Лабораторна робота № 22

Тема: “Навчання без вчителя. Кластеризація і класифікація (розпізнавання) образів методом k-середніх”

Мета:

1. *Поновити та поглибити знання щодо моделі навчання без вчителя (unsupervisered learning) і задачі кластеризації.*
2. *Поновити та поглибити знання щодо методу k - середніх кластеризації даних.*

Теоретичний мінімум

Що таке навчання без вчителя?

Інтелект, це пара функцій сумісного застосування - “розпізнавач” і “виконавець”. Перша з них розпізнає належність об'єкту, що подається на вхід функції, цільовому класу (задача класифікації). Друга функція, це автомат, який на підставі результатів розпізнавання, активує той, чи інший, протокол дій. У цьому процесі основною і найбільш складною є перша функція.

В попередніх лабораторних роботах розглядався процес побудови “розпізнавача” на основі моделі машинного навчання з вчителем (supervised learning), що передбачає наявність, так званих, розмічених даних. Розмічені дані являють собою набір образів об'єктів, наперед розбитих за характеристиками на цільові класи.

Отже клас, це множина об'єктів ототожнених за своїми характеристиками. Об'єктам кожного із класів назначена певна марка. Марки різних класів попарно різні між собою. Набір даних, розмічених таким чином, називається навчальною вибіркою. Будується навчальна вибірка на підставі досвіду розв'язування подібних задач.

Серед відомих методів навчання “розпізнавачів” з вчителем, у попередніх лабораторних роботах, присвячених задачам класифікації і пошуку регресії були розглянуті метод *k*-найближчих сусідів та метод опорних векторів.

Проте, при розв'язуванні реальних задач далеко не завжди є можливість наперед сформувати навчальні вибірки. У таких випадках метод повинен мати здатність самостійного розбиття вхідних даних на класи, генерування марок класів і відповідного маркування вхідних об'єктів.

Класи, сформовані певним алгоритмом називаються кластерами, а задача розбиття вхідних даних на кластери називається задачею кластеризації.

Процес побудови моделі навчання без використання навчальної вибірки називається навчанням без вчителя (unsupervised learning).

Серед відомих методів навчання “розпізнавачів” без вчителя, у попередніх лабораторних роботах був розглянутий метод *k*-середніх (*k*-means).

- Ідея методу викладена у лабораторній роботі 17. Цей метод знаходить широке використання у багатьох, зокрема сегментування (кластеризація) ринку, торгівля акція, оброблення природних мов, машинний зір і т.і.

ЗАВДАННЯ:

Завдання полягає дещо у повторюванні завдання роботи 17, але є більш узагальненим. Узагальнення полягає у реалізації алгоритму кластеризації, більш наближеної до реальних задач. Для спрощення і можливості візуалізації результатів, розглядаються об'єкти із двома характеристиками. Цей алгоритм, як і у роботі 17, є параметричним. Тобто, для роботи алгоритму треба задати кількість кластерів на вхідній множині об'єктів. Якість розв'язку задачі кластеризації досягається шляхом варіації значень вхідних параметрів алгоритму.

Проте, розвинення цього алгоритму надають більші можливості для розв'язування задач кластеризації. По-перше, можливість не задавати кількість кластерів, а визначати самостійно їх оптимальну кількість безпосередньо у процесі навчання. По-друге, самостійно обчислювати значення якості навчання. Цей матеріал є змістом наступних робіт.

Завдання цієї роботи має такий зміст:

А) Описати вхідні дані, які містяться наданому файлі **data_clustering.txt .**

Б) Проаналізувати наданий код (Додаток) і розробити його блок-схему першого рівня. .

В) Поновити наданий код, налаштувати і запустити на виконання алгоритм k-середніх навчання без вчителя з використанням наданих даних і значенням параметрів за замовчанням.

Г) Змінити значення параметрів алгоритму і на підставі результатів візуалізації результатів, зробити висновок що до якості розв'язку у кожному випадку.

Контрольні питання:

1. Що таке навчальна вибірка?
2. Що таке ‘supervised learning’ і у чому сутність цього поняття?
3. Що таке модель ‘unsupervised learning’ і у чому сутність цього поняття?
4. У чому полягає ідея алгоритму *k*-середніх кластеризації даних?

Додаток.

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn import metrics

# data download
X = np.loadtxt('data_clustering.txt', delimiter= ',')

# numer of clusters
num_clusters = 5

# data visualisation
plt.figure()
plt.scatter(X[:,0], X[:,1], marker='o', facecolors='none', edgecolor='black', s=80)
x_min, x_max = X[:,0].min()-1, X[:,0].max()+1
y_min, y_max = X[:,1].min()-1, X[:,1].max()+1
plt.title('input data')
plt.xlim(x_min, x_max)
plt.ylim(y_min, y_max)
plt.xticks(())
plt.yticks(())
#1. dataset visualization
#plt.show()

print(x_min, x_max, y_min, y_max)

#create object KMeans
kmeans= KMeans(init= 'k-means++', n_clusters=num_clusters, n_init= 10)
#kmeans= KMeans(n_clusters= 6)
#learning kmens
kmeans.fit(X)

# visualization borders of clusters
step_size = 0.01
#grid
#x_min, x_max = X[:,0].min()-1, X[:,0].max()+1
#y_min, y_max = X[:,1].min()-1, X[:,1].max()+1

x_vals, y_vals = np.meshgrid(np.arange(x_min, x_max, step_size),
                             np.arange(y_min, y_max, step_size))
```

(продовження коду)

```
# grid point mark prediction
output= kmeans.predict(np.c_[x_vals.ravel(), y_vals.ravel()])

# visualiszation of region
output = output.reshape(x_vals.shape)
plt.figure()
plt.clf()
plt.imshow(output,interpolation= 'nearest',
           extent= (x_vals.min(), x_vals.max(),
                    y_vals.min(), y_vals.max()),
           cmap= plt.cm.Paired,
           aspect= 'auto',
           origin= 'lower')
#2.
#plt.show()

# visualiszation of region + scatter
plt.scatter(X[:,0], X[:,1], marker= 'o', facecolor= 'none',
            edgecolor='black', s= 80)
#3.
#plt.show()

#4. + center of clusters
cluster_centers= kmeans.cluster_centers_
plt.scatter(cluster_centers[:,0], cluster_centers[:,1],
            marker= 'o', s=200, linewidths= 4, color='black',
            zorder= 12, facecolor= 'black')
plt.title('Bordes of clusters')
plt.xlim(x_min, x_max)
plt.ylim(y_min, y_max)
plt.xticks()
plt.yticks()
plt.show()
```