Лабораторна робота №2
Студента 2 курсу магістратури
Групи "статистика"
Варіант №4

Горбунов Даніел Денисович

19 жовтня 2022 р.

Частина перша.

Вступ.

У даній роботі використано оптимальні кластеризації на даних з першої лабораторної роботи у візуалізації на основі техніки багатовимірного шкалування.

Хід роботи.

Підготовча робота над даними.

Першочергово треба розібратися з тим, що за дані записані у файлі.

```
> # Зчитуємо дані
> data <- read.table("./mult4.txt", header=T)
> # Стандартизуємо дані
> data.std <- scale(data)
```

Тепер можна переходити до основної частини роботи.

Класичне багатовимірне шкалування на даних за варіантом.

Раніше переконалися, що результати кластеризації методу центроїдів та методі медоїдів (на основі евклідової метрики) більш-менш однакові. Тому з точки зору скорочення часу на обчислення, скористаємося першим методом.

```
> # Працюємо з кластеризацією з 3-х та 13-и кластерів
> set.seed(777)
> data.kmeans.3 <- kmeans(data.std, 3, nstart = 50)
> data.kmeans.13 <- kmeans(data.std, 13, nstart = 50)
```

Підрахунок матриць відстаней.

```
> # Обчислення L2-відстаней
> dist.l2 <- dist(data.std, method = "euclidean")
> # Обчислення L1-відстаней
> dist.l1 <- dist(data.std, method = "manhattan")
> # Обчислення minimax-відстаней
> dist.mm <- dist(data.std, method = "maximum")
```

Шкалування на основі ℓ_2-відстані.

```
> scaled.l2 <- cmdscale(d = dist.l2)
```

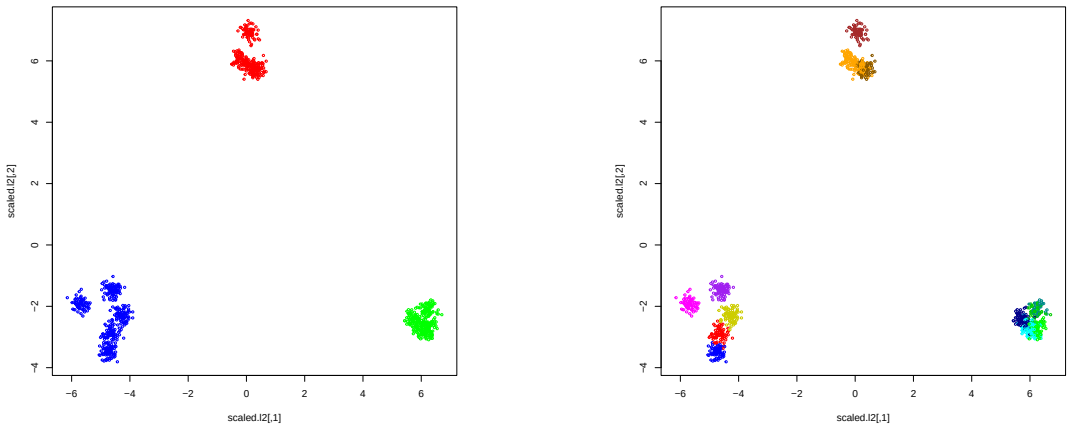

Рис. 1: Діаграма розсіювання даних після MDS. Використовується евклідова відстань.

Шкалування на основі ℓ_1-відстані.

```
> scaled.l1 <- cmdscale(d = dist.l1)
```

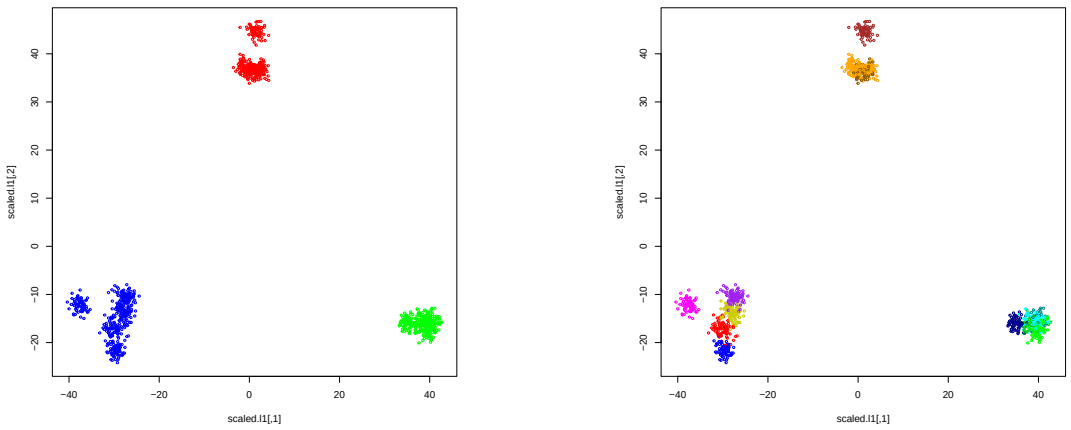

Рис. 2: Діаграма розсіювання даних після MDS. Використовується відстань сіті-блок.

Шкалування на основі мінімаксної відстані.

```
> scaled.mm <- cmdscale(d = dist.mm)
```

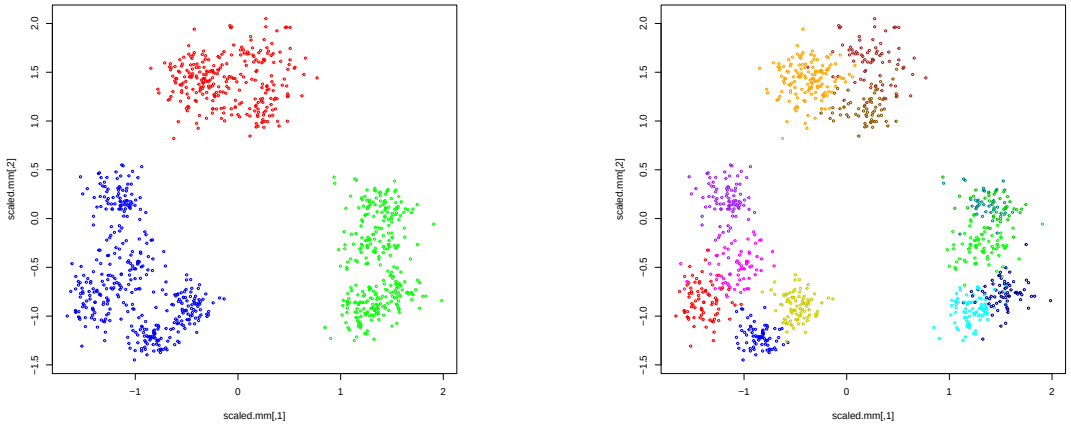

Рис. 3: Діаграма розсіювання даних після MDS. Використовується мінімаксна відстань.

Класичне багатовимірне шкалування на даних про напої.

```
> # Зчитуємо дані про напої
> data.drinks <- read.csv("./drinks_100ml.csv", header=T)
> data.drinks <- data.drinks[,c("Вуглеводи", "Ціна")]
> colnames(data.drinks) <- c("Carbohydrates", "Price")
> # Працюємо з кластеризацією з 2-х та 3-х кластерів
> set.seed(777)
> drinks.kmeans.2 <- kmeans(data.drinks, 2, nstart = 50)
> drinks.kmeans.3 <- kmeans(data.drinks, 3, nstart = 50)
```

Підрахунок матриць відстаней.

```
> # Обчислення L2-відстаней
> drinks.dist.l2 <- dist(data.drinks, method = "euclidean")
> # Обчислення L1-відстаней
> drinks.dist.l1 <- dist(data.drinks, method = "manhattan")
> # Обчислення minimax-відстаней
> drinks.dist.mm <- dist(data.drinks, method = "maximum")
```

Шкалування на основі ℓ_2-відстані.

```
> drinks.scaled.l2 <- cmdscale(d = drinks.dist.l2)
```

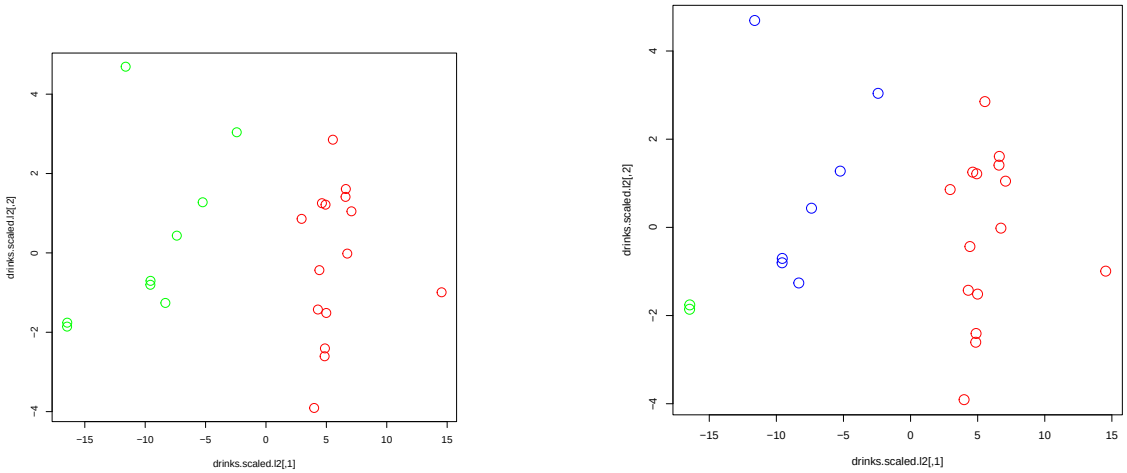

Рис. 4: Діаграма розсіювання даних після MDS. Використовується евклідова відстань.

Шкалування на основі ℓ_1-відстані.

```
> drinks.scaled.l1 <- cmdscale(d = drinks.dist.l1)
```

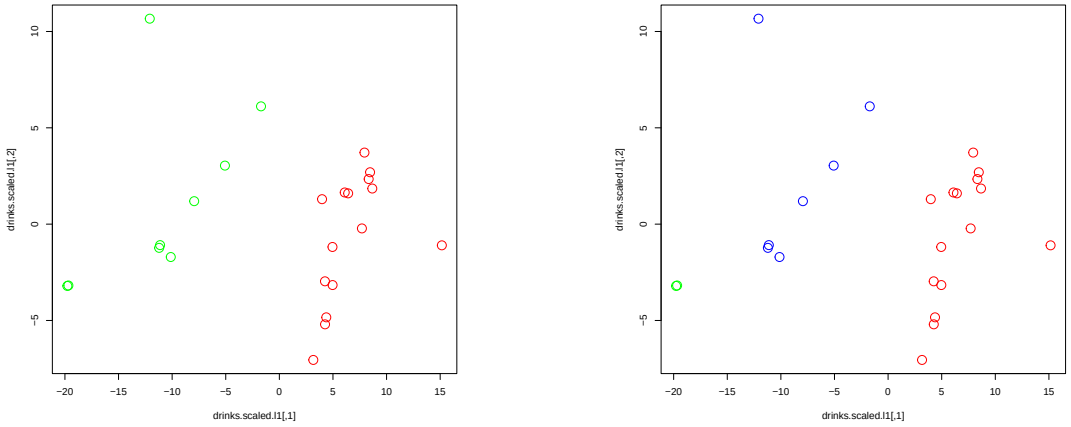

Рис. 5: Діаграма розсіювання даних після MDS. Використовується відстань сіті-блок.

Шкалування на основі мінімаксної відстані.

```
> drinks.scaled.mm <- cmdscale(d = drinks.dist.mm)
```

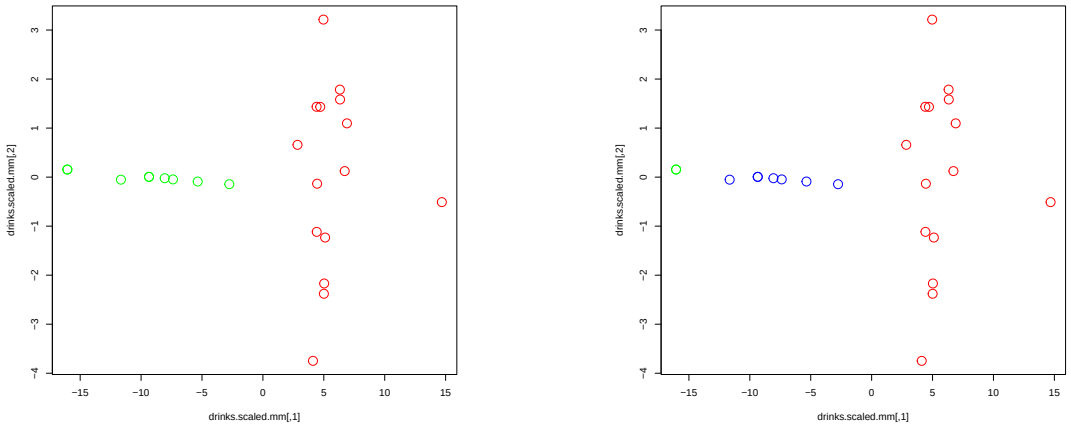

Рис. 6: Діаграма розсіювання даних після MDS. Використовується мінімаксна відстань.

Висновки.

Багатовимірне шкалування непогано спрацювало на мінімаксній відстані – там було відносно легко розгледіти розділення між кластерами. Хоча на інших метриках чітіше спостерігався ефект розбиття на три основні групи кластерів, що побачили у попередній роботі.

Частина друга.

Вступ.

Хід роботи.

Опишемо функцію, яка робить проекцію на перші дві канонічні компоненти на основі шкальованих даних за MDS.

```
> library("CCA")
> # x -- дані
> # dist -- матриця відстаней між об'єктами за x
> # clust -- номери кластерів для кожного об'єкта з x
> # k -- розмірність шкальованого простору
> cc.mds.proj.2d <- function(x, dist, clust, k)
+ {
+   # Класичне багатовимірне шкалування
+   mds.x <- cmdscale(dist, k, eig = TRUE)$points
+   # Кількість кластерів
+   clust.num <- length(levels(as.factor(clust)))
+   # Кількість об'єктів
+   n <- nrow(mds.x)
+   # Застосування CCA для x
+   C <- matrix(
+     data = as.numeric(
+       rep(clust, clust.num) == rep(1:clust.num, each = n)),
+     ncol = clust.num,
+     nrow = n)
+   cc_res<-rcc(mds.x, C, 0.1, 0.1)
+   cc_res$scores$xscores[,1:2]
+ }
```

Шкалування на основі ℓ_2-відстані.

```
> scaled.l2.3 <- cc.mds.proj.2d(
+   x = data.std,
+   dist = dist.l2,
+   clust = data.kmeans.3$cluster,
+   k = 4)
> scaled.l2.13 <- cc.mds.proj.2d(
+   x = data.std,
+   dist = dist.l2,
+   clust = data.kmeans.13$cluster,
+   k = 4)
```

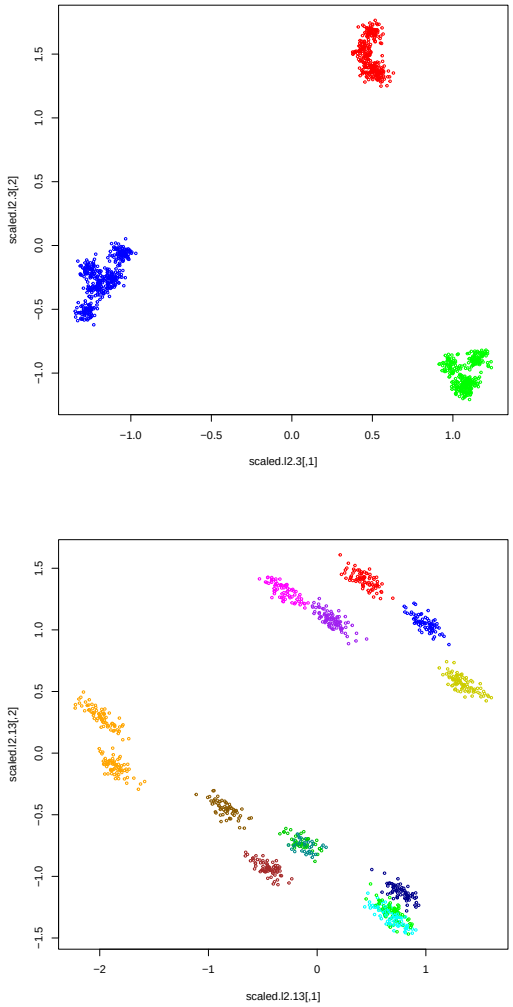

Рис. 7: Діаграма розсіювання даних після MDS + проектування на канонічні компоненти. Використовується евклідова відстань.

Шкалування на основі ℓ_1-відстані.

```
> scaled.l1.3 <- cc.mds.proj.2d(
+   x = data.std,
+   dist = dist.l1,
+   clust = data.kmeans.3$cluster,
+   k = 4)
> scaled.l1.13 <- cc.mds.proj.2d(
+   x = data.std,
+   dist = dist.l1,
+   clust = data.kmeans.13$cluster,
+   k = 4)
```

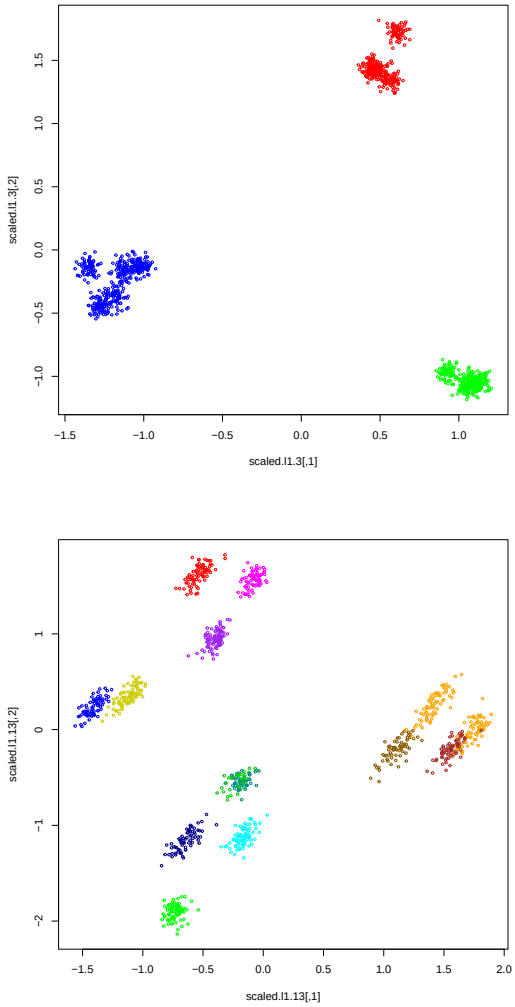

Рис. 8: Діаграма розсіювання даних після MDS + проектування на канонічні компоненти. Використовується відстань сіті-блок.

Шкалування на основі мінімаксної відстані.

```
> scaled.mm.3 <- cc.mds.proj.2d(
+   x = data.std,
+   dist = dist.mm,
+   clust = data.kmeans.3$cluster,
+   k = 4)
> scaled.mm.13 <- cc.mds.proj.2d(
+   x = data.std,
+   dist = dist.mm,
+   clust = data.kmeans.13$cluster,
+   k = 4)
```

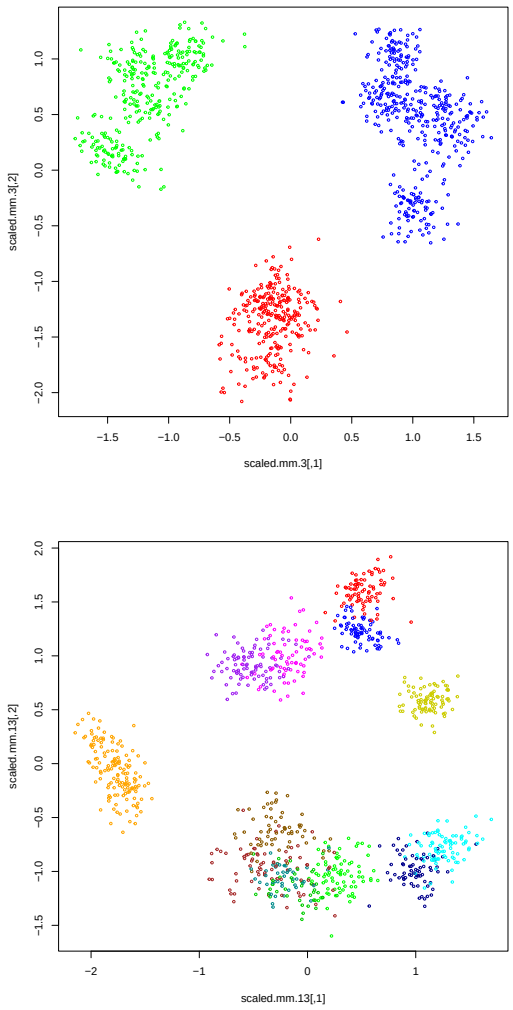

Рис. 9: Діаграма розсіювання даних після MDS + проектування на канонічні компоненти. Використовується мінімаксна відстань.

Висновки.

Візуалізація на основі MDS та проектування на канонічні компоненти спрацювала на даних з варіанту добре. Здається, що розділення було трохи виразнішим.