Majeure *Science des données* 2017-18

UP1 'Statistiques exploratoires et outils mathématiques'

TP n°2 - Analyse en composantes principales : robustesse - *TP 1H30* 19 octobre 2017

Cette séance de 1h30 de TP est consacrée à la validation et stabilité de l'Analyse en Composantes Principales.
La mise en œuvre se fera sur les données de modèles de criminalités [(1)] : fichier crime_data_pca.csv" pour la partie 2.
Pour cela vous aurez besoin de télécharger le fichier sur : https://campus.emse.fr/
Vous devez déposer votre TP par binôme sur campus : soit le(s) script(s) développé(s) + un compte-rendu selon le format proposé sur campus.

Il s'agit :

- De visualiser la qualité de la réduction en fonction des propriétés intrinsèques à l'échantillon
- De comprendre l'enjeu de la validation d'une méthode de traitement de données (plus généralement d'un modèle de prévision)
- De mettre au point un algorithme de Bootstrap pour calculer l'erreur standard d'un estimateur d'un paramètre de l'ACP
- De mettre au point une méthode de validation et de calcul de l'intervalle de confiance sur le nombre d'axes choisi en fonction de l'inertie expliquée par la réduction de dimension en ACP
- De savoir utiliser une toolbox *bootstrap*

Ceci fait suite à la première partie (TP n°1) qui concerne l'évaluation des critères de qualité de la mise en œuvre d'une ACP centrée et d'une ACP normée et de sa qualité de réduction pour les variables et les individus.

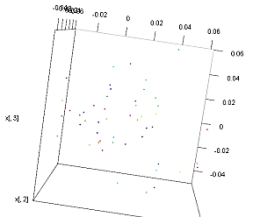
Partie 1 - l'ACP et étude de nuage de point

1. Etude de la forme du nuage initiale et de sa répercussion sur la réduction de dimension

1.1 Nuage isotrope

On veut générer des données pour 3 variables d'un nuage de points proches d'une sphère. Pour cela vous pouvez choisir plusieurs méthodes mais une possibilité est de générer un échantillon de taille *n* des 3 variables X, Y, Z comme 3 vecteurs gaussiens indépendants de loi N(0,1) puis de récupérer le vecteur V à 3 composantes tel que V/||V||. Car la densité f(v) d'un vecteur gaussien est isotrope et dépend que de v.

Vous pouvez visualiser votre nuage par la :
open3d()
plot3d(x[,1], x[,2], x[,3], col = rainbow(n))



Vous devez écrire le script qui génère ces données, puis mettre en œuvre l'ACP
En fonction de la taille de l'échantillon, suivre l'évolution de la matrice de covariance ou de corrélation, la cascade des valeurs propres, ainsi que la qualité de la projection des individus. Interpréter

1.2 Nuage non isotrope

On cherche maintenant à voir sur des formes non isotropes si des corrélations plus fortes entre certaines variables changent l'ACP et comment.
On génère des données pour 3 variables qui sont dans un nuage de points pour lesquelles deux variables sont corrélées fortement. Pour cela vous pouvez choisir plusieurs méthodes mais une possibilité est de générer un échantillon de taille *n* pour 3 variables X, Y,Z puis appliquer une relation linéaire entre les deux premières X et Y , puis entre Y et Z. Vous pouvez modifier votre corrélation entre deux variables en rajoutant un bruit.

Tester l'ACP sur les données générées et en fonction de la taille de l'échantillon suivre l'évolution de la matrice de covariance ou de corrélation, la cascade des valeurs propres, ainsi que la qualité de la projection des individus. Interpréter.

On vous donne le cas de 3 variables générées selon le script suivant :

x <- sort(rnorm(1000))

y <- rnorm(1000)

z <- rnorm(1000) + atan2(x, y)

Que vous donne une telle ACP ? Devez-vous normées les données ?

1.3 Points extrémaux

Dans le cas d'un nuage dont la forme est non isotrope (choisir votre générateur de nuage) vous choisissez soit de rajouter certains points extrémaux pour une ou deux variables (vous allez étirer ainsi votre nuage selon cette direction) soit au contraire ôter des points extrémaux (ou quelques points) pour compacter votre nuage selon quelques variables. Proposer quelques essais d'ACP centrée ou ACP normée sur ces données, qu'observez-vous ? Que pouvez-vous en conclure sur le fait de centrée ou normée les données.

2ère partie

1. Afin de vous familiariser avec la technique du bootstrap : vous devez mettre en place l'estimation de l'erreur standard *se* associée à deux statistiques d'intérêt que sont la moyenne et la médiane selon le principe du bootstrap.

Pour instancier ce problème, on vous propose **l'exemple fourni en cours**: lors d'une expérimentation sur des souris on a tiré au sort 16 souris : 7 reçoivent un traitement alors que les 9 autres reçoivent un placebo. Leurs durées de vie sont mesurées en jours et donnent les deux groupes suivants :

Groupe1 (placebo) 52,10,40,104,50,27,146,31,46
Groupe2 (traitement) 94,38,23,197,99,16 ,141

On cherche à savoir si les deux moyennes (m1, m2) et les médianes (μ1 et μ2) sont significativement différentes.

Pour l'erreur standard *se* associée à la différence entre les deux moyennes (m1 et m2), vous disposez de la formulation analytique de cette erreur standard ; soit, comme étant la racine carré de la somme des carrés de deux erreurs standards associées (se_1 et se_2) à la moyenne de chacun des groupes. La différence associée est donnée par *(m1- m2)/se*. La significativité de cette différence est donc calculable.

Pour l'erreur standard *se* associée à la différence entre les deux médianes (μ1 et μ2), vous ne disposez pas d'une formulation analytique de ces erreurs standards : il est nécessaire de pouvoir calculer cette erreur standard pour chacune des médianes. Plus généralement il n'existe pas de formulation simple pour évaluer la fiabilité de grandeurs autres que les valeurs moyennes, d'où l'intérêt du Bootstrap pour calculer l'erreur standard sur *moyenne* et surtout pour la *médiane.*

1) Calculer pour l'échantillon initial des 2 groupes de souris : les valeurs moyennes, médianes et l'erreur standard de la moyenne, ainsi que l'erreur standard *se* associée à la différence entre les deux moyennes (m1 et m2)
2) Mettre au point une fonction (sous R) qui permet de calculer l'erreur standard de la *moyenne* par bootstrap, puis de la médiane. Chaque fonction doit permettre de :
   - Générer les *B* échantillons *Bootstrapé*s à partir d'un échantillon initial : ce qui consiste à faire un tirage aléatoire parmi cet échantillon avec remise, chaque échantillon Bootstrapé est de même taille que l'échantillon initial
   - Calculer les reliquats de la statistique d'intérêt (*ici moyenne ou médiane)*
   - Puis l'erreur standard de l'estimateur du paramètre (ou statistique d'intérêt)

Vous devez en autre, suivre l'évolution de cet estimateur en fonction du nombre B d'échantillonnage (*dim de B*) choisi et suivre son évolution.

A la fin de cette étape, vous disposer d'un script qui calcule pour la moyenne et l'erreur standard bootstrappée et son suivi graphiquement en fonction de la taille de B.

2. On s'intéresse maintenant à l'erreur standard *se* associée sur un autre paramètre qui est le coefficient de corrélation entre deux variables X et Y.

1) Vous devez retrouver l'algorithme de bootstrap à partir de ce que vous avez fait en 1.

2) et mettre au point une fonction (sous R) qui permet de calculer l'erreur standard du coefficient de corrélation par bootstrap.

Chaque fonction doit permettre de :

- Générer les *B* échantillons Bootstrap à partir d'un échantillon initial : ce qui consiste à faire un tirage aléatoire parmi cet échantillon avec remise, chaque échantillon du Bootstrap est de même taille que l'échantillon initial.
- Calculer les reliquats de la statistique d'intérêt (*soit ici le coefficient de corrélation).*
- Puis l'erreur standard de l'estimateur du paramètre (ou statistique d'intérêt)

3) Vous devez suivre l'évolution de cet estimateur en fonction du nombre B d'échantillonnage (*dim de B*) choisi et suivre son évolution sur un plot par exemple.

4) Appliquer et tester sur les données des deux premières variables du fichier criminalité fournis pour l'ACP. Tracer l'évolution des grandeurs d'intérêt en fonction de la taille de B ? Quelle conclusion faites-vous en fonction du choix du nombre d'échantillonnage B ?

3. Erreur standard en ACP

Lors d'une ACP, le pourcentage de variance expliquée par la première composante est égale à : $\theta = \lambda_1 / \sum_{i=1}^{p} \lambda_i$. De même les *p* vecteurs propres associés à l'ACP posent le problème de la fiabilité également des vecteurs propres. On s'intéresse en particulier au premier vecteur propre u_1 associé à la plus grande valeur propre λ_1. Vous devez formaliser la méthode de Bootstrap sur ces deux grandeurs pour estimer l'erreur standard associée et la mettre en œuvre sous R . Pour cela vous devez disposer d'un script d'ACP (TP1) et de script de bootstrap (partie1). L'application se fera sur le jeu de données de criminalité.

4. Calcul de l'Intervalle de confiance par boostrap-t

1) Mettre en place la procédure du calcul de l'IC du bootstrap- t sur les données précédentes (souris de la partie1) pour le calcul de la moyenne et, sur la moyenne d'un échantillon de type loi normale de dimension n, de moyenne 10 et d'écart type 2.
2) Mettre en place l'algorithme de l'IC par quantile et l'appliquer au coefficient de corrélation. Suivre graphiquement l'évolution de l'IC en fonction de de la taille de B et tracer sa distribution.
3) Enfin on vous propose d'approcher la variabilité associée à chaque valeur propre dans une ACP. Pour cela, on récupère plusieurs valeurs de λ_k (pour k =< p) en répétant B fois la procédure suivante : on effectue un tirage avec remise des observations pour avoir un échantillon de taille *n* d'individus, on lance une ACP à chaque étape , puis on vérifie la règle de Kaiser pour le choix des facteurs : soit un facteur (au sens de l'ACP) est jugé pertinent si le quantile d'ordre 0.05 ($\lambda_k{}^{0.05}$) (borne de l'intervalle de confiance associé) des valeurs propres Bootstrap est supérieur à 1. Si un doute subsiste on vous propose une autre vision du test d'égalité des valeurs propres : un facteur est jugé pertinent si sa valeur propre est significativement supérieure à celle du facteur suivant, cad s'il y a empiètement des IC de chaque valeur propre. Soit un facteur k est sélectionné si $\lambda_k^{0.05} > \lambda_{k+1}^{0.95}$. Mettre en œuvre et tester sur les données.

La mise en œuvre se fera sur les données de criminalités (1) : fichier crime_data_pca.csv

Quelques fonctions utiles : sample() , rbind() ,st()…, boot()

Calcul d'intervalle de confiance et usage de la toolbox boot

Estimation de l'intervalle de confiance IC

Estimateur $\hat{\theta}$ et son erreur standard $se_F(\hat{\theta})$ son IC usuel à 90% est $\hat{\theta} \pm 1.64 se_F(\hat{\theta})$, suppose que $(\theta(F_0) - \hat{\theta})/ se_F(\hat{\theta})$ suit une loi normale centrée réduite.

Deux approximations : 1ère) si l'estimateur du paramètre $\hat{\theta}$ suit une loi normale et la 2ème) que $Z = (\theta(F_0) - \hat{\theta})/ se_F(\hat{\theta})$ suit une loi normale centrée réduite : alors que si $\hat{\theta}$ a une distribution normale, Z suit une loi de Student de degré n-1 . **Avec le Bootstrap** , l'hypothèse de normalité n'est pas nécessaire ce qui permet de calculer l'intervalle de confiance (généralisation de la méthode du test de Student).

Intervalle de confiance par : **B*ootstrap-t*** (*la distribution de Z sur les données*), donné par :

- pour chaque *b* échantillons générés par les *B* tirage de bootstrap
  - calculer pour chaque b $Z^{*b} = \frac{\hat{\theta}^{*b} - \hat{\theta}}{se(\hat{\theta}^{*b})} \times \frac{1}{\sqrt{n}}$ avec $\widehat{\theta}^{*b}$ valeur de estimateur pour échantillon bootsrap X^{*b}
- trier les valeurs de Z^{*b} par ordre croissant
- le αième fractile[1] de la loi de *Z* , est estimé par la valeur $\hat{t}^{\alpha}$ telle que $\alpha = \#\{Z^{*b} \leq \hat{t}^{\alpha}\}/B$ et l'intervalle bootstrap $IC(2\alpha)$:

avec

$$IC_{inf} = \hat{\theta} - Z^{*\left(\frac{B(1-\alpha)}{2}\right)} \times se_F \times \frac{1}{\sqrt{n}}$$

$$IC_{sup} = \hat{\theta} - Z^{*(B\times\alpha/2)} \times se_F \times \frac{1}{\sqrt{n}}$$

OR *: Estimation de l'intervalle de confiance est applicable à des statistiques ponctuelles comme la moyenne d'un échantillon mais non applicable pour des statistiques plus complexes pour lesquelles on ne dispose pas d'estimateur statistique (comme la médiane, un coefficient de corrélation…)*

Méthode par quantile : intervalle de confiance sur les quantilés bootstrap, donné par :

1. pour chaque *b* échantillons générés par les *B* tirage de bootstrap
2. calculer pour chaque *b*, *repliquat bootstrap* $\hat{\theta}^{*b}$

3.trier les valeurs de $\hat{\theta}^{*b}$ par ordre croissant : soit le $\hat{\theta}_B{}^{*(\alpha)}$ est le fractile empirique d'ordre α des $\hat{\theta}^{*b}$, soit la *B*-ième valeur dans la liste ordonnée des *B* reliquats de $\hat{\theta}^{*}$. L'intervalle approximé de niveau (1-2α) est alors :

$$IC = \left[\hat{\theta}^{*\left(\frac{B\times(\alpha)}{2}\right)}; \hat{\theta}^{*\left(\frac{B(1-\alpha)}{2}\right)}\right]$$

Avec $\left(\frac{B\times(\alpha)}{2}\right)$ *est indice de la borne inferieure des répliquats des estimateurs* , idem pour borne supérieure

---

Rappels et supports

- On peut tracer de l'inertie expliquée λ_j par les *j* différentes composantes principales obtenues (*j* allant de *1 à p*) : en utilisant la fonction barplot() *Creates a bar plot with vertical or horizontal bars*

[1] Si B=1000, l'estimation de valeur $\hat{t}^{\%5}$ est la 50ième plus grande valeur de de Z^{*b}

- Soit les nouvelles coordonnées de l'individu *i* sur chacune des composantes soit C_i^j : la qualité de la projection de l'individu Q_i^k, (*k* étant le nombre de composantes principales retenues) définie par : $Q_i^k = \frac{\sum_{j=1}^{k}(C_i^k)^2}{\sum_{j=1}^{p}(C_i^p)^2}$
- La contribution de l'individu *i* à l'inertie de l'axe factoriel *j*, est définie par $\gamma_i^j = \frac{\frac{1}{n}(C_i^j)^2}{\lambda_j}$
- Une composante principale, est reliée à une variables initiale X^j en calculant un coefficient de corrélation linéaire entre une composante c et une variable j définit par : $r(c, X^j) = \sqrt{\lambda}\, u_j$ pour ACP normée (u_j coordonnées du vecteur u pour la *j* variable et λ la valeur propre associée à la composante c $r(c, X^j) = \frac{\sqrt{\lambda}}{\sqrt{Var(Xj)}} u_j$ pour ACP centrée
- les fonctionnalités existantes sous R pour vous familiariser avec ces toolboxes, comme : princomp, boot et des fonctions comme apply, quantile , replicate, … onction dudi.pca ()

Dans la : library(ade4) fonction dudi.pca ().

---

(1) Les données test dans fichier : crime_data_pca.csv

Description: These data are crime-related and demographic statistics for 47 US states in 1960. The data were collected from the FBI's *Uniform Crime Report* and other government agencies to determine how the variable crime rate depends on the other variables measured in the study.

Number of cases: 47

Variable Names:

1. R: Crime rate: # of offenses reported to police per million population
2. Age: The number of males of age 14-24 per 1000 population
3. S: Indicator variable for Southern states (0 = No, 1 = Yes)
4. Ed: Mean # of years of schooling x 10 for persons of age 25 or older
5. Ex0: 1960 per capita expenditure on police by state and local government
6. Ex1: 1959 per capita expenditure on police by state and local government
7. LF: Labor force participation rate per 1000 civilian urban males age 14-24
8. M: The number of males per 1000 females
9. N: State population size in hundred thousands
10. NW: The number of non-whites per 1000 population
11. U1: Unemployment rate of urban males per 1000 of age 14-24
12. U2: Unemployment rate of urban males per 1000 of age 35-39
13. W: Median value of transferable goods and assets or family income in tens of $
14. X: The number of families per 1000 earning below 1/2 the median income