Лабораторна робота №3
з дисципліни ”Чисельні методи у статистиці”

Горбунов, 5 курс, ”Прикладна та теоретична статистика”

19 травня 2022 р.

Вступ.

Хід роботи.

Постановка задачі.

Взяти дані Бемпеса по горобцях і побудувати класифікацію ознак за ступенем скорельованості (по суті, це відповідає пошуку кореляційних плеяд - процедурі, що часто застосовується в зоології) і одержати оцінки значущості одержаних груп. Метод кластеризації (single linkage, complete linkage, метод Варда, ще якийсь) – на ваш розсуд, але обґрунтуйте, чому обрали саме такий метод.

Короткий огляд таблиці.

Запопронована для аналізу таблиця складається з даних Гермона Бемпеса про горобців (Passer domesticus), частина з яких вижила під час потужного шторму в Англії, а частина з них загибла. Таблиця складається з таких змінних:

1. Стать (sex), m = самець; f = самка
2. Вік, лише для самців (age), a = дорослий, y = молодий
3. Виживання (surv), TRUE якщо вижив, FALSE якщо загинув
4. Загальна довжина (у мм) ”З кінчика дзьоба до кінчика хвоста” (tl)
5. Розмах крила (у мм) (ae)
6. Вага (у г) (w)
7. Довжина дзьоба та голови (mm), ”з кінчика дзьоба до потилиці” (lbh)
8. Довжина плечової кістки (у дюймах) (lh)
9. Довжина стегнової кістки (у дюймах) (lf)
10. Довжина тібіотарсуса (у дюймах) (lt)
11. Ширина черепа (у дюймах) (ws)
12. Довжина кіля (у дюймах) (lks)

Початкова обробка даних.

Таблиця подана у форматі .xls, яку можна считати в R за допомогою пакета readxl:

```
library(readxl)
birds.dat <- read_xls("./birds.xls")
# Вилучаємо колонку з номерами
birds.dat <- data.frame(birds.dat)[,-1]
# Колонки таблиці приймаємо за змінні в середовищі
attach(birds.dat)
```

Виміри довжин різних частин горобця подано або у міліметрах, або у дюймах. Для зручності зведемо це до єдиної шкали вимірювання, у міліметрах:

```
# Зводимо до єдиного виміру (з дюймів до міліметрів)
TL <- log(tl)
AE <- log(ae)
LBH <- log(lbh)
LH <- log(lh^25.4)
LF <- log(lf^25.4)
LT <- log(lt^25.4)
WS <- log(ws^25.4)
LKS <- log(lks^25.4)
```

Інколи пропонують розглядати кубічний корінь з ваги:

```
# Нормування ваги
W <- w^(1/3)
```

Визначення методу кластеризації.

Поставлена задача – визначити з методом кластеризації для пошуку кореляційних плеяд серед відомих змінних. Інтуїтивно було б правильно вибрати ієрархічну кластеризацію: тоді можна було б зрозуміти в залежності від ступеня скорельованості які кластери у нас утворюються, що не входить до того чи іншого кластеру, а що згодом увійде з певним рівнем значущості. Зокрема для ієрархічної кластеризації є можливість подати результати у вигляді дендрограми, на якій все зрозуміло подано що, куди і як входить.

Вибір метрики для кластеризації, відштовхуючись з умови задачі, буде базуватися на основі вибіркової кореляції між змінними:

$$d(X^1, X^2) = 1 - \left| \frac{c\hat{o}v(X^1, X^2)}{\sqrt{S_0^2(X^1)}\sqrt{S_0^2(X^2)}} \right|$$

де $c\hat{o}v(X^1, X^2)$ – вибіркова коварція змінних X^1 та X^2, $S_0^2(X^1)$ – це вибіркова дисперсія X^1. От вибір методу підрахунку відстані між кластерами (між точкою і кластером) є поки не зовсім зрозумілим: єдине що приходить в голову, що змінних досить мало, а тому застосування методі повного зв'язку може створити поодинокі компатні клстери, що не відповідають дійсності. Для того, що переконатися у виборі способу підрахунку відстані між кластерами, дослідимо поведінку (розташування) змінних на осях головних компонент та методу класичного багатовимірного шкалування.

Метод головних компонент.

Скористаємося побудовою головних компонент з використанням кореляційної матриці (чесно, не хочеться відкидати вагу внаслідок неіснування єдиної шкали вимірювання, спільної з довжинами та вагою, тому не використовується коваріація):

```
# Побудова таблиці
data1 <- cbind(W, TL, AE, LBH, LH, LF, LT, WS, LKS)
FEATURES <- colnames(data1)
# PCA
pca <- princomp(data1, cor = T)
plot(pca)
# Перші дві компоненти
plot(pca$loadings[,1], pca$loadings[,2], type = "n",
     xlab = "Comp.1", ylab = "Comp.2", main = "PC loadings")
grid()
text(pca$loadings[,1], pca$loadings[,2], label = FEATURES)
# Перші три компоненти
plot3d(pca$loadings[,1], pca$loadings[,2], pca$loadings[,3], type = "n",
       xlab = "Comp.1", ylab = "Comp.2", zlab = "Comp.3", main = "PC loadings")
text3d(pca$loadings[,1], pca$loadings[,2], pca$loadings[,3], texts = FEATURES)
```

Подивимося на діаграму власних чисел: Видно, що злам починається з першої компоненти,

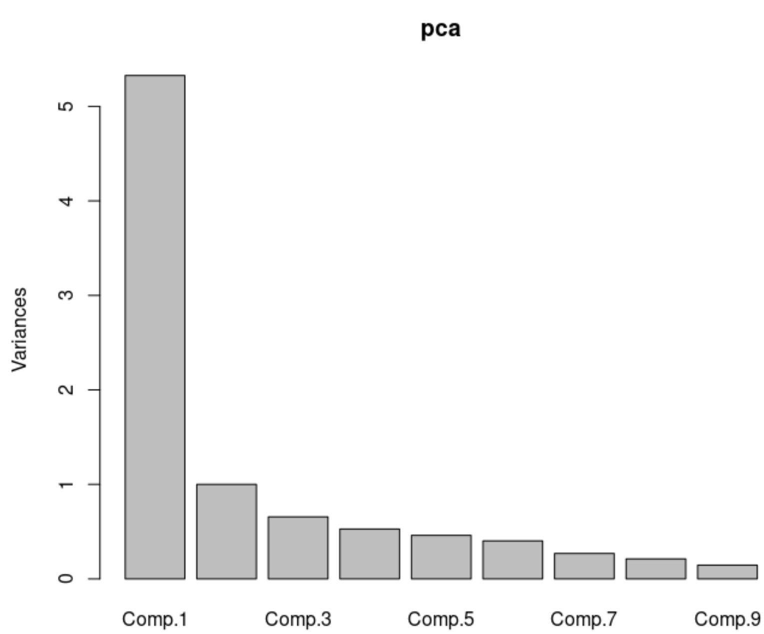

Рис. 1: Діаграма власних чисел.

однак видно що наступні компоненти зберігають досить значну частку дисперсії даних. Якщо обрати перші дві компоненти, то ми втрачаємо $\approx 41\%$ інформації, а для трьох комопнент – вже менше, десь 29.7%. Ми обираємо перші три компоненти для можливості якось обережно побудувати діаграми розсіювання, які можна нормально сприймати.

Діаграма розсіювання навантажень на змінні за першими двома компонентами:

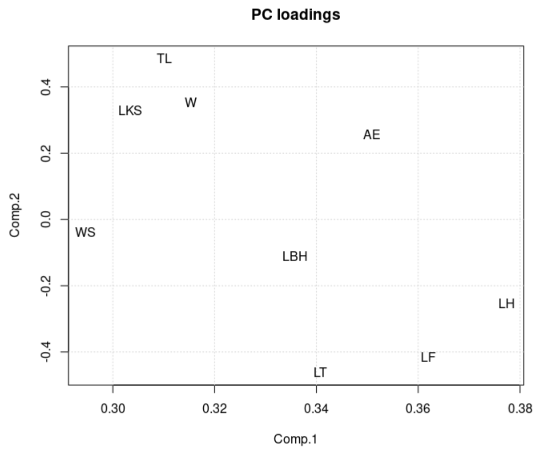

Рис. 2: Діаграма розсіювання навантажень на змінні.

Діаграма розсіювання навантажень на змінні за першими трьома компонентами: На дво-

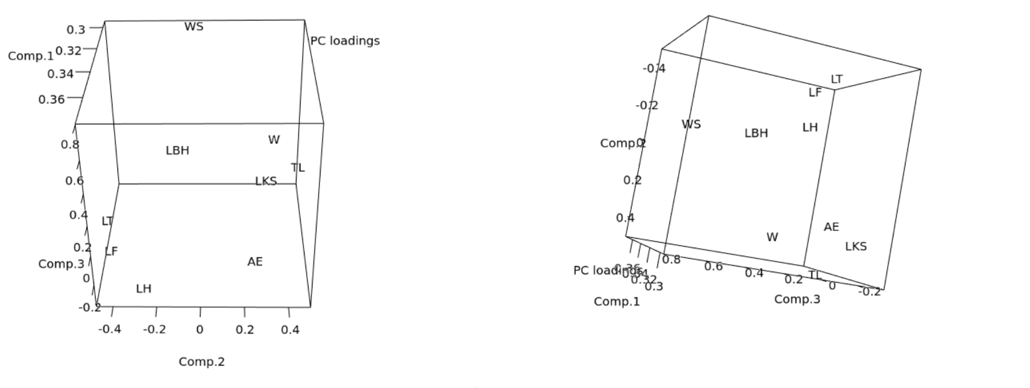

Рис. 3: Діаграма розсіювання навантажень на змінні.

вимірній діаграмі розсіювання не так однозначно, як могло б здаватися. Чітко видно, що групи змінних w, tl, lks та lbg, lh, lf, lt можуть бути складовими деяких двох кластерів. Проблема з ae та ws – вони "рівновіддалені" від цих двох хмарин, тому за цією діаграмою важко щось припустити щодо цих змінних. На тривимірній діаграмі можна спостерігати більш-менш аналогічну картину – хіба що здається, що ws виявляється найбільш віддаленою точкою серед змінних.

Класичне багатовимірне шкалування.

Доцільно було б одразу розглянути проектування у простір меншої вимірності зі збереженнями метричних властивостей змінних. Це можна зробити за допомогою класичного багатовимірного шкалювання:

```
# MDS 2d
mds2d <- cmdscale(as.dist(1 - abs(cor(data1))), k = 2)
plot(mds2d[,1], mds2d[,2], type = "n", xlab = "Axis1", ylab = "Axis2",
     main = "Multi-dimensional scaling, 2d")
grid()
text(mds2d[,1], mds2d[,2], labels = FEATURES)
# MDS 3d
mds3d <- cmdscale(as.dist(1 - abs(cor(data1))), k = 3)
plot3d(mds3d[,1], mds3d[,2], mds3d[,3], type = "n",
       xlab = "Axis1", ylab = "Axis2", zlab = "Axis3",
       main = "Multi-dimensional scaling, 3d")
text3d(mds3d[,1], mds3d[,2], mds3d[,3], texts = FEATURES)
```

Координати змінних після проектування у двовимірний та тривимірний простір відповідно дорівнюють: Побудуємо діаграми розсіювання: Картинка кардинально змінилася у порівнянні

2D	x	y	3D	x	y	z
W	-0.19614316	-0.09031226	W	-0.19614316	-0.09031226	-0.026263696
TL	-0.28437554	-0.01394478	TL	-0.28437554	-0.01394478	-0.169303082
AE	-0.13275721	0.14952330	AE	-0.13275721	0.14952330	-0.112763636
LBH	0.09508242	-0.09235448	LBH	0.09508242	-0.09235448	0.122631178
LH	0.13538885	0.11475065	LH	0.13538885	0.11475065	0.007646835
LF	0.23466186	0.07181740	LF	0.23466186	0.07181740	-0.033025256
LT	0.28019551	0.08559843	LT	0.28019551	0.08559843	-0.066374423
WS	0.07130733	-0.34657588	WS	0.07130733	-0.34657588	0.006581042
LKS	-0.20336004	0.12149762	LKS	-0.20336004	0.12149762	0.270871038

Табл. 1: Координати ознак при переході до багатовимірного шкалування.

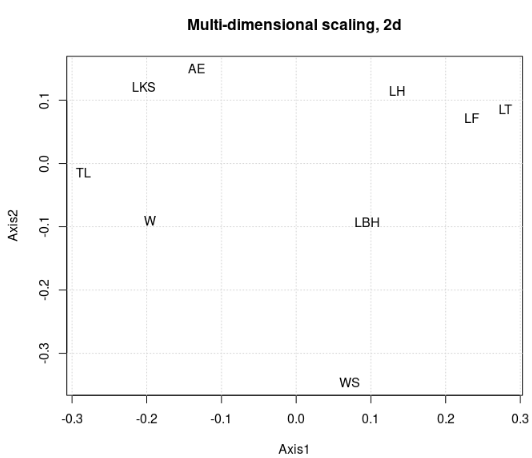

Рис. 4: Зниження вимірності до двовимірного простору.

з тим, що розглядалося для навантажень на змінні у головних компонентах. Чітко можна зобразити дві групи, які б утворювали відповідні кластери: змінні ae, lks, tl, w та lh, lt, lf, lbh. Змінна WS все ще далеко розташована від груп, однак на певному кроці кластеризації її віднесуть до правого кластера. Далі, подивимося ситуацію у тривимірному просторі:

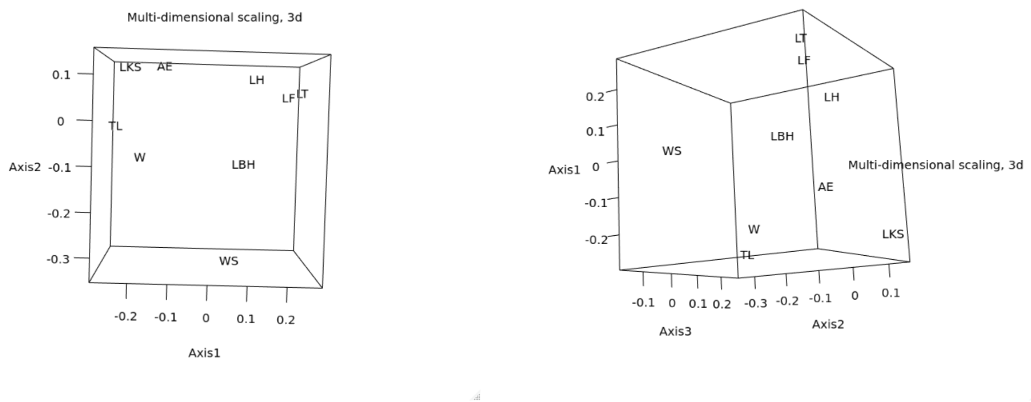

Рис. 5: Зниження вимірності до тривимірного простору.

Пояснення майже повторюється з випадком двовимірної діаграми. І в цьому разі змінна ws сильно віддалена від двох груп. Для підрахунку відстаней між кластерми скористаймось методом середнього зв'язку (або можна було б одного зв'язку, аби не використовувати повний зв'язок що призведе до неприродніх "компактних" множин).

Застосування кластеризації з рісемплінгом.

Враховуючи попередні міркування, реалізуємо кластеризацію:

```
pv1 <- pvclust(data1, method.hclust = "average", method.dist = "abscor")
plot(pv1)
pvrect(pv1, lty = 2)

clusters.picked <- pvpick(pv1)
```

Для демонстрації роботи алгоритму, побудуємо дендгрограму кластеризації:

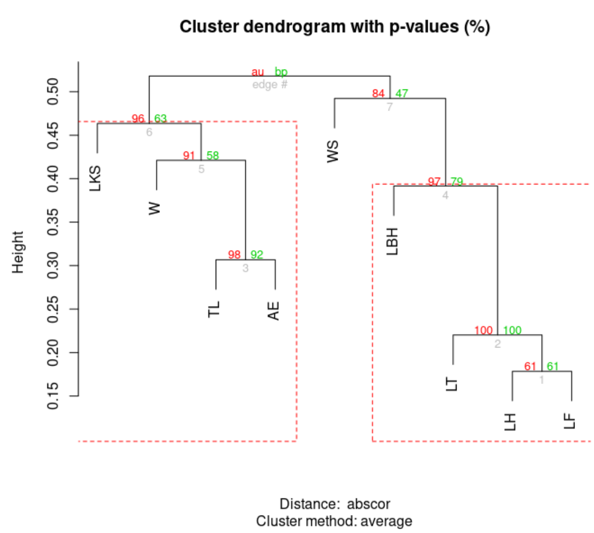

Рис. 6: Дендрограма кластеризації з "вірогідними" рівнями.

pvclust запроваджує два види досягнутих рівней значущості: AU (Approximately Unbiased) p-value та BP (Bootstrap Probability) value. AU p-value обчислюється за допомогою модифікованого бутстрепу (multiscale bootstrap resampling), який більш придатний до використання у порівнянні зі звичайною бутстреп-технікою, використаною для підрахунку BP. Найкращий вибір робиться для тих кластерів, AU p-value яких перевищує 0.95. Це ми, власне, бачимо, для тих груп змінних, які висувалися в якості кандидатів на утворення окремих кластерів. Змінну ws вважаємо за таку, що не входить до жодного з цих кластерів.

Висновки.

За ступенем скорельованості вдалося отримати два кластери з різних ознак: перший складається з ae, lks, tl, w, а другий – з lh, lt, lf, lbh. Таке розбиття має у певній мірі логічну інтерпретацію: між вагою та довжиною тіла є безпосередній зв'язок, а розмах крила та довжина тіла пов'язані пропорційно. Більш-менш аналогічно пояснюється скорельованість довжин конкретних частин тіла горобця.