

Київський національний університет імені Тараса Шевченка
 Кафедра теорії ймовірностей, статистики та актуарної математики

Р. Майборода

Самостійна робота по курсу

“Статистичний аналіз багатовимірних даних”
Для студентів магістратури за напрямом “статистика”

*Iндивідуальні завдання
та рекомендації по виконанню*
Документ в процесі розробки. Версія від 18.11.2018

Київ — 2018

Вступ

Для виконання завдань потрібно встановити R та RStudio на своєму комп'ютері.

Щоб встановити R для Windows зайдіть на сторінку

<http://cran.r-project.org/bin/windows/base/>

і виберіть Download R 3.4.1 for Windows (номер версії, скоріше за все, буде вже іншим). Після цього запустіть програму, яка буде завантажена на ваш комп'ютер і відповідайте на її запити.

Якщо вам потрібна версія R для іншої операційної системи, зайдіть на сторінку

<http://www.r-project.org/>

і виберіть там варіант, який вас влаштовує.

Для того, щоб встановити RStudio, зайдіть на сторінку

www.rstudio.com

і виберіть там варіант для завантаження. Встановлювати RStudio треба після того, як буде встановлено R.

Книжку [1], присвячену статистичному аналізу даних за допомогою R, можна отримати за адресою:

<http://probability.univ.kiev.ua/userfiles/mre/compsta.pdf>

Завдання 1.

Кластеризація з відомою кількістю кластерів.

Частина 1.

1. У архіві

<http://probability.univ.kiev.ua/userfiles/mre/mult6task.rar>

знайдіть файл з іменем `mult<N>.txt`, де `<N>` — номер Вашого варіанту. Запишіть його на Ваш комп'ютер у зручному для читання місці.

2. У файлі міститься таблиця модельованих даних. У першому її рядочку знаходяться назви змінних, а кожен наступний рядочок відповідає одному спостереженню. Кожен стовпчик таблиці відповідає одній змінній.

Прочитайте цю таблицю у R за допомогою функції `read.table()` і проведіть кластерний аналіз цих даних використовуючи методи центроїдів та медоїдів. У методі медоїдів використайте звичайну евклідову відстань між спостереженнями.

Спробуйте різні кількості кластерів, починаючи від 2-х до 20-ти.

3. Виберіть оптимальну кількість кластерів, використовуючи внутрішньокластерну суму квадратів для методу центроїдів та графік середніх силуетів як у методі центроїдів так і у методі медоїдів.

Відповідні графіки наведіть у звіті.

4. Для двох-трьох варіантів кластеризації, які Ви визнаєте найкращими, відобразіть результати на діаграмах розсіювання, використовуючи метод головних компонент та метод канонічних компонент.

Діаграми наведіть у звіті.

5. Порівняйте обрані Вами варіанти кластеризації з п. 4 використовуючи індекс Ренда та поліпшений індекс Ренда.

За потреби, наведіть таблицю спряженості для розглянутих кластеризацій.

Виберіть остаточний варіант кластеризації, який Ви вважаєте оптимальним. Зробіть висновки (який з використаних методів виявився вдалим, який — ні, які методи дали однакові результати і т.п.).

Рекомендації по виконанню.

Частина перша. Прочитаємо дані, використовуючи функцію `verb|read.table()|`:

```
> samp<-read.table("c:\\rem\\mult6\\mult0.txt")
```

Тепер дані знаходяться у фреймі даних `samp`.

Провести кластеризацію центроїдним методом можна, використовуючи функцію `kmeans()`, їй потрібно передати дані і вказати кількість кластерів. Крім того, можна задати опцію `nstart`, що вказує, скільки варіантів початкових наборів центрів буде випробуватись (ці набори вибираються випадково, якщо не задати їх явно у опції `centers`).

Значення номерів кластерів до яких відносяться об'єкти, функція вміщує у атрибуті `$cluster` свого результату.

Можна вивести діаграму розсіювання, у якій точки з різних кластерів пофарбовані у різні кольори. У наступному прикладі це зроблено для випадку чотирьох кластерів.

```
> km.res <- kmeans(samp, 4, nstart = 25) # кластеризація
> km.res$tot.withinss # внутрішньокластерна сума квадратів
[1] 34148.6

> km.res$betweenss    # міжкластерна сума квадратів
[1] 266704.5

> km.res$betweenss/km.res$tot.withinss
[1] 7.810114

> pal<-c("black","red","green","blue")      # палітра кольорів
> plot(samp[,1],samp[,2],cex=0.2,col=pal[km.res$cluster])
```

Внутрішньокластерна сума квадратів у 7.8 разів менша ніж міжкластерна, отже можна сподіватись, що ця кластеризація виявляє певну структуру даних.

На діаграмі (рис.1) візуально виділяються приблизно 8 кластерів, але результати кластеризації їм явно не відповідають. (На діаграмах розсіювання інших змінних може бути видно щось зовсім інше, спробуйте.) Варто продовжити спроби.

Побудуємо діаграму внутрішньокластерних сум квадратів в залежності від кількості кластерів — рис. 2:

```
> library(factoextra)
> fviz_nbclust(samp, kmeans, method = "wss",k.max = 20)
```

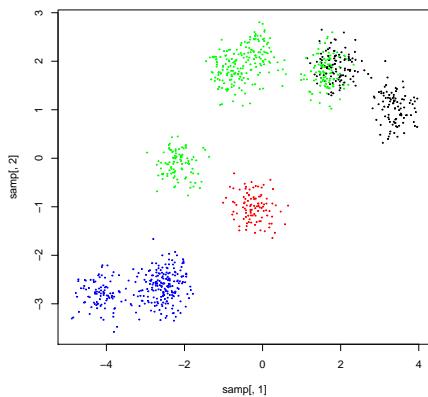


Рис. 1: Діаграма розсіювання перших двох змінних. Кластеризація методом центроїдів.

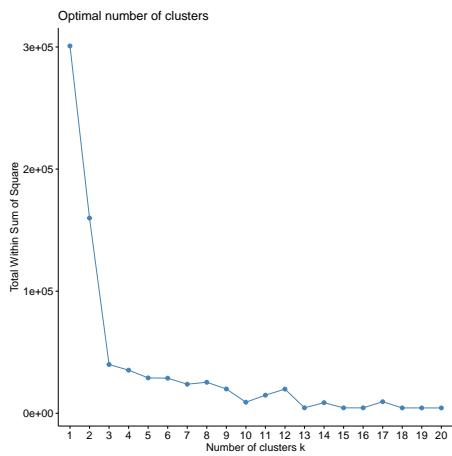


Рис. 2: Внутрішньогрупові суми квадратів. Кластеризація методом центроїдів.

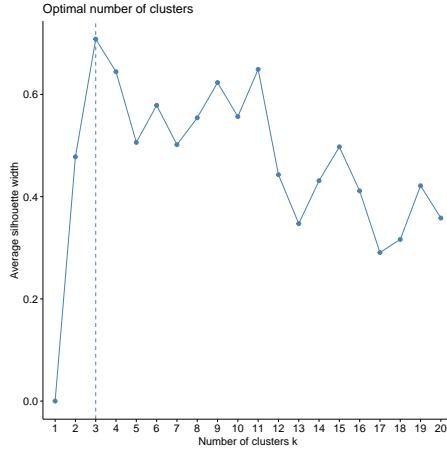


Рис. 3: Середні силуєти. Кластеризація методом центроїдів.

На цій діаграмі видно “злам” при $k = 3$ і невеликий “провал” при $k = 10$. Ці числа є претендентами на оптимальну кількість кластерів.

Побудуємо діаграму середніх силуєтів — рис. 3

```
> fviz_nbclust(samp, kmeans, method = "silhouette", k.max = 20)
```

На цій діаграмі помітні максимуми при $k = 3, k = 11, k = 15$.

Розглянемо випадок $k = 10$. (У роботі Ви перевірите всіх претенден-тів)

Проведемо кластеризацію і подивимось на діаграму розсіювання да-них у просторі перших двох головних компонент — рис. 4:

```
> km.res <- kmeans(samp, 10, nstart = 25)
> pal<-c("black", "red", "blue", "green", "magenta", "chocolate",
+        "darkblue", "darkred", "aquamarine", "grey")
> plot(princomp(samp)$scores[,1:2], col=pal[km.res$cluster], cex=0.2)
```

На рисунку помітні дев’ять кластерів з десяти (один виявився “прихова-ним” за іншими), вони розбиваються на три групи. Не зовсім очевидно, наскільки запропоноване розбиття дійсно відображає структуру даних, наприклад, у верхній групі — тут кластери не виглядають відділеними один від одного.

(Тут можна іще подивитись інші пари головних компонент, напри-клад, другу і третю, або покрутити дані у тривимірній графіці).

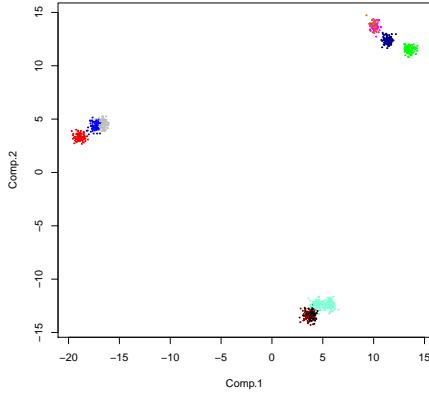


Рис. 4: Діаграма розсіювання перших двох головних компонент, Кластеризація методом центроїдів.

Відобразимо діаграму розсіювання даних у просторі третьої і четвертої канонічних компонент:

```
> require(CCA)
> cl<-km.res$cluster
> k<-length(levels(as.factor(cl)))
> n<-nrow(samp)
> C<-matrix(data=as.numeric(rep(cl,k)==rep(1:k,each=n)),ncol=k,nrow=n)
> cc_res<-rcc(samp,C,0.1,0.1)
> plot(cc_res$scores$xscores[,3:4],col=pal[km.res$cluster],cex=0.2)
```

На цій діаграмі структура “трьох груп” кластерів не помітна, але видно всі 10 кластерів і вони розташовані окремо один від одного. Ті кластери, що розмістилися поруч на цій діаграмі, розташовані у різних місцях на діаграмі головних компонент.

Отже можна зробити висновок, що у даних виділяються 10 кластерів, що розбиваються на три групи.

Аналогічне дослідження слід провести, використовуючи метод методів. Для цього можна використати функцію `pam()` з бібліотеки `cluster`:

```
> library(cluster)
> pam.res<-pam(samp,10)
```

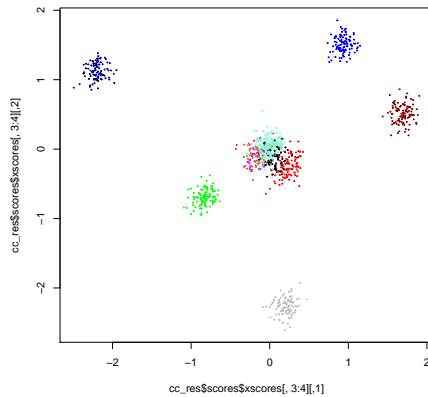


Рис. 5: Діаграма розсіювання перших двох канонічних компонент, Кластеризація методом центройдів.

Цей метод може дати результати, що відрізняються від результатів методу центройдів. Щоб порівняти, наскільки різними вийшли кластеризації, можна підрахувати індекс Ренда між ними функцією `rand.index()` з бібліотеки `fossil`:

```
> library(fossil)
> rand.index(pam.res$clustering, km.res$cluster)

[1] 0.9728388
```

Індекс виявився рівним 1, тобто ці кластеризації повністю однакові.

Це підтверджує і таблиця спряженості:

```
> library(MASS)
> table(pam.res$clustering, km.res$cluster)
```

	1	2	3	4	5	6	7	8	9	10
1	0	0	0	0	0	0	0	0	96	0
2	77	0	0	0	0	0	0	0	0	0
3	0	0	0	0	43	61	0	0	0	0
4	0	0	0	0	0	0	0	0	114	0
5	0	0	0	98	0	0	0	0	0	0

Частина 2.

Підберіть самі для аналізу дані, за якими було б цікаво провести класифікацію. Застосуйте до них алгоритми, що були використані у частині 1. (Перш ніж виконувати аналіз даних, бажано узгодити їх використання та план роботи з викладачем). Дані потрібно описати у звіті (навести числові значення, якщо їх не дуже багато, або вказати, де їх можна знайти).

Якщо у Вас немає ідеї щодо вибору даних, можна самостійно заповнити рольову гратку і провести її обробку.¹

Для цього виберіть який-небудь добре знайомий вам твір (книжку, фільм, телесеріал, комп'ютерну гру...) в якому є не менше семи дійових осіб з чітко окресленими особистостями, що відрізняються одна від одної.²

Наприклад, візьмемо книгу А. Мілна про Вінні-Пуха та його друзів. З неї можна вибрати Вінні, Крістофера Робіна (КР), Паця, Кролика, Тигру, Сову, Кенгу, Іа.

Запишіть імена обраних дійових осіб у стовпчик. Виберіть випадковим чином трьох з них (можна для цього скористатись генератором випадкових вибірок — функцією `sample()` у R, але не обов'язково). Подумайте, яку спільну якість особистості мають два з них, за котрою вони відрізняються від третього. Наприклад, взявши Кролика, Тигру і Сову, можна помітити, що Кролик і Сова — розсудливі, а Тигра — спонтанний ("вискакучий"). Такі якості у методі рольових граток називають *конструктами*. У конструкта є два *полюси*, у нашому прикладі це розумність (позитивний полюс³) і спонтанність — негативний.

Спробуйте розмістити всіх обраних вами дійових осіб (будемо називати їх об'єктами нашої гратки) вздовж осі, що з'єднує позитивний та негативний полюси конструкта: той об'єкт, що найкраще відповідає позитивному полюсу, отримує найбільший ранг (КР — 8). Той, що найбільш

¹Про методи психологічного аналізу особистості за допомогою рольових граток (граток Келлі) можна прочитати у [3].

²Ви можете попросити заповнити гратку когось із своїх друзів. Тоді твір має бути знайомим для нього/неї, а не обов'язково для вас. **Зауваження.** Недоцільно заповнювати гратку гуртом, це має робити хтось один. Твір, за яким гратка заповнюється має бути реально існуючим, а не, скажімо, намітками сценарію фільму, який ви зираєтесь коли-небудь зняти.

³Позитивність і негативність тут не є моральними чи якимось іншими оцінками, а задаються довільно для зручності опису та порівняння з іншими конструктами.

лижчий до негативного полюса — найменший (Тигра — 1).

Якщо обраний вами конструкт неможливо застосувати бо багатьох об'єктів (наприклад, він придатний до опису тільки осіб однієї статі, а у вашій гратці представлені обидві) спробуйте узагальнити його так, щоб він став застосовним до всіх, або хоча б майже всіх. Тим об'єктам, що не отримали рангу за даним конструктом, привласніть значення **NA** (пропущене значення).

Після цього задайте наступний конструкт за тією ж схемою: випадково виберіть набір з трьох елементів і т.д. Намагайтесь обирати нові конструкти так, щоб вони не повторювали вже вибраних. При бажанні можна внести до складу конструктів будь-які риси, що, на вашу думку, потрібні для характеризації об'єктів і були пропущені раніше.

Остаточна кількість виділених конструктів повинна бути не менше 7-ми.

Залишіть отримані вами дані у вигляді таблиці, рядочки яких відповідають об'єктам (дійовим особам), а стовпчики - конструктам. У таблицю занесіть відповідні ранги. Об'єктам та конструктам дайте скорочені імена (не більше 2-х символів). Ці імена будуть використовуватись при візуалізації та при описі результатів кластеризації. У звіті наведіть цю таблицю (гратку), назву твору, за яким складена гратка та пояснення скорочених імен.

Спробуйте обробити цю гратку, використовуючи методи з частини першої. При цьому можна на роль об'єктів використовувати дійових осіб. Тоді конструкти слід трактувати, як змінні, що описують об'єкти. (Кластеризація рядочків гратки).

Можна, навпаки, розглядати конструкти як об'єкти для кластеризації, а значення рейтингів дійових осіб за цими конструктами розглядати, як змінні, що описують конструкти. (Кластеризація стовпчиків).

Проведіть окремо кластеризацію рядочків, окремо - кластеризацію стовпчиків.

Подивіться, чи можна виявити зв'язок між цими кластеризаціями?

Завдання 2.

Класичне багатовимірне шкаловання.

Частина 1.

Застосуйте класичне багатовимірне шкаловання для візуалізації даних з частин 1 і 2 завдання 1.

При цьому для шкаловання спочатку перетворіть індивідуальні дані у таблицю відмінностей, використавши для цього три варіанти відстаней: евклідову, манхаттанську і максимальну. Для кожного варіанту виведіть двовимірну діаграму розсіювання для результатів. Точки на діаграмах розфарбуйте з урахуванням кластеризацій, отриманих у завданні 1.

Порівняйте отримані візуалізації з результатами завдання 1, зробіть висновки.

Рекомендації по виконанню.

Для обчислення відстаней можна скористатись функцією `dist(x, method)`.

Параметр `x` — фрейм даних (або матриця індивідуальних змінних), за яким розраховуються відстані між об'єктами (рядочками).

`method` — опція, що задає тип відстаней. Може бути "`euclidean`" (за умовчанням), "`maximum`", "`manhattan`" та ін.

Для багатовимірного шкаловання можна використати функцію `cmdscale(d, k=2, eig=FALSE, ...)`

де `d` — матриця відстаней для шкаловання,

`k` — вимірність простору, в якому підшукується відповідна конфігурація точок,

`eig` — логічний параметр, що вказує, чи потрібно підраховувати всі власні числа матриці коваріацій для конфігурації у просторі великої вимірності.

Приклад:

```
> samp<-read.table("c:\\rem\\mult6\\mult0.txt")
> km.res <- kmeans(samp, 10, nstart = 25)
> pal<-rainbow(10)
> d <- dist(samp,method="maximum") # minimum distances between the rows
> fit <- cmdscale(d,eig=TRUE, k=2) # k is the number of dimensions
> x <- fit$points[,1]
> y <- fit$points[,2]
> plot(x, y,col=pal[km.res$cluster],cex=0.2)
```

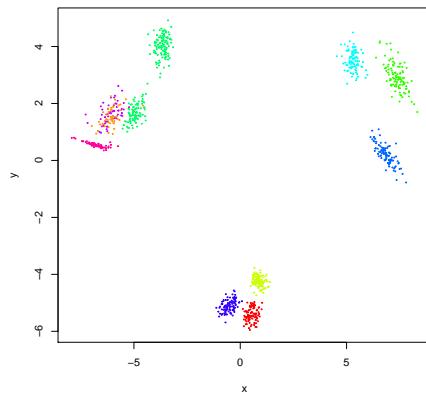


Рис. 6: Результат класичного багатовимірного шкалювання.

Результат — на рис 6.

Частина 2.

Розробіть скрипт для застосування техніки проекції на канонічні компоненти до конфігурації точок отриманих методом класичного багатовимірного шкалювання у просторі великої вимірності.

Використайте цей скрипт для візуалізації даних із завдання 1 з урахуванням їх кластеризації.

Завдання 3.

Ієрархічна класифікація (кластеризація).

Частина 1.

Застосуйте техніку ієрархічної класифікації до даних з частин 1 і 2-ї завдання 1.

Використайте методи одного, повного і середнього зв'язку, евклідову, манхеттенську і максимальну відстані.

Побудуйте дендрограми класифікації, порівняйте їх, зробіть висновок про можливу кількість кластерів.

Підрахуйте кофенетичні відстані та відповідні коефіцієнти кореляції з відстанями між об'єктами. Зробіть висновок про те, яка з отриманих дендрограм найкраще відображає реальні відстані між спостережуваними об'єктами.

Зробіть обрізання дендрограм на обраному рівні, отримайте кластеризацію і порівняйте результати з отриманими у завданні 1.

Висновки запишіть у звіті.

Частина 2.

Застосуйте ієрархічний кластерний аналіз до даних, обраних Вами у частині 2 завдання 1.

Спробуйте використати різні відстані та різні методи кластеризації. Порівняйте результати з отриманими у завданні 1.

Рекомендації по виконанню.

Для проведення ієрархічної кластеризації можна скористатись функцією `hclust()`. Для цього попередньо потрібно підрахувати відстані між спостереженнями (якщо спостереження задані індивідуальними значеннями змінних). Для відображення дендрограм можна застосувати функцію `plot`. У наступному скрипті відображення дендрограм виконано двома різними способами, причому опції встановлені так, щоб назви об'єктів не виводились знизу під дендрограмою. (Назви доцільно друкувати, якщо об'єктів/кінцевих гілок не більше 15-20).

```
> hc <- hclust(dist(samp))
> plot(hc, labels=F)
> hcd <- as.dendrogram(hc)
> plot(hcd, leaflab="none")
```

Результат — на рис. 7.

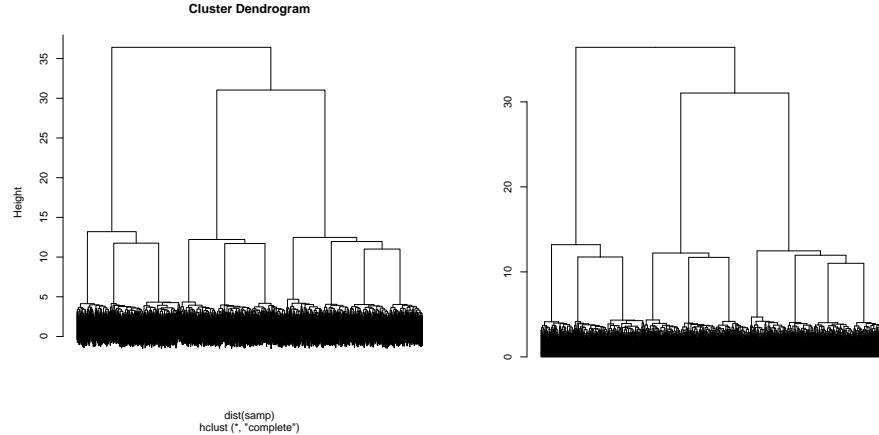


Рис. 7: Результат ієрархічної кластеризації.

Функція `as.dendrogram()` у цьому скрипті виділяє зі звіту функції `hclust()` власне дендрограму і вміщує її у об'єкт під назвою `hcd`. Цей об'єкт можна використовувати далі для більш детального відображення.

Наприклад, на рис. 7 можна виділити 10 кластерів, які утворюються, якщо обрізати дендрограму на рівні $h = 8$. Але структура даних всередині кожного з цих кластерів на рисунку не зрозуміла, через великий обсяг даних. Доцільно розділити дендрограму на одну верхню частину і 10 окремих нижніх, що відповідають різним виділеним кластерам.

Це зручно зробити, використовуючи функції з пакету `dendextend`. Як приклад його застосування, наведемо наступний скрипт:

```
> library(dendextend)
> hup<-cut(hcd, h=6)$upper
> labels(hup)<-letters[1:10]
> plot(hup,
+ main="Upper tree of cut")
> h1<-cut(hcd, h=6)$lower[[1]]
> labels(h1)=NULL
> plot(h1,main="a")
```

Результат його виконання — на рис. 8. На рисунку ліворуч зображення верхня частина дендрограми, обрізана на висоті $h=6$. Гілки на місці обрізання позначені латинськими літерами a, \dots, j . Праворуч — дендрограма, що є продовженням гілки a . На цій дендрограмі добре видно,

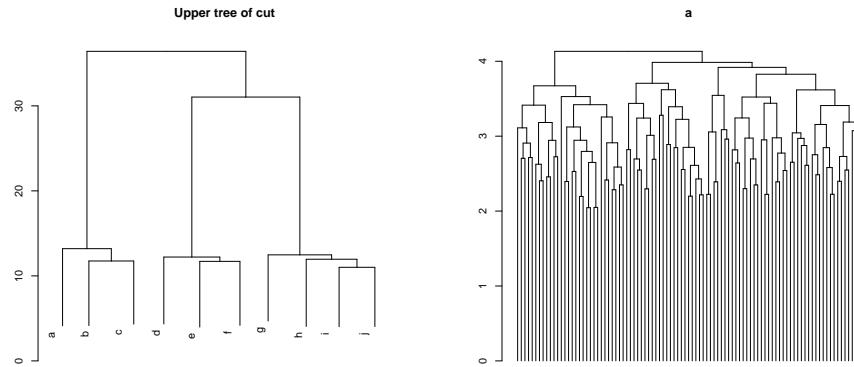


Рис. 8: Результат ієрархічної кластеризації.

що об'єкти всередині кластера а розташовані на приблизно одинакових відстанях один від одного, розбиття їх на менші кластери недоцільне (з точки зору обраного методу кластеризації).

Розбиття дендрограми було зроблено функцією `cut()`. Верхню частину дендрограми ця функція повертає у атрибуті `$upper`, а список нижніх частин — у атрибуті `$lower`. Опція `h` вказує висоту розбиття.

Кофенетичні відстані можна підрахувати, використовуючи функцію `cophenetic()`, а їх кореляції з початковими відстанями — за допомогою функції `cor()`:

```
> coph <- cophenetic(hc)
> cor(coph, dist(samp))
```

[1] 0.9900337

— коефіцієнт кореляції близький до 1, отже, дендрограма добре відповіє початкові відстані.

Для отримання за дендрограмою кластеризації на фіксовану кількість кластерів, використовують функцію

```
cutree(tree, k = NULL, h = NULL),
```

де опція `k` задає кількість кластерів, а `h` — висоту, на якій проводиться обрізання дендрограми. (Задати треба тільки одну з цих опцій).

Як результат функція видає вектор з отриманою кластеризацією.

Завдання 4.

Частина 1. Дані для аналізу знаходяться у архіві `multi.rar`, який можна отримати за адресою

<http://probability.univ.kiev.ua/userfiles/mre/multi.rar>

Розпакуйте цей файл і виберіть набір даних у форматі `txt` з номером, який відповідає номеру вашого варіанту.

Для вибраного набору даних проведіть аналіз головних компонент, намагаючись виявити приховану геометричну структуру у даних. Для цього знайдіть головні компоненти набору, виберіть з них ті, які на Вашу думку містять суттєву інформацію і відобразіть їх на дво- або три-вимірному рисунку. Якщо при цьому виявиться цікава геометрична структура, опишіть її.

Якщо одразу не вдається виявити структуру, спробуйте відобразити інші головні компоненти. Якщо при відображені якого-небудь набору структура виділяється — опишіть її у звіті, навівши відповідні рисунки. Якщо ні — опишіть у звіті, які саме експерименти Ви зробили і вкажіть, що переконливого результату досягти не вдалося.

Після цього дослідіть кластерну структуру даних використовуючи техніку ієрархічної кластеризації.

Спочатку спробуйте проаналізувати безпосередньо початкові дані застосовуючи різні техніки кластеризації з евклідовою метрикою. Опишіть результати у звіті.

Чи виділилась та структура, яку вдалось виявити за головними компонентами? Якщо ні, спробуйте застосувати ієрархічну кластеризацію до тих даних, які були отримані методом головних компонент. Використайте евклідову метрику, метрику Махалонобіса, метрику сіті-блок.

У звіті наведіть дендрограми кластеризації, 2D і 3D діаграми розсіювання з розфарбуванням кластерів а також матричну діаграму розсіювання початкових даних з розфарбуванням.

Застосуйте до даних (як до початкових, так і до визначених головних компонент) техніку спектральної кластеризації. Чи дозволяє вона виділити ті кластери, які вдається помітити, досліджуючи дані візуально?

Наскільки техніка спектральної кластеризації виявилась більш, або менш зручною ніж техніка ієрархічної кластеризації?

Зробіть висновки у звіті.

Рекомендації по виконанню завдання 4.

Для знаходження головних напрямків та значень головних компонент

для багатовимірних спостережень можна використати функцію `princomp()`. Її потрібно передати параметри:

`x` — фрейм даних зі значеннями змінних, які використовуються для побудови головних напрямків;

`cor` — логічна опція, що вказує, чи шукати головні напрями кореляційної матриці (`cor=T`), чи коваріаційної (`cor=F` — за умовчанням).

Результат `princomp()` записується у об'єкт з атрибутами:

`$sdev` — квадратні корені з власних чисел, що відповідають головним компонентам (стандартні відхилення компонент);

`$loadings` — навантаження початкових змінних на головні компоненти;

`$scores` — значення головних компонент для спостережуваних об'єктів.

Наприклад, розглянемо дослідження даних, що знаходяться у файлі `Spiral.txt`.

```
> z<-read.table("c:\\rem\\multi\\Spiral.txt",header=F)
> pairs(z,cex=0.1)
```

— прочитали файл і нарисували діаграму розсіювання пар (рис. 9). На діаграмі можна побачити, що у даних виділяється більш щільне ядро і периферія, де точки розташовані менш щільно. На деяких діаграмах (наприклад, V5:V6) виділяються кластери, що розташовані окремо від основної маси точок. Але виразної геометричної структури даних не помітно.

Застосуємо метод головних компонент:

```
> res<-princomp(z)
> plot(res)
> summary(res)
```

Importance of components:

	Comp.1	Comp.2	Comp.3	Comp.4	Comp.5
Standard deviation	4.9416783	2.9788187	2.3869830	1.67201859	1.02181729
Proportion of Variance	0.5701522	0.2071715	0.1330272	0.06527157	0.02437746
Cumulative Proportion	0.5701522	0.7773237	0.9103510	0.97562254	1.00000000
	Comp.6	Comp.7	Comp.8	Comp.9	Comp.10
Standard deviation	4.875519e-08	4.168705e-08	0	0	0
Proportion of Variance	5.549879e-17	4.057366e-17	0	0	0
Cumulative Proportion	1.000000e+00	1.000000e+00	1	1	1

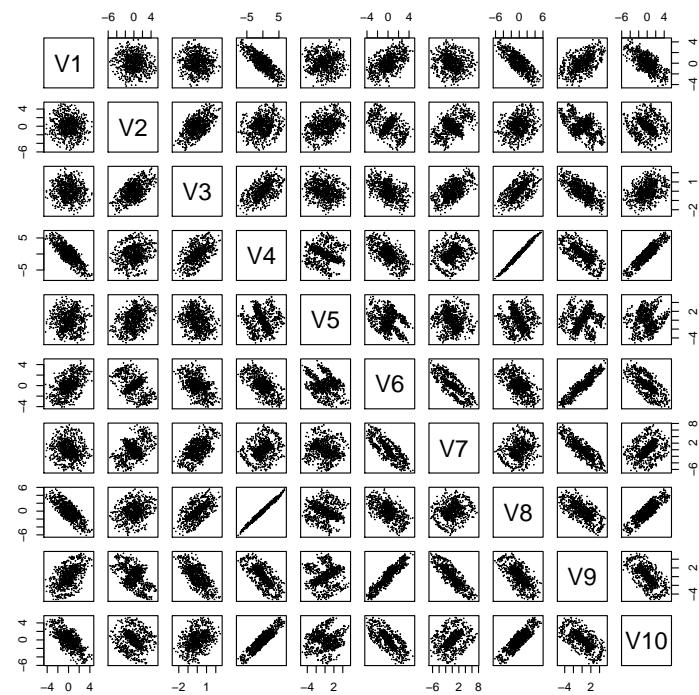


Рис. 9: Діаграма розсіювання пар для Spiral.txt

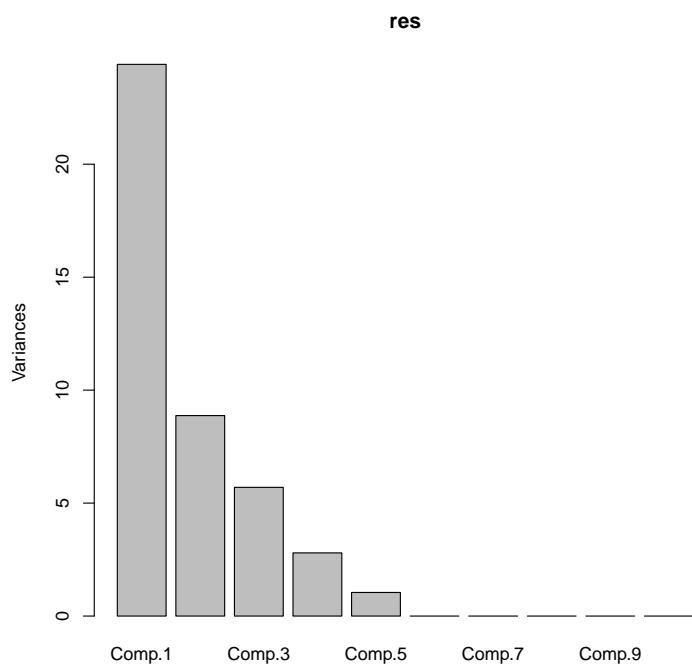


Рис. 10: Діаграма власних чисел для Spiral.txt

На діаграмі власних чисел (рис. 10) помітно виділяється перша головна компонента (вона пояснює 57% дисперсії даних). Власні числа компонент з 2 по 5 спадають поступово, зламів не помітно. Компоненти з 6 по 10 виглядають практично несуттєвими. Модель з перших трьох компонент пояснює 91% дисперсії даних.

Виведемо попарні діаграми розсіювання для перших трьох компонент (рис. 11)

```
> plot(res$scores[, 1:2])
> plot(res$scores[, 2:3])
> plot(res$scores[, c(1,3)])
```

На цих рисунках структуру можна вгадати, але зручніше це робити, використовуючи тривимірну графіку:

```
library(rgl)
plot3d(res$scores[, 1:3])
```

(див. рис. 12) Обертаючи цю тривимірну діаграму, можна побачити, що дані розбиваються на два кластери. Перший кластер утворюють точки, що формують циліндр, який іде по діагоналі квадрата, утвореного першою і третьою компонентами. Точки другого кластеру розташовані вздовж спіралі, що огортає цей циліндр на певній відстані від нього.

Спробуємо застосувати ієрархічний кластерний аналіз (див. завдання 3) до виділених головних компонент.

```
> z.dist<-dist(res$scores[, 1:3])
> z.hclust<-hclust(z.dist, method ="single")
> plot(as.dendrogram(z.hclust), leaflab="none")
```

Результат — на рис. 13. На дендрограмі досить виразно виділяються 2 або 3 кластери. Оскільки виявлена у завданні 5 структура даних складалась з двох елементів (основний циліндр і спіраль, що його огортає) спробуємо розбити спостереження на два кластери функцією `cutree()`:

```
> groups2<-cutree(z.hclust, k=2)
```

(опція `k` вказує кількість кластерів, у змінній `groups2` буде масив номерів кластерів, до яких належить відповідний об'єкт).

Використаємо номери отриманих кластерів для розфарбування діаграм розсіювання:

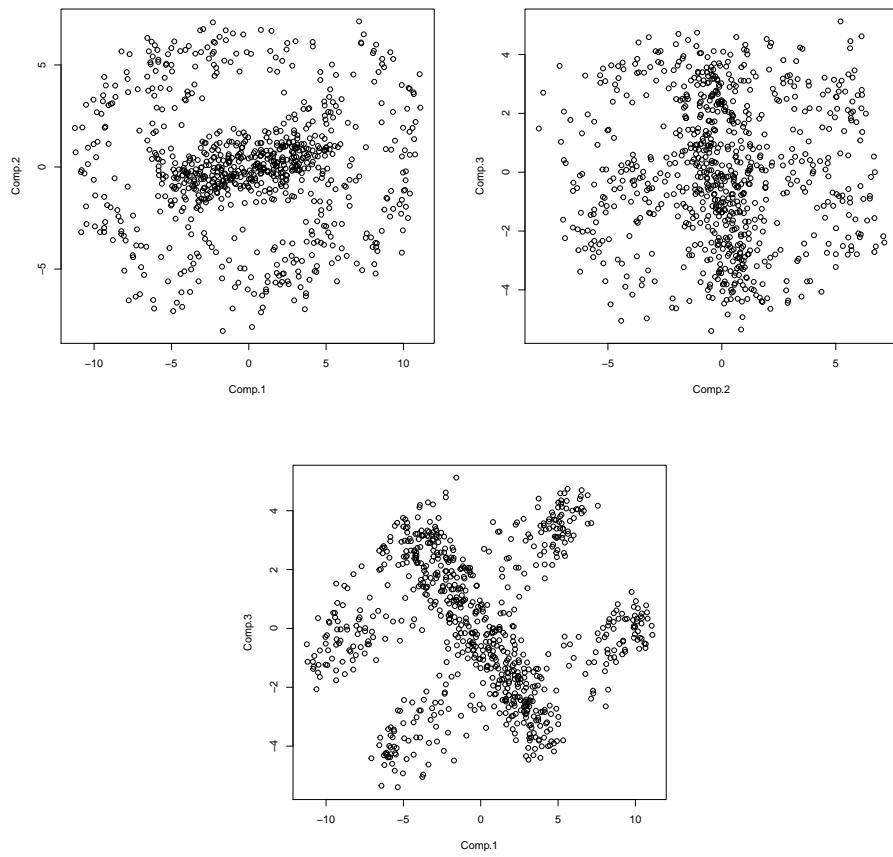


Рис. 11: Попарні діаграми головних компонент для Spiral.txt

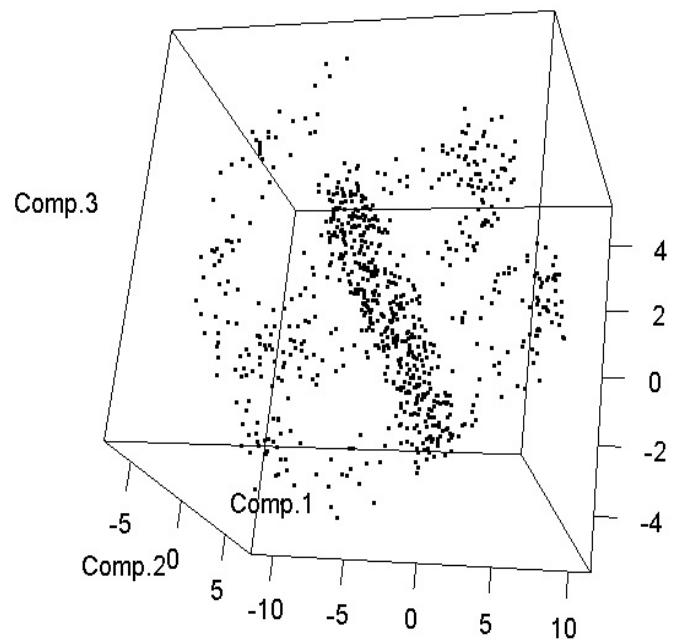


Рис. 12: 3D-діаграма перших трьох головних компонент.

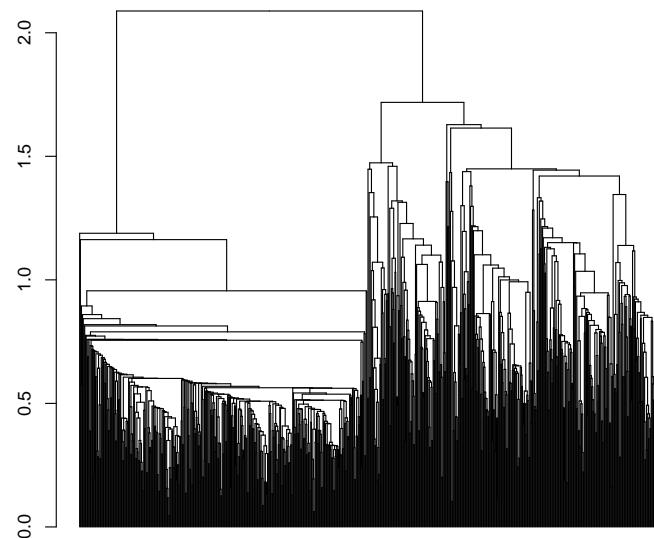


Рис. 13: Дендрограмма кластеризації для Spiral.txt

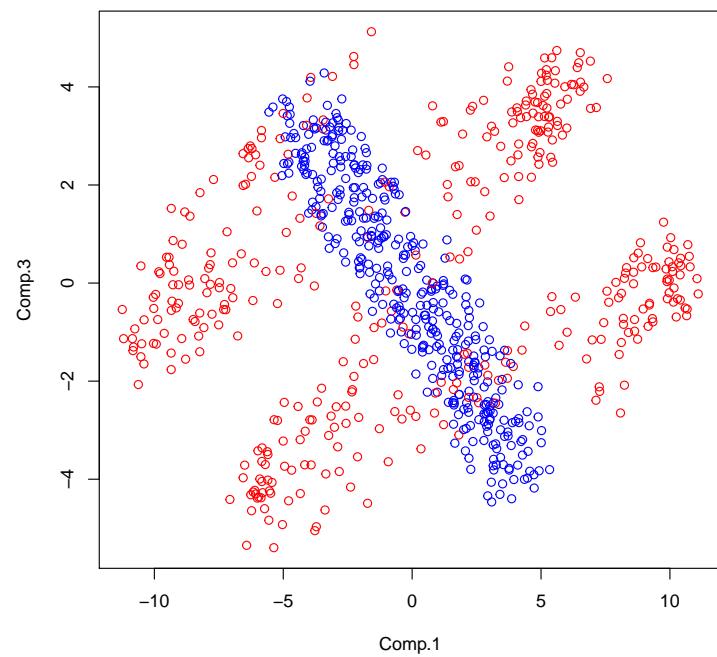


Рис. 14: Кластери ієрархічної кластеризації у просторі 1-ї і 3-ї головних компонент Spiral.txt

```
> plot(res$scores[,c(1,3)], col=c("red", "blue") [groups2])
```

(див. рис. 14). Як бачимо, структура даних виділилась при кластеризації вірно.

Спробуємо провести спектральну кластеризацію з використанням функції `specc()` з пакету `kernlab`.

У цю функцію потрібно передати параметри:

`x` — матриця даних, рядочки якої містять значення характеристик об'єктів.

`centers` — якщо це одне число, то воно задає кількість кластерів у кластеризації, якщо вектор — номери початкових центрів кластерів у кластеризації.

`kpar` — опція, що задає параметр σ функції, яка перетворює відстані між об'єктами у елементи матриці суміжності. Значення за умовчанням — `automatic` відповідає автоматичному вибору одного σ для всіх пар об'єктів. При заданні `kpar=local` використовується локальна підгонка σ .

Наприклад, для наших даних з файлу `Spiral.txt` кластеризація на два кластери може виглядати так:

```
> library(kernlab)
> skm=spec(z, centers=2)
> plot(res$scores[,c(1,3)], col=c("red", "blue") [skm])
```

(див. рис. 15). Як бачимо, при такому підході алгоритм не виділив ті кластери, які були помічені візуально.

При використанні для кластеризації проекції на перші три головні компоненти отримуємо адекватне розбиття на кластери (див. рис. 16):

```
> skm=spec(res$scores[,c(1,2,3)], centers=2)
> plot(res$scores[,c(1,2)], col=c("red", "blue") [skm])
```

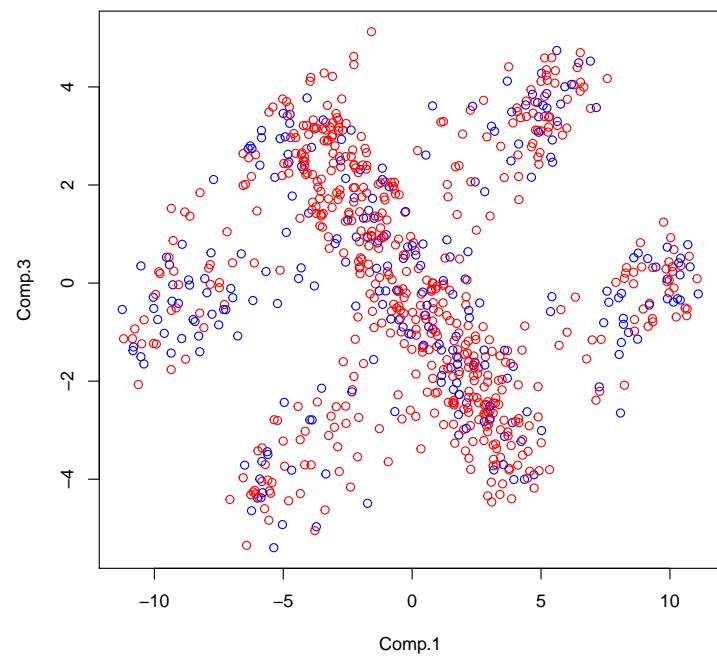


Рис. 15: Кластери спектральної кластеризації за початковими даними у просторі 1-ї і 3-ї головних компонент Spiral.txt

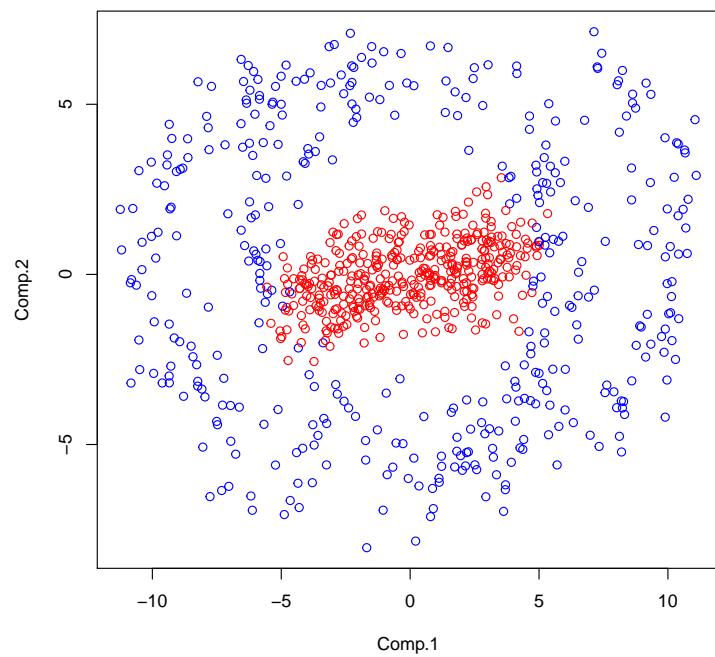


Рис. 16: Кластери спектральної кластеризації за проекцією на перші три головні компоненти у просторі 1-ї і 2-ї головних компонент Spiral.txt

Література

- [1] Майборода Р. Комп'ютерна статистика: професійний старт.— 2017
- [2] Майборода Р.Є., Сугакова О.В. "Аналіз даних за допомогою пакета R". , 2015. 65 с.
- [3] Франселла Ф., Банністер Д. Новый метод исследования личности. Москва, Прогресс, 1987. 236с.
- [4] James G., Witten D., Hastie T., Tibshirani R. An Introduction to Statistical Learning with Applications in R.— Springer NY 2013.— 440p.
- [5] Kassambara Alboukadel Practical Guide to Cluster Analysis in R.— STHDA, 2017.— 187p.
- [6] von Luxburg Ulrike A Tutorial on Spectral Clustering. Technical Report No. TR-149 .— Max Planck Institute for Biological Cybernetics 2006, 26p.
http://www.kyb.mpg.de/fileadmin/user_upload/files/publications/attachments/luxburg06_TR_v2_4139%5b1%5d.pdf