$d;V=s$v
# Nouveaux vecteurs singuliers
Ud=Di%*%U
Vm=Mi%*%V
# Vérifications
t(Vm)%*%M%*%Vm
t(Ud)%*%D%*%Ud
X-Ud%*%diag(L)%*%t(Vm)
```

2 Application de la SVD à l'ACP

L'objectif est de bien comprendre quels sont les résultats / matrices produites par les différentes façons d'aborder l'analyse en composantes principales.

2.1 Données

Les données sont celles de l'exemple introduction à l'ACP : les notes en maths, français, physique et anglais de 9 lycéens virtuels.

```
# Matrice des données
note=matrix(c(6,6,5,5.5,8,8,8,8,
6,7,11,9.5,14.5,14.5,15.5,15,
14,14,12,12.5,11,10,5.5,7,
5.5,7,14,11.5,13,12.5,8.5,9.5,
9,9.5,12.5,12),nrow=9,byrow=TRUE)
note=data.frame(note)
```

```
nomi=c("jean","alai","anni","moni",
 "didi","andr","pier","brig","evel")
nomv=c("Math","Phys","Fran","Angl")
dimnames(note)[[1]]=nomi
dimnames(note)[[2]]=nomv
# Vérification
note
```

Statistiques élémentaires :

```
summary(note)
boxplot(note)
X=as.matrix(note)
# Matrice des variances covariances
var(X)
# Matrice des corrélations
cor(X)
```

2.2 ACP pas à pas

Tout langage matricielle permet de construire très facilement une ACP. Attention, la formule de variance utilise $(n-1)$ au lieu de n comme diviseur afin de retrouver les résultats des fonctions R. Un logiciel "français" peut donc fournir des résultats légèrement différents.

```
# Matrice centrée
Xb=scale(X,scale=F)
Xb
# Covariance
S=t(Xb)%*%Xb/8
# SVD
res=svd(Xb)
# Matrices des vecteurs propres
U=res$u;V=res$v
# Valeurs propres et variance expliquée
L=res$d*res$d/8; pct=L/sum(L)
L;pct
# Composantes principales
```

```
C=Xb%*%V
```

Construction de représentations graphiques rudimentaires.

```
boxplot(as.data.frame(C))
# Coordonnées "isométriques lignes"
plot(C,type="n")
text(C,nomi)
abline(h=0,v=0)
# Coordonnées "isométriques colonnes"
plot(V[,1]*sqrt(L[1]),V[,2]*sqrt(L[2]),
   type="n")
text(V[,1]*sqrt(L[1]),V[,2]*sqrt(L[2]),
   nomv)
abline(h=0,v=0)
```

2.3 Fonctions R spécifiques

Encore plus d'effort est nécessaire pour produire des graphiques lisibles. Autant utiliser les fonctions R adaptées. D'abord avec la fonction `princomp`.

```
res1.acp=princomp(note)
summary(res1.acp)
plot(res1.acp)
biplot(res1.acp)
```

Combien d'axes faut-il retenir ? Donner une interprétation de chacun d'eux.

Puis avec la fonction `prcomp`.

```
res2.acp=prcomp(note)
summary(res2.acp)
plot(res2.acp)
biplot(res2.acp)
```

Comparer les aides de ces fonctions pour comprendre d'où viennent les différences. Attention pour `princomp`, le nombre de lignes (n) doit être plus grand que le nombres de colonnes p ; ce n'est pas une contrainte pour `prcomp`.

Que sont les différents résultats de ces deux fonctions : `sdev, rotation, center, scale, x` de `prcomp` et `sdev, loadings, center, scores` de `princomp` par rapport aux matrices `U, V, L, C` précédentes.

2.4 Package FactoMineR

Ce package donne accès à la plupart des méthodes factorielles et de classification non supervisée multidimensionnelles. Son utilisation nécessite, si ce n'est déjà fait, une installation préalable par la commande `install.packages("FactoMineR")` ou par l'utilisation des menus de la fenêtre inférieure droite de RStudio.

```
library(FactoMineR)
PCA(note)
```

Ce package fournit de très (un peu trop) nombreux résultats et des graphiques dont le flux n'est pas pas toujours compatible avec RStudio. D'autres options et fonctions sont abordées dans les scénarios suivants.