ĐẠI HỌC ĐÀ NẴNG
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA ĐIỆN TỬ - VIỄN THÔNG

BÁO CÁO BÀI TẬP 3
NHÓM 10

Sinh viên thực hiện: Đinh Văn Quang 20DT1
Hà Phước Phúc 20DT2
Nguyễn Văn Qúy 21DT2
Lớp học phần: 20.38
Giảng viên hướng dẫn: TS. Hoàng Lê Uyên Thục

Đà Nẵng, 9/2024

Bài 3: Phân loại rượu vang Ý bằng phương pháp template matching

Cơ sở dữ liệu: https://github.com/MukeshTirupathi/Wine-Classifier-Italy?tab=readme-ov-file

1. Mô tả phương pháp Template Matching

- Template Matching là một kỹ thuật tìm kiếm các vùng của dữ liệu hoặc hình ảnh tương tự như một mẫu đã train. Mẫu là một dữ liệu nhỏ với một số đặc tính nhất định. Mục tiêu của việc so khớp mẫu là tìm mẫu trong dữ liệu hoặc hình ảnh. Để tìm thấy nó, người dùng phải cung cấp hai dữ liệu đầu vào: nguồn là dữ liệu để tìm mẫu và dữ liệu mẫu là dữ liệu sẽ được tìm thấy trong dữ liệu nguồn.

- Về cơ bản, nó là một phương pháp tìm kiếm và tìm vị trí của dữ liệu mẫu trong hình ảnh hoặc dữ liệu lớn hơn. Ý tưởng ở đây là tìm các vùng giống hệt nhau của dữ liệu hoặc hình ảnh khớp với mẫu đã cung cấp.

- Thuật toán K-Nearest Neighbors (KNN) giúp xác định các điểm hoặc nhóm gần nhất cho một điểm truy vấn, sử dụng công thức khoảng cách Manhattan. Thuật toán này phân loại vector test bằng cách gán nó vào class của mẫu gần nhất trong tập dữ liệu huấn luyện

$$d(x,y) = \sum_{i=1}^{n} |x_i - y_i|$$

Bước 1: Thu thập và chuẩn bị dữ liệu:

Có 3 lớp loại rượu vang gồm có 3 class:

- Class 1: 59 mẫu
- Class 2: 71 mẫu
- Class 3: 48 mẫu

Tổng cộng có 178 mẫu rượu vang

- Mỗi mẫu có 13 đặc trưng (chiều): alcohol, malic_acid, ash, alcalinity_of_ash, magnesium, total_phenols, flavanoids, nonflavanoid_phenols, proanthocyanins, color_intensity, hue, od280/od315_of_diluted_wines, proline.

Phân chia dữ liệu:

- Chọn ngẫu nhiên 1 mẫu từ 178 mẫu trong bộ tập dữ liệu để làm mẫu test, còn lại 177 mẫu làm tập train.

Bước 2: Tính toán khoảng cách

- Đo khoảng cách giữa mẫu kiểm tra và từng mẫu trong tập huấn luyện.
- Sử dụng công thức khoảng cách Manhattan

Bước 3: So sánh với mẫu gần nhất:

- Sau đó lấy giá trị khoảng cách nhỏ nhất để xác định loại rượu vang thuộc lớp nào.

Bước 4: Gán nhãn

- Nhãn của mẫu trong tập huấn luyện có khoảng cách nhỏ nhất sẽ được gán cho mẫu kiểm tra.

2. Mô tả cơ sở dữ liệu. Nêu cách chọn tập train, chọn dữ liệu test

- Mô tả cơ sở dữ liệu:

+ Bộ dữ liệu về rượu vang từ kho lưu trữ máy học UCI, gồm 178 trường hợp là kết quả phân tích hóa học của các loại rượu vang được trồng trong cùng một khu vực ở Ý nhưng có nguồn gốc từ ba giống khác nhau.

+ Phân tích xác định số lượng của 13 thành phần được tìm thấy trong mỗi loại trong ba loại rượu vang, bao gồm: *Alcohol, Malic acid, Ash Alcalinity of ash, Magnesium, Total phenols, Flavanoids, Nonflavanoid phenols, Proanthocyanins, Color intensity, Hue, OD280/OD315 of diluted wines, Proline.*

+ Tất cả các thuộc tính đều liên tục.

+ Tập tin wine.data lưu trữ 3 classes (class1: 59, class2: 71, class3: 48), trong đó mỗi hàng là bộ 13 đặc trưng của một chai rượu (mẫu).

- Cách chọn dữ liệu train:

+ Chọn bất kỳ 1 trong 178 mẫu để test, còn lại 177 mẫu để train.

3. Tiến hành phân loại rượu vang Ý: giải thích cách làm, code. Lưu ý copy và dán code vào đây, có giải thích đầy đủ

- Đoạn code thực thi chương trình:

```python
# 3 giong ruou vang (3 class), class 1: 59, class 2: 71, class 3: 48
# Tong cong có 178 chai ruou, moi chai rouu có 13 đac trung khac nhau
# Chon 1/178 chai de test, 177 chai de train
import pandas as pd
import numpy as np

columns = ['class', 'Alcohol', 'Malic acid', 'Ash', 'Alcalinity of ash', 'Magnesium',
           'Total phenols', 'Flavanoids', 'Nonflavanoid phenols', 'Proanthocyanins',
           'Color intensity', 'Hue', 'OD280/OD315', 'Proline']

data_wine = pd.read_csv("wine/wine.data", header=None, names=columns)

data_wine = data_wine.copy()  # Tạo bản sao của DataFrame

test_vector = np.array(list(map(float, input("Type input 13 features  of vector test: ").split(','))))

# Tính Manhattan distance cho tất cả các mẫu dữ liệu
def manhattan_distance(row):
    return np.sum(np.abs(test_vector - row[1:]))  # Bỏ qua cột 'class'

data_wine['distance'] = data_wine.apply(manhattan_distance, axis=1)

print("Khoảng cách từ vector test đến từng mẫu:")
print(data_wine[['class', 'distance']])

# Tìm hàng có khoảng cách nhỏ nhất
nearest_row = data_wine.loc[data_wine['distance'].idxmin()]

print(f"\nKết quả: Vector test thuộc class {nearest_row['class']} với khoảng cách nhỏ nhất là {nearest_row['distance']:.2f}")

print(data_wine.head())

```

- Giải thích cách làm, code:

Bước 1. Import các thư viện cần thiết:

Pandas để xử lý dữ liệu dạng bảng

Numpy để thực hiện các phép tính số học, cung cấp hỗ trợ cho các ma trận và mảng lớn, đa chiều.

Bước 2. Định nghĩa danh sách các cột cho bộ dữ liệu rượu.

Bước 3. Đọc dữ liệu từ file CSV vào DataFrame data_wine, sử dụng tên cột đã định nghĩa.

Bước 4. Nhập vector test gồm bộ 13 số ngăn cách nhau bởi dấu phẩy và chuyển đổi input từ chuỗi thành mảng số thực

```python
test_vector = np.array(list(map(float, input("Type input 13 features  of vector test: ").split(','))))
```

Bước 5. Định nghĩa hàm tính Manhattan distance

```python
# Tính Manhattan distance cho tất cả các mẫu dữ liệu
def manhattan_distance(row):
    return np.sum(np.abs(test_vector - row[1:]))
```

Bằng cách tính tổng các giá trị tuyệt đối của hiệu giữa vector test và mỗi hàng (mẫu) dữ liệu (bỏ qua cột 'class').

$$d(x,y) = \sum_{i=1}^{n} |x_i - y_i|$$

Bước 6. Áp dụng hàm khoảng cách Manhattan cho mỗi hàng trong DataFrame:

```
18
19  data_wine['distance'] = data_wine.apply(manhattan_distance, axis=1)
20
```

Kết quả khoảng cách được lưu vào cột mớii 'distance' trong bản copy data_wine

Bước 7. In ra khoảng cách từ vector test đến từng mẫu.

Bước 8. Trong cột distance, tìm hàng có khoảng cách nhỏ nhất:

```
24  # Tìm mẫu có khoảng cách nhỏ nhất
25  nearest_row = data_wine.loc[data_wine['distance'].idxmin()]
26
```

Sử dụng hàm idxmin() để tìm chỉ số của hàng có khoảng cách nhỏ nhất, sau đó lấy toàn bộ thông tin của hàng đó

Bước 9. In kết quả, cho biết vector test thuộc class nào và khoảng cách nhỏ nhất.

4. Ghi lại kết quả

Kết quả sau khi chạy code

```
Type input vector test: 12.93,3.8,2.65,18.6,102,2.41,2.41,.25,1.98,4.5,1.03,3.52,210
Khoảng cách từ vector test đến từng hàng:
     class  distance
0        1    889.54
1        1    853.76
2        1    981.32
3        1   1292.51
4        1    546.89
..     ...       ...
173      3    550.82
174      3    553.26
175      3    657.06
176      3    661.55
177      3    373.48

[178 rows x 2 columns]

Kết quả: Vector test thuộc class 2.0 với khoảng cách nhỏ nhất là 92.37

Process finished with exit code 0
```

5. Nhận xét

Mẫu test:

- Vector input được nhập là là vector đặc trưng của một chai rượu với 13 đặc trưng ví dụ: **12.93, 3.8, 2.65, 18.6, 102, 2.41, 2.41, .25, 1.98, 4.5, 1.03, 3.52, 210**

Khoảng cách Manhattan:

- Kết quả của từng hàng dữ liệu (vector rượu) được tính bằng cách sử dụng khoảng cách Manhattan từ vector test đến các vector trong tập huấn luyện.
- Khoảng cách này thể hiện mức độ khác biệt giữa các đặc trưng của mẫu test với các mẫu trong tập huấn luyện.
- Giá trị khoảng cách nhỏ nhất (92.37) được tìm thấy ở một mẫu thuộc class 2, nghĩa là mẫu test gần nhất với các mẫu rượu trong lớp 2.

Dự đoán kết quả:

- Kết quả hiển thị: "Vector test thuộc class 2.0 với khoảng cách nhỏ nhất là 92.37" có nghĩa là mẫu test có đặc điểm gần nhất với rượu vang thuộc class 2, nên mô hình dự đoán mẫu test thuộc về lớp 2.

Tài liệu tham khảo

[1] https://www.youtube.com/watch?v=jlpaqf_M8xU

[2] https://www.geeksforgeeks.org/k-nearest-neighbours/

[3] https://github.com/MukeshTirupathi/Wine-Classifier-Italy/blob/main/LC_MukeshTirupathi_2016515.ipynb

Source code:
https://github.com/haphucc/Module1_AI_24_25/tree/main/Ruou_Vang_Y