KHAI PHÁ DỮ LIỆU

Trường Đại học Nha Trang
Khoa Công nghệ thông tin
Bộ môn Hệ thống thông tin
Giáo viên: TS.Nguyễn Khắc Cường

CHỦ ĐỀ 3

TIỀN XỬ LÝ DỮ LIỆU

Tiền xử lý dữ liệu

- Giới thiệu:
  - Với các dataset vừa và nhỏ, các kỹ thuật preprocessing của chủ đề 2 thường được áp dụng.
  - Riêng các dataset lớn (số lượng feature lớn, số lượng sample lớn), cần phải áp dụng thêm một kỹ thuật rất quan trọng, đó là: DIMENSION REDUCTION
  - Dimension reduction = giảm bớt số lượng feature
  - Lý do:
    - Tăng tốc độ thực hiện các giải thuật KPDL
    - Giảm kích thước lưu trữ
    - Có thể tăng độ chính xác của các kết quả KPDL

Tiền xử lý dữ liệu

- Phân tích cấu trúc, thuộc tính của larger dataset
  - Các sample, feature, value có thể được phân tích theo một cách nào đó nhằm xóa bớt, thêm vào, biến đổi để thay đổi cấu trúc của dataset trước khi áp dụng các giải thuật KPDL → làm tăng hiệu quả của các giải thuật KPDL.
  - Đặc biệt, số lượng feature (thuộc tính, đặc trưng) lớn thường làm giảm hiệu quả của các kỹ thuật KPDL.
  - Do đó, việc thay đổi các tập hợp các feature gốc của dataset thành một tập hợp feature mới, nhỏ gọn hơn (dimension reduction) là một trong các bước preprocessing quan trọng, nhất là đối với các datatset lớn.

Tiền xử lý dữ liệu

- Phân tích cấu trúc, thuộc tính của larger dataset
  - Mục tiêu của dimension reduction:
    - Computing time
    - Predictive / Descriptive accuracy
    - Representation of the data-mining model
- Trích đặc trưng
  - Feature selection:
    - Sử dụng một idea nào đó để lựa chọn một tập feature nhỏ hơn từ tập feature gốc.
    - Các feature được lựa chọn là các feature quan trọng, có thể đại diện cho dataset mà vẫn đảm bảo quality của dataset không hoặc ít bị thay đổi.

Tiền xử lý dữ liệu

- Trích đặc trưng
  - Feature extraction / Feature transformation:
    - Feature extraction:
      - Sau bước feature selection, các feature quan trọng được trích xuất và giữ lại, các feature ít quan trọng bị loại bỏ.
    - Feature transformation:
      - Một số feature gốc được nhập lại (sử dụng một phép tính nào đó) và biến thành một feature mới.
- Số chiều của dữ liệu
  - Số chiều = dimenionality = số lượng các feature

Tiền xử lý dữ liệu

- Các giải thuật giảm số chiều
  - Giải thuật dùng công cụ thống kê:
    - Giải thuật này có thể dùng trong trường hợp không có thông tin cụ thể nào về hình dạng của đường cong (mặt cong) biểu diễn phân bố của dữ liệu.
    - Giả sử có dataset như sau
      - một số sample thuộc 2 lớp A và B
      - số lượng sample mỗi lớp là n_1 và n_2

X	Y	C
0.3	0.7	*A*
0.2	0.9	*B*
0.6	0.6	*A*
0.5	0.5	*A*
0.7	0.7	*B*
0.4	0.9	*B*

Tiền xử lý dữ liệu

X	Y	C
0.3	0.7	*A*
0.2	0.9	*B*
0.6	0.6	*A*
0.5	0.5	*A*
0.7	0.7	*B*
0.4	0.9	*B*

- Các giải thuật giảm số chiều
  - Giải thuật dùng công cụ thống kê:
    - Tập con các sample thuộc mỗi lớp là:

$$X_A = \{0.3, 0.6, 0.5\}, \quad X_B = \{0.2, 0.7, 0.4\}$$
$$Y_A = \{0.7, 0.6, 0.5\}, \quad Y_B = \{0.9, 0.7, 0.9\}$$

    - Thực hiện các phép tính test như sau:

$$\mathrm{SE}(X_A - X_B) = \sqrt{\left(\frac{\mathrm{var}(X_A)}{n_1} + \frac{\mathrm{var}(X_B)}{n_2}\right)} = \sqrt{\left(\frac{0.0233}{3} + \frac{0.06333}{3}\right)} = 0.1699$$

$$\mathrm{SE}(Y_A - Y_B) = \sqrt{\left(\frac{\mathrm{var}(Y_A)}{n_1} + \frac{\mathrm{var}(Y_B)}{n_2}\right)} = \sqrt{\left(\frac{0.01}{3} + \frac{0.0133}{3}\right)} = 0.0875$$

Tiền xử lý dữ liệu

X	Y	C
0.3	0.7	*A*
0.2	0.9	*B*
0.6	0.6	*A*
0.5	0.5	*A*
0.7	0.7	*B*
0.4	0.9	*B*

- Các giải thuật giảm số chiều
  - Giải thuật dùng công cụ thống kê:
    - Thực hiện các phép tính test như sau:

$$\frac{|\text{mean}(X_A)-\text{mean}(X_B)|}{\text{SE}(X_A-X_B)}=\frac{|0.4667-0.4333|}{0.1699}=0.1961<0.5$$

$$\frac{|\text{mean}(Y_A)-\text{mean}(Y_B)|}{\text{SE}(Y_A-Y_B)}=\frac{|0.6-0.8333|}{0.0875}=2.6667>0.5$$

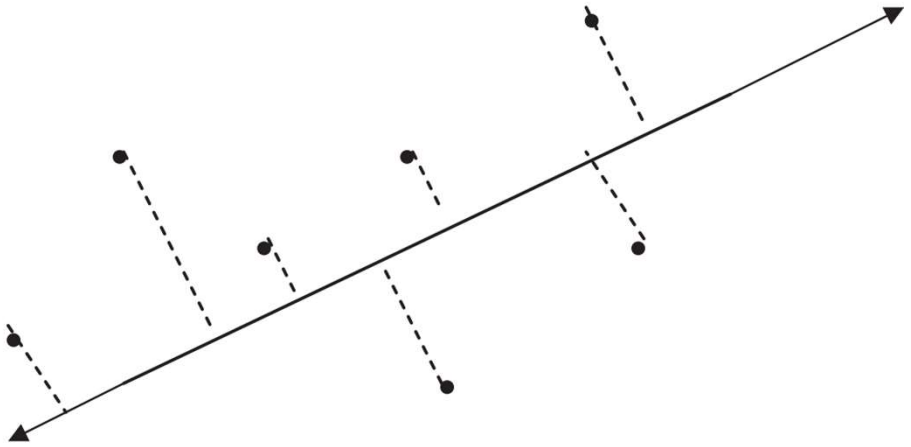
    - Ý nghĩa của công thức test trên:
      - Các value thuộc feature X có mức độ tập trung gần với mean (0.5) → mức độ discrimination thấp
      - Các value thuộc feature Y có mức độ phân tán cao so với mean (0.5) → mức độ discrimination cao
    - Kết luận: Có thể loại feature X

Tiền xử lý dữ liệu

- Các giải thuật giảm số chiều
  - Giải thuật PCA: được hiểu **ở mức độ ứng dụng** như sau:
    - PCA = Principal Components Analysis
    - Còn gọi là phương pháp Karhunen–Loeve (K–L)
    - Là một trong các giải thuật giảm số chiều tuyến tính
    - Dùng công cụ thống kê
    - Rất hay dùng cho các dataset lớn
    - Giả thiết quan trọng:
      - High information = high variance
    - Ý tưởng: $Y = A \cdot X$ → tìm ma trận A sao cho feature X được biến đổi thành feature Y có phương sai lớn nhất
    - Lấy feature Y làm "the first principal component"
      → dùng làm trục tọa độ đầu tiên trong không gian mới

Tiền xử lý dữ liệu

- Các giải thuật giảm số chiều
  - Giải thuật PCA: được hiểu **ở mức độ ứng dụng** như sau:
    - Lấy feature Y làm "the first principal component"
      → dùng làm trục tọa độ đầu tiên trong không gian mới
    - Trục tọa độ này là trục đi theo một hướng mà hướng này làm cho phương sai của các data point là cực đại
    - Nếu chiếu các data point lên trục này thì tổng bình phương các khoảng cách từ các data point đến các điểm chiếu của nó là cực tiểu



Tiền xử lý dữ liệu

- Các giải thuật giảm số chiều
  - Giải thuật PCA: được hiểu **ở mức độ ứng dụng** như sau:
    - Vấn đề:
      - Rất khó để tìm được một cách trực tiếp ma trận A thỏa điều kiện trên (cực đại phương sai)
    - Ý tưởng giải quyết: tính xấp xỉ, cụ thể là
      - Tính covariance matrix

$$S_{n\times n} = \frac{1}{(n-1)}\left[\sum_{j=1}^{n}\left(x_j - x'\right)^{\mathrm{T}}\left(x_j - x'\right)\right] \qquad x' = (1/n)\sum_{j=1}^{n} x_j$$

      - Tìm giá trị riêng (eigenvalues) của ma trận S
      - Chọn ra m giá trị riêng lớn nhất
      - Tìm m vector riêng (eigenvector) tương ứng với m giá trị riêng đó

Tiền xử lý dữ liệu

- Các giải thuật giảm số chiều
  - Giải thuật PCA: được hiểu **ở mức độ ứng dụng** như sau:
    - Dùng m vector riêng trên để ghép lại thành các cột của ma trận xấp xỉ với ma trận A cần tìm
    - Dùng ma trận đó như một linear transformation để biến đổi các data point gốc từ không gian n-dimensional sang không gian mới m-dimensional → trong không gian mới này làm cho độ phân tán của các data point là cực đại.
    - Tóm tắt:
      - Tìm covariance matrix trong không gian n-dimensional $S_{n \times n}$
      - Tìm các giá trị riêng của ma trận S và sắp xếp giảm dần

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n \geq 0$$

      - Chọn m giá trị riêng có giá trị lớn nhất

Tiền xử lý dữ liệu

- Các giải thuật giảm số chiều
  - Giải thuật PCA: được hiểu **ở mức độ ứng dụng** như sau:
    - Tóm tắt:
      - Tính m vector riêng tương ứng với m giá trị riêng đó

$$e_1, e_2, \dots e_m$$

      - Dùng m vector riêng đó để xây dựng thành ma trận biến đổi các sample vector (data point) từ n-dimensionakl space sang m-dimensional space
      - Các vector riêng đó được gọi là các principal axes
    - Ví dụ:
      - Một ứng dụng face detection dùng hình chụp mẫu mặt người của Yale Face Database có size = 200 x 200 = 40.000 dimension
      - Nhận xét: một sample (một hình) có số lượng feature rất lớn
      - Yêu cầu: giảm bớt các feature nhưng vẫn đủ quality

Tiền xử lý dữ liệu

- Các giải thuật giảm số chiều
  - Giải thuật PCA: được hiểu **ở mức độ ứng dụng** như sau:
    - Ví dụ:
      - Một số hình mẫu trong Yale Face Database

      - Idea:
        - Dùng PCA để biến đổi các sample từ không gian 40,000 chiều sang không gian 116 x 98 = 11.368 chiều

Tiền xử lý dữ liệu

- Các giải thuật giảm số chiều
  - Giải thuật PCA: được hiểu **ở mức độ ứng dụng** như sau:
    - Ví dụ:
      - Code Python: tham khảo tại https://machinelearningcoban.com/2017/06/21/pca2/

```
import numpy as np
from scipy import misc                          # for loading image
np.random.seed(1)

# filename structure
path = 'unpadded/' # path to the database
ids = range(1, 16) # 15 persons
states = ['centerlight', 'glasses', 'happy', 'leftlight',
          'noglasses', 'normal', 'rightlight','sad',
          'sleepy', 'surprised', 'wink' ]
prefix = 'subject'
surfix = '.pgm'

# data dimension
h = 116 # hight
w = 98 # width
D = h * w
N = len(states)*15
K = 100

# collect all data
X = np.zeros((D, N))
cnt = 0
for person_id in range(1, 16):
    for state in states:
        fn = path + prefix + str(person_id).zfill(2) + '.' + state + surfix
        X[:, cnt] = misc.imread(fn).reshape(D)
        cnt += 1

# Doing PCA, note that each row is a datapoint
from sklearn.decomposition import PCA
pca = PCA(n_components=K) # K = 100
pca.fit(X.T)

# projection matrix
U = pca.components_.T
```

Tiền xử lý dữ liệu

- Các giải thuật giảm số chiều
  - Giải thuật PCA: được hiểu **ở mức độ ứng dụng** như sau:
    - Ví dụ:
      - Kết quả:

      - Nhận xét:
        - Các hình sau khi giảm số chiều, có:
          - Kích thước lưu trữ nhỏ hơn
          - Số feature nhỏ hơn
          - Vẫn thể hiện khá đầy đủ các đặc trưng của khuôn mặt

Chuẩn bị dữ liệu

Q / A