Python编程与人工智能实践

算法篇：数据降维-MDS

于泓
鲁东大学
信息与电气工程学院
2021.9.24

MDS（multidimensional scaling）多维标度分析

MDS是一种基于**距离度量**的数据降维方法，要求将高维数据 $\mathbf{X} \in R^{d \times m}$ 转换为低维数据 $\mathbf{Z} \in R^{d' \times m}$ 后，样本点间相对位置关系不变。

$$\sum_{\min} \left(\left\| \mathbf{z}_i - z_j \right\| - d_{ij} \right)^2$$

低维样本 $\mathbf{z}_i$ 与 $\mathbf{z}_j$ 之间的欧式距离

高维样本 $\mathbf{x}_i$ 与 $\mathbf{x}_j$ 之间的距离（**度量方法任意**）

(1) $d_{ii} = 0$

(2) $d_{ij} = d_{ji}$

(3) $d_{ik} + d_{jk} \geq d_{ij}$

$$\sum_{\min} \left(\left\| z_i - z_j \right\| - d_{ij} \right)^2$$

上式没有唯一解

$$\left\| z_i' - z_j' \right\| = \left\| (z_i - z_0) - (z_j - z_0) \right\| = \left\| z_i - z_j \right\|$$

例如：

$$X=[x_1, x_2] = \begin{bmatrix} 0,1,0,0,0 \\ 1,0,0,0,0 \end{bmatrix}$$

$$D = \begin{bmatrix} 0 & \sqrt{2} \\ \sqrt{2} & 0 \end{bmatrix}$$

$$z_1 = \left[\frac{\sqrt{2}}{2} \right], z_2 = \left[-\frac{\sqrt{2}}{2} \right]$$

$$z_1 = [0], z_2 = \left[\sqrt{2} \right]$$

$$\sum_{\min} \left(\left\| z_i - z_j \right\| - d_{ij} \right)^2$$

直接求 Z 比较困难，转而求

$$B = Z^T Z \qquad Z \in R^{d' \times m} \qquad B \in R^{m \times m}$$

B是一个实对称矩阵，如果能够求得B，那么：

$$B = U \Lambda U^T = \left(\Lambda^{\frac{1}{2}} U^T \right)^T \left(\Lambda^{\frac{1}{2}} U^T \right) = Z^T Z$$

特征向量　　特征值

$$\begin{aligned} d_{ij}^2 &= \left\| z_i - z_j \right\|^2 = \left\| z_i \right\|^2 + \left\| z_j \right\|^2 - 2 z_i^T z_j \\ &= z_i^T z_i + z_j^T z_j - 2 z_i^T z_j \\ &= b_{ii} + b_{jj} - 2 b_{ij} \end{aligned}$$

可通过调整偏移量使该项为0

$$\sum_{i=1}^{m} d_{ij}^2 = \sum_{i=1}^{m} b_{ii} + \mathrm{m} b_{jj} + \sum_{i=1}^{m} 2 z_i^T z_j = \sum_{i=1}^{m} b_{ii} + \mathrm{m} b_{jj} - 2 \left(\sum_{i=1}^{m} z_i^T \right) z_j$$

$$= \mathrm{track(B)} + \mathrm{m} b_{jj}$$

$$\sum_{j=1}^{m} d_{ij}^2 = \mathrm{m} b_{ii} + \sum_{j=1}^{m} b_{jj} + \sum_{j=1}^{m} 2 z_i^T z_j = m b_{ii} + \mathrm{track(B)}$$

$$\sum_{i=1}^{m} \sum_{j=1}^{m} d_{ij}^2 = 2\mathrm{m} * \mathrm{track(B)}$$

$$\begin{cases} d_{ij}^2 = b_{ii} + b_{jj} - 2b_{ij} \\ \sum_{i=1}^{m} d_{ij}^2 = \text{track(B)} + \text{m}b_{jj} \\ \sum_{j=1}^{m} d_{ij} = mb_{ii} + \text{track(B)} \\ \sum_{i=1}^{m}\sum_{j=1}^{m} d_{ij} = 2\text{m*track(B)} \end{cases}$$

解方程可得：

$$b_{ij} = -\frac{1}{2}\left(d_{ij}^2 - \frac{1}{\text{m}}\sum_{i=1}^{m} d_{ij}^2 - \frac{1}{\text{m}}\sum_{j=1}^{m} d_{ij}^2 + \frac{1}{\text{m}^2}\sum_{i=1}^{m}\sum_{j=1}^{m} d_{ij}^2 \right)$$

MDS 一般步骤：

（1） 利用给定数据计算距离矩阵（不相似矩阵）D

（2）计算降维后矢量z的互相关矩阵B

（3）对B进行特征值分解，选取较大的若干特征值与特征矢量获取Z

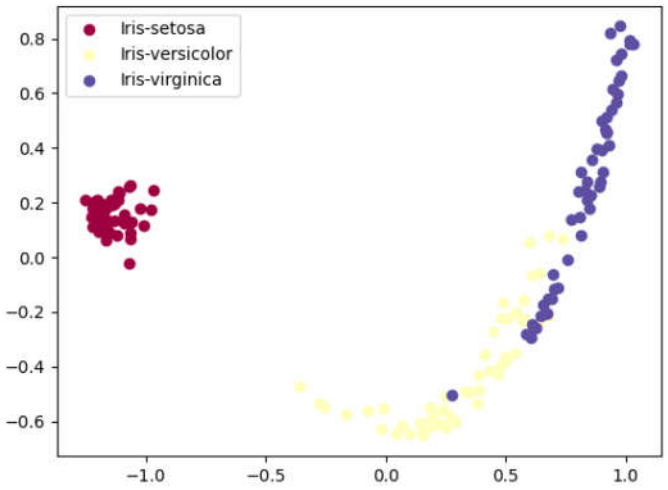
代码实现：

```python
import numpy as np
import matplotlib.pyplot as plt

# x 维度 [N,D]
def cal_pairwise_dist(x):

    N,D = np.shape(x)

    dist = np.zeros([N,N])

    for i in range(N):
        for j in range(N):
            dist[i,j] = np.dot((x[i]-x[j]),(x[i]-x[j]).T)
            # dist[i,j] = np.sqrt(np.dot((x[i]-x[j]),(x[i]-x[j]).T))
            # dist[i,j] = np.sum(np.abs(x[i]-x[j]))

    #返回任意两个点之间距离
    return dist
```

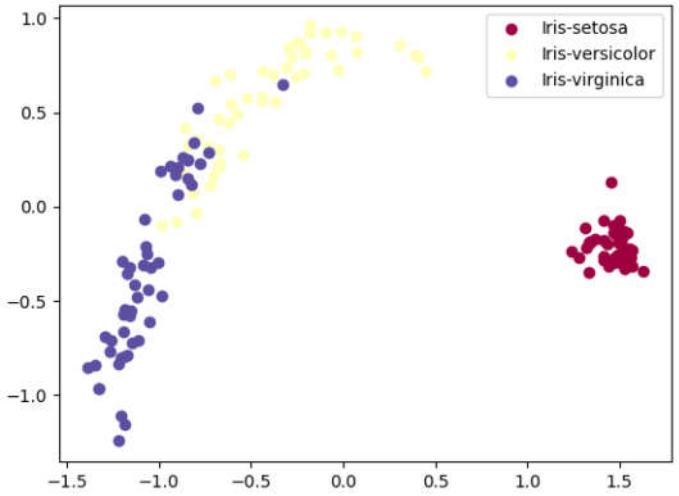
```python
# dist N*N 距离矩阵样本点两两之间的距离
# n_dims 降维
# 返回 降维后的数据
def my_mds(dist, n_dims):

    n,n = np.shape(dist)

    dist[dist < 0 ] = 0
    dist = dist**2
    T1 = np.ones((n,n))*np.sum(dist)/n**2
    T2 = np.sum(dist, axis = 1, keepdims=True)/n
    T3 = np.sum(dist, axis = 0, keepdims=True)/n

    B = -(T1 - T2 - T3 + dist)/2

    eig_val, eig_vector = np.linalg.eig(B)
    index_ = np.argsort(-eig_val)[:n_dims]
    picked_eig_val = eig_val[index_].real
    picked_eig_vector = eig_vector[:, index_]

    return picked_eig_vector*picked_eig_val**(0.5)
```

```python
def draw_pic(datas,labs):
    plt.cla()
    unque_labs = np.unique(labs)
    colors = [plt.cm.Spectral(each)
      for each in np.linspace(0, 1,len(unque_labs))]
    p=[]
    legends = []
    for i in range(len(unque_labs)):
        index = np.where(labs==unque_labs[i])
        pi = plt.scatter(datas[index, 0], datas[index, 1], c =[colors[i]] )
        p.append(pi)
        legends.append(unque_labs[i])

    plt.legend(p, legends)
    plt.show()
```

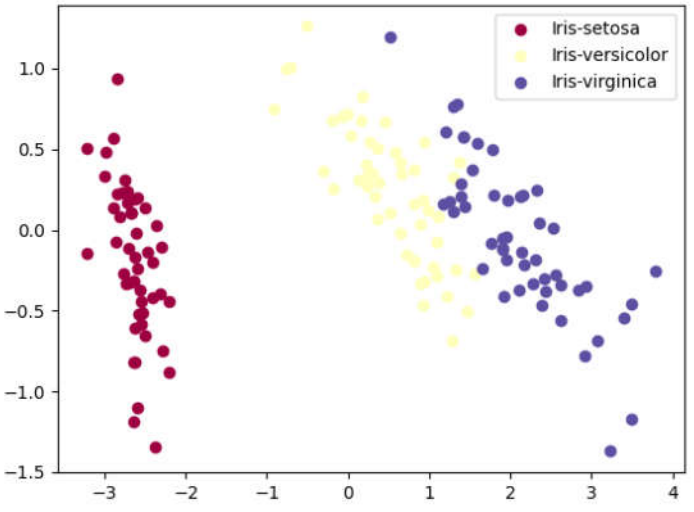
```python
if __name__ == "__main__":

    # 加载数据
    data = np.loadtxt("iris.data",dtype="str",delimiter=',')
    feas = data[:,:-1]
    feas = np.float32(feas)
    labs = data[:,-1]

    # 计算距离
    dist = cal_pairwise_dist(feas)

    # 进行降维
    data_2d = my_mds(dist, 2)

    #绘图
    draw_pic(data_2d,labs)
```



二范数



1范数



二范数平方