聚类分析概述

- 聚类分析又称群分析，它是研究（样品或指标）分类问题的一种多元统计方法，所谓类，通俗地说，就是指相似元素的集合。
- 聚类分析内容非常丰富，有系统聚类法、有序样品聚类法、动态聚类法、模糊聚类法、图论聚类法、聚类预报法等。

聚类分析的分类：

- **样品聚类**（Q 型聚类）：对事件(Cases)进行聚类，或是说对观测量进行聚类。
- **变量聚类**（R 型聚类）：进行变量聚类，找出彼此独立且有代表性的自变量，而又不丢失大部分信息

距离的度量：

- 为了将样品（或指标）进行分类，就需要研究样品之间关系。
- 目前用得最多的方法有两个：
  - 一种方法：相似系数；
  - 另一种方法：距离 ；

距离的定义方式：

- 绝对值距离：

$$d_{ij}=\sum_{k=1}^{p}|x_{ik}-x_{jk}| \tag{1}$$

- 欧氏(Euclidean)距离：

$$d_{ij}=\left[\sum_{k=1}^{p}(x_{ik}-x_{jk})^2\right]^{\frac{1}{2}} \tag{2}$$

- 切比雪夫(Chebychev)距离:

$$d_{ij}=\max_{1\leqslant k\leqslant p}|x_{ik}-x_{jk}| \tag{3}$$

- 明氏(Minkowski)距离:

$$d_{ij}=\left[\sum_{k=1}^{p}|x_{ik}-x_{jk}|^q\right]^{\frac{1}{q}} \tag{4}$$

相似系数

- 夹角余弦：

$$\cos\theta_{st}=\frac{\sum_{i=1}^{n}x_{is}x_{it}}{\sqrt{\sum_{i=1}^{n}x_{is}^2\cdot\sum_{i=1}^{n}x_{it}^2}} \tag{5}$$

- Pearson相关系数:

$$\cos\theta_{st}=\frac{\sum_{i=1}^{n}(x_{is}-\bar{x_s})(x_{it}-\bar{x_t})}{\sqrt{\sum_{i=1}^{n}(x_{is}-\bar{x_s})^2\cdot\sum_{i=1}^{n}(x_{it}-\bar{x_s})^2}} \tag{6}$$

- 指数相似系数：

$$c_{ij}=\frac{1}{p}\sum_{k=1}^{p}\exp\left[-\frac{3}{4}\frac{(x_{ik}-x_{jk})^2}{s_k^2}\right] \tag{7}$$

系统聚类的基本思想

** 物以类聚 **

- 相近的聚为一类(以距离表示，样品聚类)
- 相似的聚为一类(以相似系数表示，变量聚类)

系统聚类的基本步骤

1. 构造n个类，每个类包含且只包含一个样品。
2. 计算n个样品两两间的距离，构成距离矩阵，记作D0。
3. 合并距离最近的两类为一新类。
4. 计算新类与当前各类的距离。若类的个数等于1，转到步骤(5)，否则回到步骤(3)。
5. 画聚类图。
6. 决定类的个数，及各类包含的样品数，并对类作出解释。

系统聚类(Hierarchical clustering)

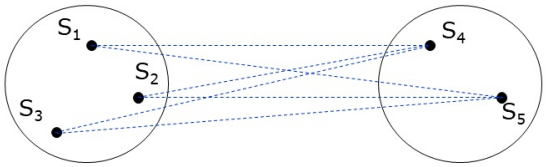
- 最短距离法(single linkage)
- 最长距离法(complete linkage)
- 中间距离法(median method)
- 可变距离法(flexible median)
- 重心法(centroid)
- 类平均法(average)

- 可变类平均法(flexible average)
- Ward最小方差法(Ward’s minimum variance)

类间距离

最长距离(complete linkage)

重心间距离(centroid)



平均距离(average)

$$D_{12}^2 = \frac{1}{6}(d_{14}^2 + d_{15}^2 + d_{24}^2 + d_{25}^2 + d_{34}^2 + d_{35}^2)$$

最短距离

实验

为了反映中国各个地区的生活水平差异，我们收集整理了2016年中国部分省市的国民经济数据，具体包括：

- 国内生产国总值
- 年末人口总数
- 城乡居民年末储蓄余额
- 在岗职位平均工资

- 住宅商品房平均价格
- 社会商品零售总额
- 进出口货物总额
- 普通高等院校在校人数
- 医院个数

现希望通过聚类分析的方法把相似的省份找出来，即把这些省份归为若干类别，从而更好的了解中国各省市地区生活水平的差异。

```
import pandas as pd
import matplotlib.pyplot as plt

%matplotlib inline
```

```
df = pd.read_excel('./data/主要城市聚类.xlsx', index_col=0)
df.head()
```

	国内生产总值	年末人口总数	城乡居民年末储蓄余额	在岗职位平均工资	住宅商品房平均价格	社会商品零售总额	进出物
北京	23014.59	1345.20	23913.97	113073	22300.0	10338.0	31941
天津	16538.19	1026.90	8743.79	81486	9931.0	5257.3	11428
石家庄	5440.60	1028.84	4868.93	54441	7798.0	2693.0	12160
太原	2735.34	367.39	3432.12	60516	7303.0	1540.8	10677
呼和浩特	3090.52	238.58	1683.96	53698	4946.0	1353.5	2072.7

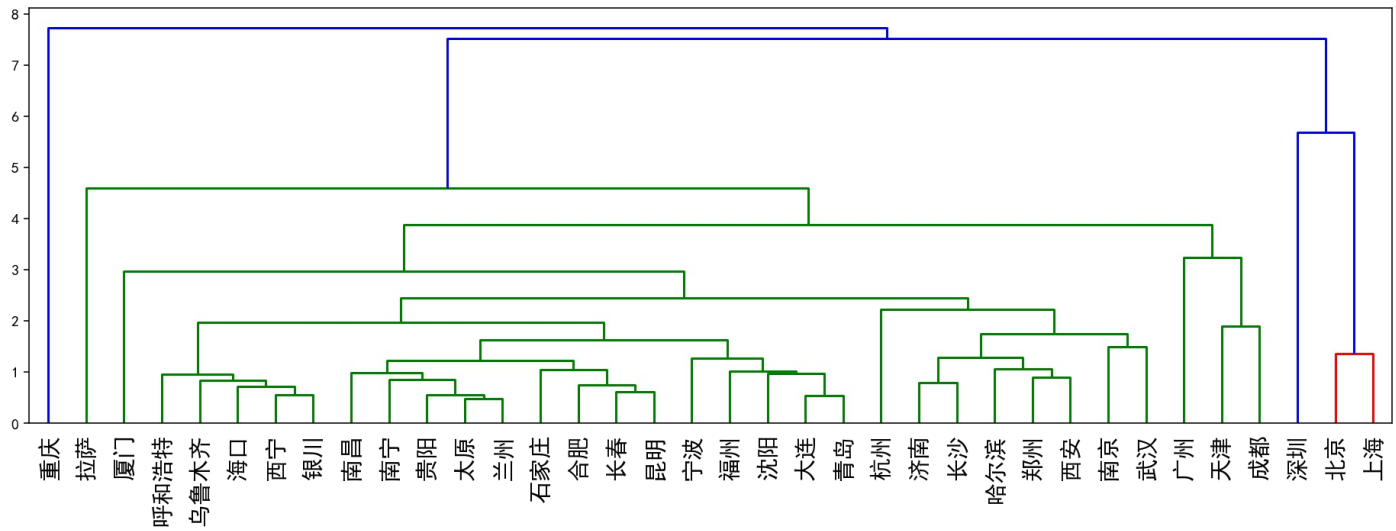
```
df.isnull().sum()
```

```
国内生产国总值          0
年末人口总数           0
城乡居民年末储蓄余额      0
在岗职位平均工资         0
住宅商品房平均价格        1
社会商品零售总额         0
进出口货物总额          0
普通高等院校在校人数      0
医院个数             0
dtype: int64
```

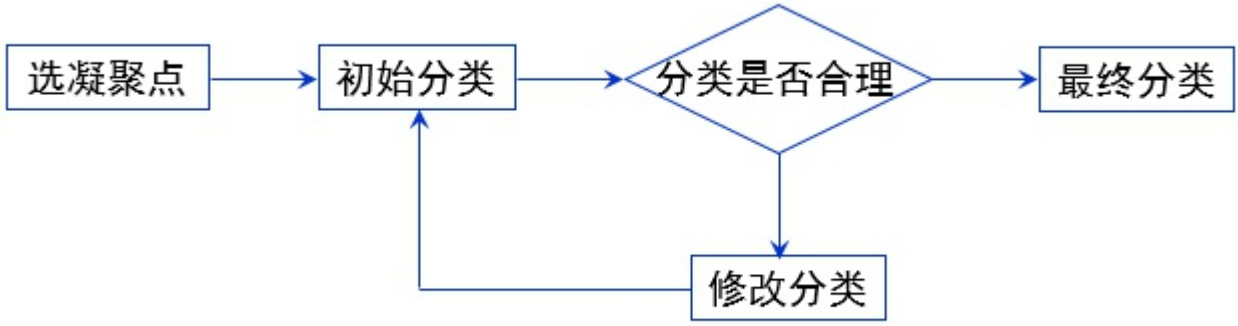
```
df.fillna(value=df.mean(), inplace=True);
```

```
from sklearn.preprocessing import scale
data = scale(df)

from scipy.cluster.hierarchy import linkage, dendrogram

##  average平均距离
Z = linkage(data, 'average')
```

```
plt.figure(figsize=(16,5))
dn = dendrogram(Z,labels=df.index, leaf_font_size=15, leaf_rotation=90 )
```



快速聚类

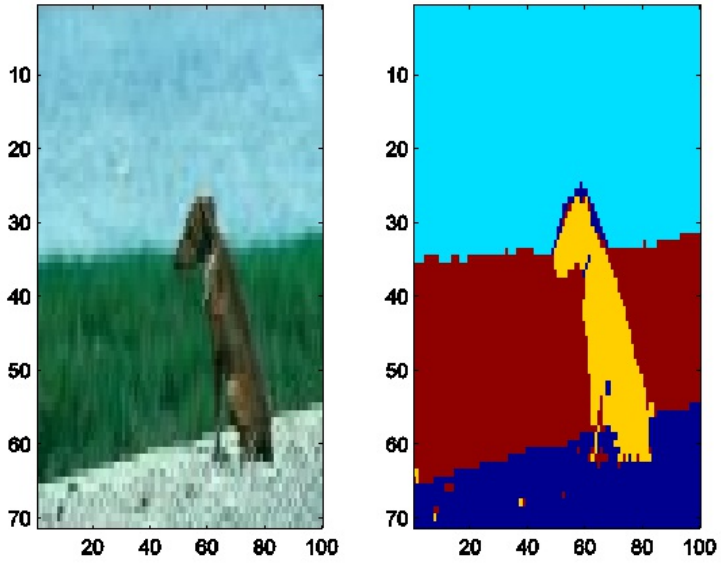


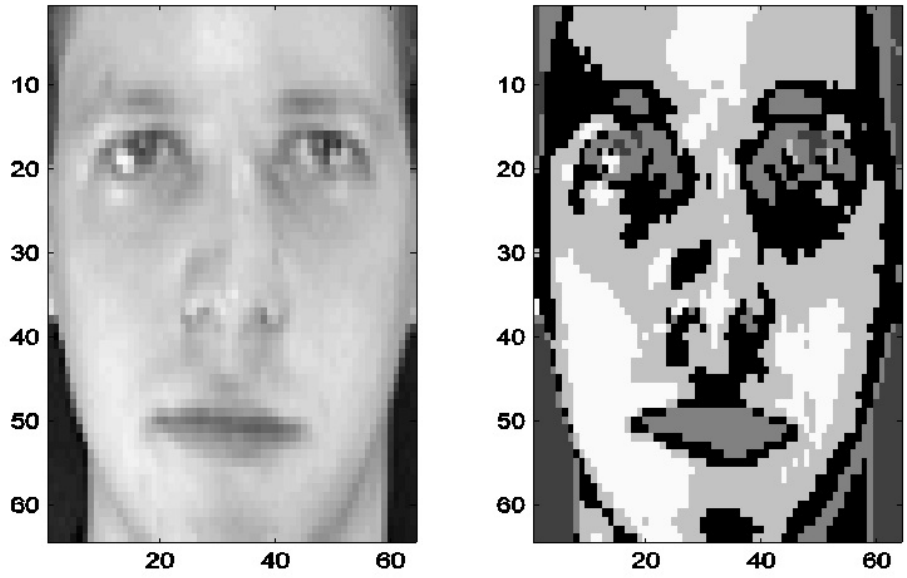
聚类分析要注意的问题

- 用什么指标(变量)表达要分析的样品？
- 用什么统计量(距离、相似系数)描述样本间的相似程度？
- 用什么方法(类间距离等)进行聚类？
- 分成几类比较合适？

在图像分割上的简单应用

1. 此图为100 x 100像素的JPG图片，每个像素可以表示为三维向量（分别对应JPEG图像中的红色、绿色和蓝色通道）；
2. 将图片分割为合适的背景区域（三个）和前景区域（小狗）；
3. 使用K-means算法对图像进行分割。



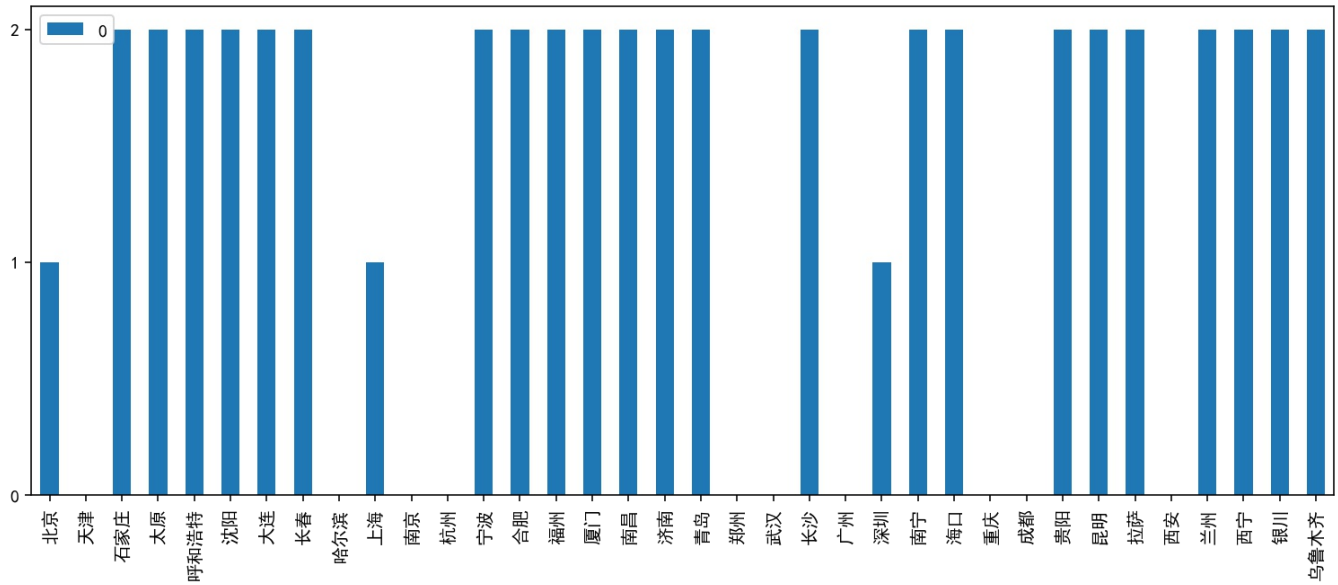


实验

```
from sklearn.cluster import KMeans
kmeans = KMeans(3)
kmeans.fit(data)
```

```
KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
    n_clusters=3, n_init=10, n_jobs=1, precompute_distances='auto',
    random_state=None, tol=0.0001, verbose=0)
```

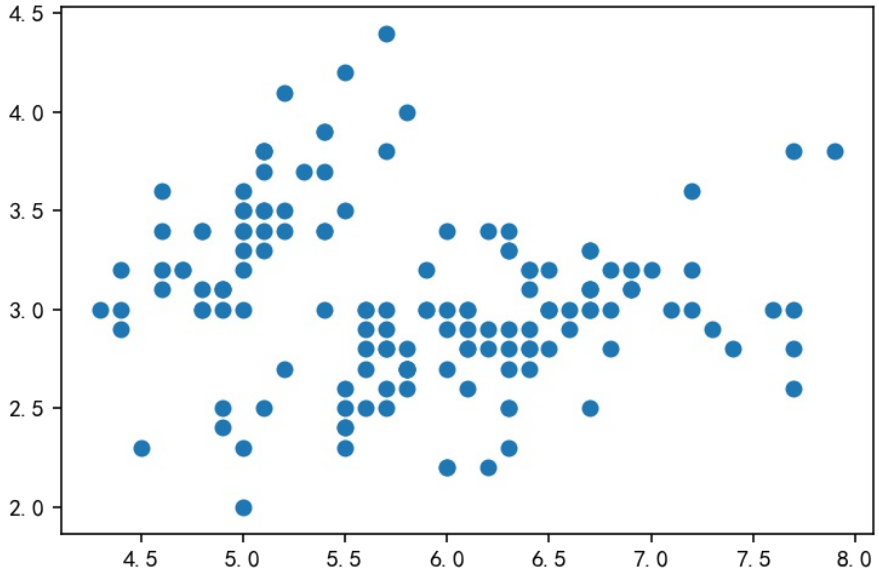
```
df_label = pd.DataFrame(kmeans.labels_, index=df.index)

df_label.plot.bar(figsize=(13,5))
plt.yticks([0,1,2]);
```



可视化

```
from sklearn.datasets import load_iris
iris = load_iris()
X = iris.data[:,0:2]
```

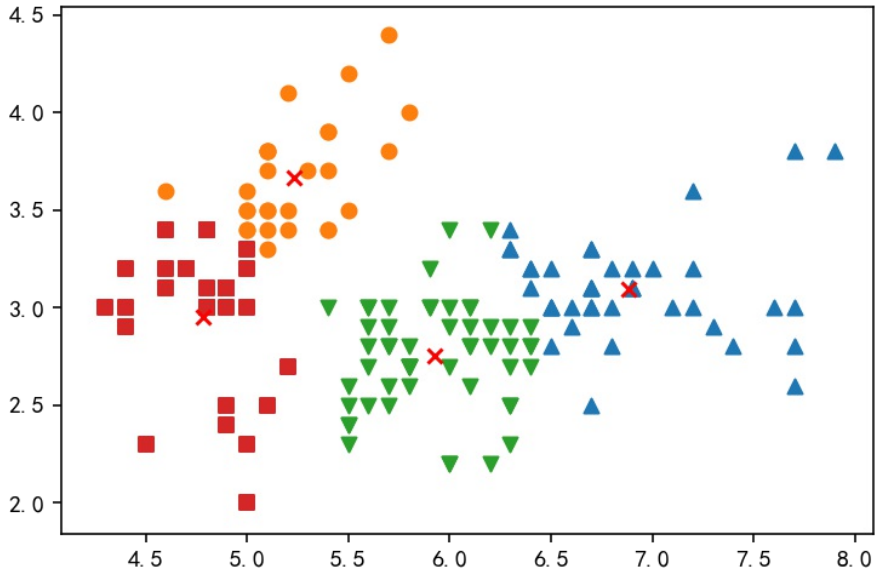
```
plt.scatter(X[:, 0], X[:, 1]);
```




```
from sklearn.cluster import KMeans
kmeans = KMeans(4)
kmeans.fit(X)
```

```
KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
    n_clusters=4, n_init=10, n_jobs=1, precompute_distances='auto',
    random_state=None, tol=0.0001, verbose=0)
```

```
y = kmeans.labels_
markers = ['^', 'o', 'v', 's', 'd']
for i,marker in zip(np.unique(y), markers):
    plt.scatter(X[y==i, 0], X[y==i, 1], marker=marker)
plt.scatter(kmeans.cluster_centers_[:,0], kmeans.cluster_centers_[:,1], m
```


4.5
4.0
3.5
3.0
2.5
2.0
4.5
5.0
5.5
6.0
6.5
7.0
7.5
8.0