Python程序设计

第十讲 机器学习应用

K-means聚类算法及其应用

张 华

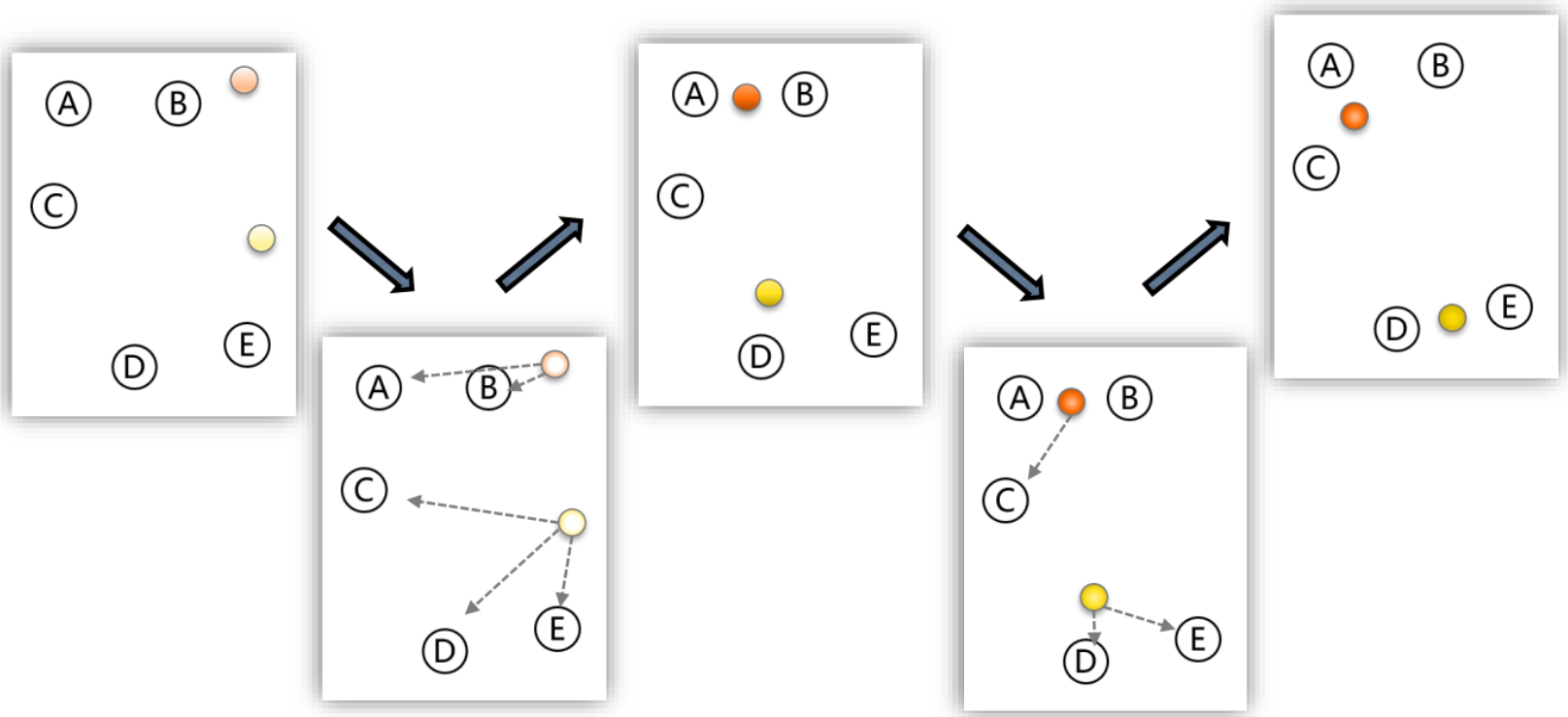
WHU

K-means聚类算法

- k-means算法以k为参数，把n个对象分成k个簇，使簇内具有较高的相似度，而簇间的相似度较低。
- 其处理过程如下：
  - （1）随机选择k个点作为初始的聚类中心；
  - （2）对于剩下的点，根据其与聚类中心的距离，将其归入最近的簇；
  - （3）对每个簇，计算所有点的均值作为新的聚类中心；
  - （4）重复2、3直到聚类中心不再发生改变。

K-means聚类算法

k-means算法示例



应用案例

- 数据介绍
  - 现有1999年全国31个省份城镇居民家庭平均每人全年消费性支出的八个主要变量数据，这八个变量分别是：食品、衣着、 家庭设备用品及服务、 医疗保健、 交通和通讯、娱乐教育文化服务、 居住以及杂项商品和服务。
  - 利用已有数据，对31个省份进行聚类。
- 实验目的
  - 通过聚类，了解1999年各个省份的消费水平在国内的情况。
- 技术路线
  - sklearn.cluster.Kmeans

应用案例

数据情况

城市	食品	衣着	家庭设备用品及服务	医疗保健	交通和通讯	娱乐教育文化服务	居住	杂项商品和服务
北京	2959.19	730.79	749.41	513.34	467.87	1141.82	478.42	457.64
天津	2459.77	495.47	697.33	302.87	284.19	735.97	570.84	305.08
河北	1495.63	515.9	362.37	285.32	272.95	540.58	364.91	188.63
山西	1406.33	477.77	290.15	208.57	201.5	414.72	281.84	212.1
内蒙古	1303.97	524.29	254.83	192.17	249.81	463.09	287.87	192.96
辽宁	1730.84	553.9	246.91	279.81	239.18	445.2	330.24	163.86
吉林	1561.86	492.42	200.49	218.36	220.69	459.62	360.48	147.76
黑龙江	1410.11	510.71	211.88	277.11	224.65	376.82	317.61	152.85
上海	3712.31	550.74	893.37	346.93	527	1034.98	720.33	462.03
江苏	2207.58	449.37	572.4	211.92	302.09	585.23	429.77	252.54
浙江	2629.16	557.32	689.73	435.69	514.66	795.87	575.76	323.36
安徽	1844.78	430.29	271.28	126.33	250.56	513.18	314	151.39
福建	2709.46	428.11	334.12	160.77	405.14	461.67	535.13	232.29
江西	1563.78	303.65	233.81	107.9	209.7	393.99	509.39	160.12
山东	1675.75	613.32	550.71	219.79	272.59	599.43	371.62	211.84
河南	1427.65	431.79	288.55	208.14	217	337.76	421.31	165.32
湖南	1942.23	512.27	401.39	206.06	321.29	697.22	492.6	226.45
湖北	1783.43	511.88	282.84	201.01	237.6	617.74	523.52	182.52
广东	3055.17	353.23	564.56	356.27	811.88	873.06	1082.82	420.81
广西	2033.87	300.82	338.65	157.78	329.06	621.74	587.02	218.27
海南	2057.86	186.44	202.72	171.79	329.65	477.17	312.93	279.19
重庆	2303.29	589.99	516.21	236.55	403.92	730.05	438.41	225.8
四川	1974.28	507.76	344.79	203.21	240.24	575.1	430.36	223.46
贵州	1673.82	437.75	461.61	153.32	254.66	445.59	346.11	191.48
云南	2194.25	537.01	369.07	249.54	290.84	561.91	407.7	330.95
西藏	2646.61	839.7	204.44	209.11	379.3	371.04	269.59	389.33
陕西	1472.95	390.89	447.95	259.51	230.61	490.9	469.1	191.34
甘肃	1525.57	472.98	328.9	219.86	206.65	449.69	249.66	228.19
青海	1654.69	437.77	258.78	303	244.93	479.53	288.56	236.51
宁夏	1375.46	480.89	273.84	317.32	251.08	424.75	228.73	195.93
新疆	1608.82	536.05	432.46	235.82	250.28	541.3	344.85	214.4

应用案例

实现过程

- 创建cityClustering.py，导入sklearn相关包

```
import numpy as np
from sklearn.cluster import KMeans
```

- 关于一些相关包的介绍：
  - NumPy是Python语言的一个扩充程序库。 支持高级大量的维度数组与矩阵运算，此外也针对数组运算提供大量的数学函数库。
  - 使用sklearn.cluster.KMeans可以调用K-means算法进行聚类。
    - 默认使用欧式距离计算两项数据的相似度。

应用案例

实现过程

加载数据

city.txt - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

北京,2959.19,730.79,749.41,513.34,467.87,1141.82,478.42,457.64
天津,2459.77,495.47,697.33,302.87,284.19,735.97,570.84,305.08
河北,1495.63,515.90,362.37,285.32,272.95,540.58,364.91,188.63
山西,1406.33,477.77,290.15,208.57,201.50,414.72,281.84,212.10
内蒙古,1303.97,524.29,254.83,192.17,249.81,463.09,287.87,192.96
辽宁,1730.84,553.90,246.91,279.81,239.18,445.20,330.24,163.86
吉林,1561.86,492.42,200.49,218.36,220.69,459.62,360.48,147.76
黑龙江,1410.11,510.71,211.88,277.11,224.65,376.82,317.61,152.85
上海,3712.31,550.74,893.37,346.93,527.00,1034.98,720.33,462.03

```
def loadData(dataFile):
    f = open(dataFile,'r+')
    lines = f.readlines()
    cityData = []
    cityName = []
    for line in lines:
        items = line.strip().split(",")
        cityName.append(items[0])
        cityData.append([float(items[i]) for i in range(1, len(items))])
    return cityData, cityName

def cityClustering(levelCount=3):
    cityData, cityName = loadData('city.txt')
    …
```

应用案例

实现过程

- 创建KMeans实例，调用fit_predict计算，获得分类标签。

```
def cityClustering(levelCount=3):
    cityData, cityName = loadData('city.txt')
    km = KMeans(n_clusters=levelCount)
    label = km.fit_predict(cityData)
    expenses = np.sum(km.cluster_centers_, axis=1)
    …
```

调用KMeans方法所需参数：

- n_clusters：用于指定聚类中心的个数
- init：初始聚类中心的初始化方法
- max_iter：最大的迭代次数

一般调用时只用给出n_clusters即可，init默认是k-means++，max_iter默认是300

其它说明：

- fit_predict()：计算簇中心以及为簇分配序号
- cityData：加载的数据
- label：聚类后各数据所属的标签（0，1，2，……）
- axis=1: 按行求和

应用案例

实现过程

- 将城市按label分成设定的簇，输出每个簇的城市和平均消费额。

```
def cityClustering(levelCount=3):
    ……
    cityCluster = [[] for i in range(levelCount)]
    for i in range(len(cityName)):
        cityCluster[label[i]].append(cityName[i])

    for i in range(len(cityCluster)):
        print("{}类省份平均消费额：{:.2f}元".format(i+1, expenses[i]))
        print(cityCluster[i])

if __name__ == '__main__':
    cityClustering(4)
```

应用案例

分析结果

3类

```
1类省份平均消费额：5113.54元
['天津', '江苏', '浙江', '福建', '湖南', '广西', '海南', '重庆', '四川', '云南', '西藏']
2类省份平均消费额：3827.87元
['河北', '山西', '内蒙古', '辽宁', '吉林', '黑龙江', '安徽', '江西', '山东', '河南', '湖北', '贵州', '陕西', '甘肃', '青海', '宁夏', '新疆']
3类省份平均消费额：7754.66元
['北京', '上海', '广东']
```

应用案例

分析结果

4类

```
1类省份平均消费额：4512.27元
['江苏', '安徽', '湖南', '湖北', '广西', '海南', '四川', '云南']
2类省份平均消费额：3788.76元
['河北', '山西', '内蒙古', '辽宁', '吉林', '黑龙江', '江西', '山东', '河南', '贵州', '陕西', '甘肃', '青海', '宁夏', '新疆']
3类省份平均消费额：7754.66元
['北京', '上海', '广东']
4类省份平均消费额：5678.62元
['天津', '浙江', '福建', '重庆', '西藏']
```

应用案例

分析结果

5类

```
1类省份平均消费额：4512.27元
['江苏', '安徽', '湖南', '湖北', '广西', '海南', '四川', '云南']
2类省份平均消费额：7517.80元
['广东']
3类省份平均消费额：3788.76元
['河北', '山西', '内蒙古', '辽宁', '吉林', '黑龙江', '江西', '山东', '河南', '贵州', '陕西', '甘肃', '青海', '宁夏', '新疆']
4类省份平均消费额：5678.62元
['天津', '浙江', '福建', '重庆', '西藏']
5类省份平均消费额：7873.09元
['北京', '上海']
```

K-means算法的不足

- k-means算法分析
  - 通过设置不同的k值，能够得到不同的聚类结果。
  - k值的不确定也是Kmeans算法的一个缺点。
  - 往往为了达到好的实验结果，需要进行多次尝试才能够选取最优的k值。
  - 而像层次聚类的算法，就无需指定k值，只要给定限制条件，就能自动地得到类别数k。