Національний технічний університет України «КПІ ім. Ігоря Сікорського» Факультет Інформатики та Обчислювальної Техніки

Кафедра інформаційних систем та технологій

Лабораторна робота №6

з дисципліни «Інтелектуальний Аналіз Даних»

на тему

«НЕІЄРАРХІЧНІ МЕТОДИ КЛАСТЕРНОГО АНАЛІЗУ»

Варіант №12

Виконав:

студент групи ІС-02

Плостак І. М.

Київ – 2021

**1. Мета роботи**

Практичне засвоєння неієрархічного кластерного аналізу багатовимірних даних на прикладb методу k-середніх.

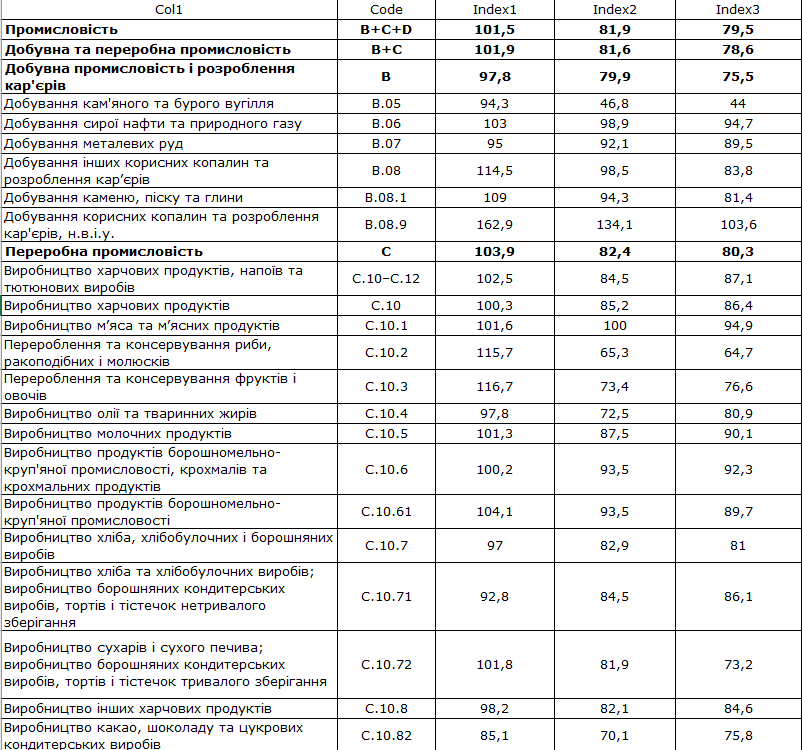
**2. Завдання до роботи**

1. Ознайомитися з конспектом лекцій та рекомендованою літературою, а також додатком Е, що містить короткі теоретичні відомості про неієрархічні методи кластерного аналізу та особливості їх застосування в Matlab.
2. Вивчити функції MATLAB: scatter, gscatter, min, pdist, hist, std.
3. Завантажити дані відповідно до вашого варіанту (табл. 5.1). Побудувати графічне зображення експериментальних даних (діаграму розсіювання). Візуально оцінити кількість кластерів k за побудованим зображенням.
4. Розробити алгоритм кластеризації k-середніх і програмно його реалізувати в середовищі MATLAB.
5. Виконати кластерний аналіз висхідних даних методом k-середніх (параметри методу див. в табл. Е.1). Визначити найбільш оптимальну кількість кластерів k.
6. Розрахувати центри отриманих кластерів. Відобразити графічно знайдені кластери (використати діаграму розсіювання у кольорі).
7. Оформити звіт з роботи.
8. Відповісти на контрольні питання.





**3. Набір даних для обробки (частина)**



**4. Лістинг основних функцій програми**

main.py:

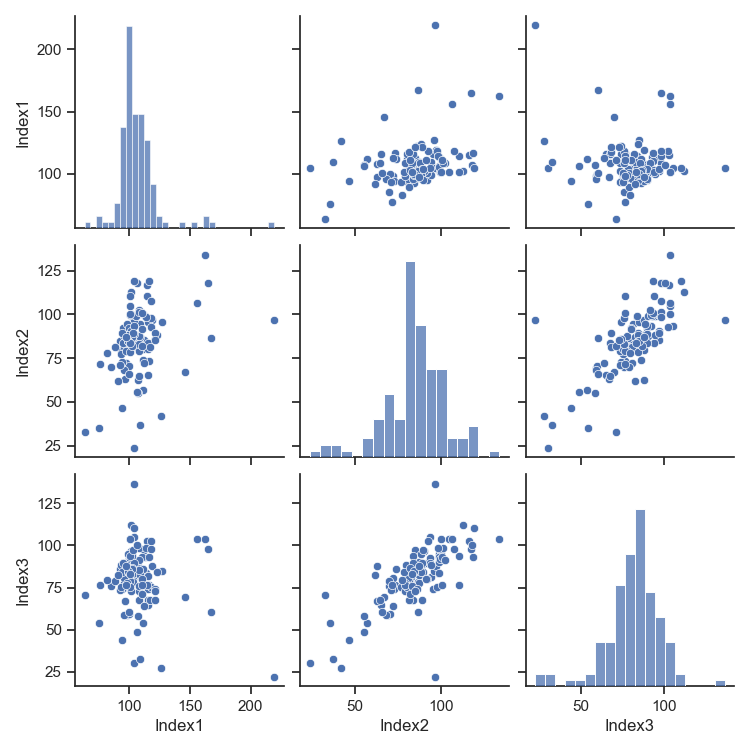
from scipy.spatial.distance import pdist  
from scipy.cluster.hierarchy import \*  
from matplotlib import pyplot as pl  
from statistics import mean  
from itertools import product  
import numpy as np  
import seaborn as sb  
import pandas as pd  
  
import selfkmeans as skm  
  
datasetFull = pd.read\_excel("./datasetFull.xlsx")  
dataset = datasetFull.drop(["Col1"], axis=1)  
sb.set\_theme(style="ticks")  
sb.pairplot(dataset)  
pl.savefig("dataset.png")  
pl.show()  
  
temp\_data = dataset.drop(["Code"], axis=1)  
  
indexes = [[], []]  
for i in range(1, 31):  
 u, iterations, centers = skm.k\_means(i, temp\_data)  
 temp\_index = skm.eff\_index(u, centers, temp\_data)  
 indexes[0].append(i)  
 indexes[1].append(temp\_index)  
  
pl.plot(indexes[0], indexes[1], marker='o')  
pl.xlabel("Amount of Clusters")  
pl.ylabel("Efficiency coefficient")  
pl.grid()  
pl.show()  
  
#bestAmount = indexes[0][indexes[1].index(max(indexes[1]))]  
  
u, iterations, centers = skm.k\_means(6, temp\_data)  
  
print(f"Кількість ітерацій: {iterations}")  
  
dataset["Clusters"] = u[0]  
datasetFull["Clusters"] = u[0]  
dataset = dataset.sort\_values(['Clusters'], ascending=[True])  
datasetFull = datasetFull.sort\_values(['Clusters'], ascending=[True])  
  
print("\nКількість елементів у кластерах:")  
print(dataset.groupby(['Clusters'])['Clusters'].count())  
  
Centers = pd.DataFrame(centers)  
Centers.set\_axis([f'Index{i+1}' for i in range(len(Centers.columns))], axis=1, inplace=True)  
print("\nКоординати центрів кластерів:")  
print(Centers)  
Centers['Clusters'] = ["C" + str(i) for i in range(1, len(Centers) + 1)]  
dataset = pd.concat([dataset, Centers], ignore\_index=True, axis=0)  
sb.pairplot(dataset,  
 hue="Clusters",  
 palette="bright6",  
 markers=[\*["o" for i in range(len(Centers))], \*["D" for i in range(len(Centers))]])  
pl.xlim(0, None)  
pl.savefig("datasetColored.png")  
pl.show()  
  
datasetFull.to\_excel("result.xlsx")

selfkmeans.py:

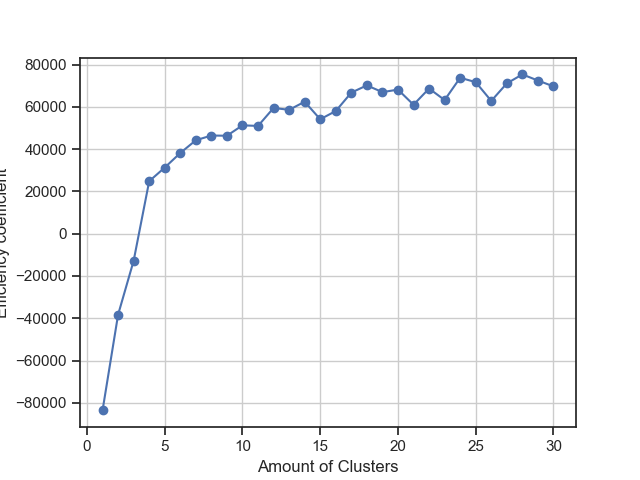
import pandas as pd  
from random import randint  
from itertools import product  
from math import inf  
  
  
def randomNums(length: int, limits: list, repeats: bool = False) -> list:  
 output = []  
 for i in range(length):  
 temp = randint(limits[0], limits[1])  
 if not repeats:  
 while temp in output:  
 temp = randint(limits[0], limits[1])  
 output.append(temp)  
 return output  
  
  
def dist(dot1: list, dot2: list, metric: str = "minkowski", p: int = 4) -> float:  
 if len(dot1) != len(dot2):  
 raise ValueError("Both coords must be in the same dimensions.")  
 if metric == "euclidean":  
 p = 2  
 elif metric == "minkowski":  
 p = 4  
 return pow(sum([(dot2[i] - dot1[i])\*\*p for i in range(len(dot2))]), 1/p)  
  
  
def q3(k: int, data: pd.DataFrame, metric: str = "minkowski") -> float:  
 return sum([sum([dist(i, j, metric=metric)\*\*2 for i, j in  
 product(data[data['Clusters'] == p+1].drop(['Clusters'], axis=1).values,  
 data[data['Clusters'] == p+1].drop(['Clusters'], axis=1).values)]) for p in range(k)])  
  
  
def k\_means(k: int, data: pd.DataFrame, metric: str = "minkowski", epsilon: int = 0.001) -> tuple:  
 centres = [data.values[i] for i in randomNums(k, [0, len(data) - 1])]  
 q\_last = inf  
 u = [[0 for i in range(len(data))] for j in range(2)]  
 m = 1  
 while True:  
 for i in range(len(data)):  
 distances = []  
 for j in range(k):  
 temp1 = data.values[i]  
 temp2 = centres[j]  
 distances.append(dist(temp1, temp2))  
 u[0][i] = distances.index(min(distances)) + 1  
 u[1][i] = min(distances)  
 data['Clusters'] = pd.DataFrame(u).values[0]  
 q\_now = q3(k, data, metric)  
 centres = pd.DataFrame([data.groupby(['Clusters'])[f'Index{i+1}'].mean() for i in range(len(data.columns) - 1)])\  
 .transpose().values.tolist()  
 data.drop(['Clusters'], axis=1, inplace=True)  
 if abs(q\_now - q\_last) < epsilon:  
 break  
 q\_last = q\_now  
 m += 1  
 return u, m, centres  
  
  
def eff\_index(u: list, centers: list, data: pd.DataFrame) -> float:  
 q = len(u[0])  
 k = len(centers)  
 correct\_u = [[0 for j in range(k)] for i in range(q)]  
 for i in range(q):  
 correct\_u[i][u[0][i]-1] = 1  
 x\_mean = [data[f'Index{i+1}'].mean() for i in range(len(data.columns))]  
 return sum([sum([correct\_u[j][i]\*(dist(centers[i], x\_mean)\*\*2 - dist(data.values[j], centers[i])\*\*2)  
 for j in range(q)])  
 for i in range(k)])

**5. Результати роботи програмного забезпечення (частина)**

Візуалізація даних:

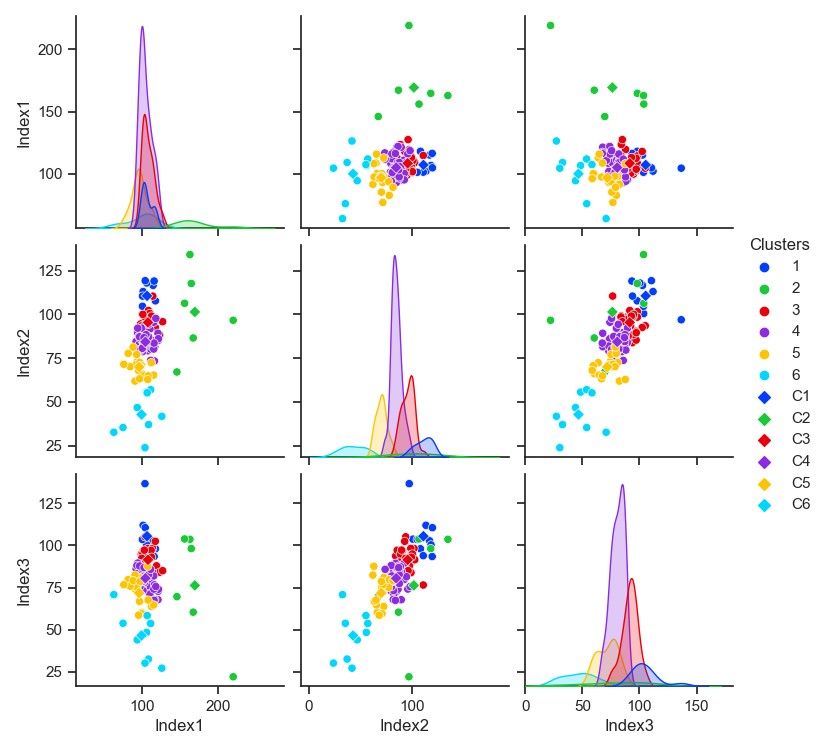


Графік залежності кількості кластерів та Індексу ефективності:

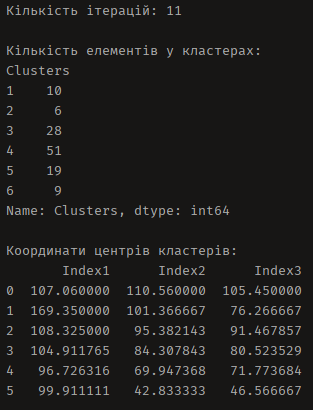


Індекс узято: <http://csc.knu.ua/media/filer_public/38/03/3803002b-e068-4a08-8a6c-a4edc183892a/datamining20170917.pdf>, Сторінки 123-124.

Візуалізація кластерів та їх центрів:



Вивід у консоль:



**6.1. Контрольні питання**

1. В чому полягає задача неієрархічного кластерного аналізу?

*Розподілити дані на кластери, не застосовуючи ієрархічний підхід, який використовує дерева.*

1. Для яких задач обробки експериментальних даних використовуються методи неієрархічного кластерного аналізу?

*За великої кількості об'єктів ієрархічні методи кластерного аналізу непридатні до застосування, через великі обчислювальні витрати і складність інтерпретації дерева кластерів. В таких випадках можуть бути використані неієрархічні методи кластерного аналізу.*

1. В чому сутність алгоритму k-середніх?

*Припустимо, що вже маємо гіпотези відносно кількості кластерів - тобто заздалегідь визначена кількість кластерів k, на які необхідно розбити наявні об'єкти. Серед множини об'єктів обирають k об'єктів в якості початкових центрів кластерів. Для кожного об'єкту розраховують відстані до центрів кластерів, і даний об'єкт відноситься до того кластера, відстань до якого виявилася мінімальною. Після чого для цього кластеру (в якому змінилася кількість об'єктів) розраховується нове положення центру кластеру (як середнє за кожною ознакою Xi) за всіма залученими до кластеру об'єктами. В загальному випадку, в результаті застосування методу k-середніх вихідна множина об'єктів розділюється рівно на k різних кластерів, розташованих на якнайбільших відстанях один від одного.*

1. Наведіть основні етапи ієрархічного кластерного аналізу за методом k-середніх.

* *Крок 1. Ініциалізація початкових параметрів методу.*

*Задати: k - кількість передбачуваних кластерів, матрицю координат центрів кластерів С(0) = {сl(0)}, l = 1,2,…, k (наприклад, обрати k об'єктів із файлу висхідних даних), матрицю U(0), початкове значення функціоналу якості кластеризації Q(0) (деяке велике число) і точність його обчислення (для зупинки алгоритму). Встановити номер ітерації m = 1.*

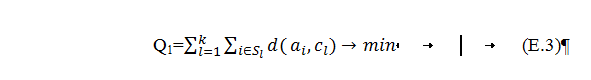
* *Крок 2. Розрахувати відстані від об'єктів n1, n0, …, nN до центрів кла-стерів С(m-1), визначених на попередній ітерації. Заповнити матрицю U(m), виходячи з розташування центрів С(m-1) обчислених на попередній ітерації. Розрахувати значення функціоналу якості кластеризації Q(m) (з урахуванням С(m-1)).*
* *Крок 3. Перевірити умову зупинки алгоритму | Q(m) – Q(m-1) | (блок 3). При цьому оцінюється, чи призвело нове об'єднання об'єктів в кластери до суттєвого покращення якості кластеризації. Якщо умова виконується, то завершити процес кластерізації. Інакше перейти на крок 4.*
* *Крок 4. Розрахувати нове положення центрів кластерів С(m) як середнє аріфметичне за координатами об'єктів, що входять у відповідні кластери.*
* *Крок 5. Встановити Q(m-1) = Q(m) і перейти на крок 2 (нової ітерації) з m = m + 1.*

1. З якою метою в алгоритмі k-середніх вводиться матриця розбиттів U?

*Для зберігання інформації про приналежність об'єкту до деякого кластеру.*

1. Які критерії зупинки автоматичної кластеризації використовуються на практиці?

*1. Сума відстаней до центрів кластерів:*

**

*l – номер кластеру, l = 1, 2, …, k, ni – вектор ознак i-го об'єкту в l-ом кластері, Sl – множина об'єктів в l-ому кластері.*

*2. Сума квадратів відстаней до центрів кластерів:*

**

*l – номер кластеру, l = 1, 2, …, k, ni – вектор ознак i-го об'єкту в l-ому кластері, Sl – множина об'єктів в l-ому кластері.*

*3. Сума внутрішньоикластерних відстаней між об'єктами:*

**

1. Яким чином оцінити кількість кластерів в алгоритмі k-середніх?

*1. Методом ліктя.*

*2. За допомогою спеціальних індексів ефективності.*

**6.2. Висновки**

В результаті виконання лабораторної роботи ми познайомилися з неієрархічним кластерним аналізом даних.

У ході аналізу представленого набору даних було виявлено, що неієрархічний метод k-середніх, порівняно з ієрархічним методом, зовсім інакше розбив дані на кластери і у данному випадку вже немає одного великого кластеру.

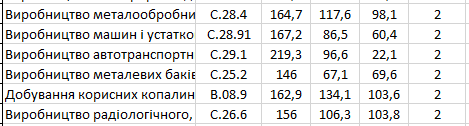
Також, що дуже важливо зазначити, метод k-середніх оснований на випадкових значеннях початкових центрів кластерів, тож загалом він постійно дає різні результати, які кожного разу «звалюються» у локальні мінімуми, що кожного разу дає різну, але приблизно однакову картину розбиття.

Коротко про кластери:

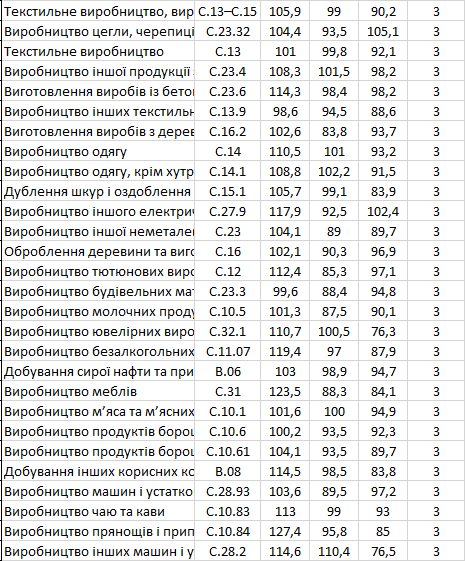
1. Перший кластер – загалом великі значення:



1. Другий кластер – великі значення першого параметру:



1. Третій кластер – набір великих значень у першому параметрі та середніх у другому та третьому:



1. Четвертий кластер – знову найбільший, середні значення. Що цікаво та цілком логічно – у цей кластер знову потрапили загальні значення, по узагальненим типам промисловостей:



1. П’ятий кластер – значення по другому та третьому параметру нижче середнього, той кластер, що знову ніби «відділився» від 4го:



1. Шостий кластер – дуже маленькі значення одного якого-сь параметру:

