聚类算法

黑马程序员
www.itheima.com

传智教育旗下
高端IT教育品牌

目录

Contents

学习目标

Learning Objectives

1. 知道什么是聚类
2. 了解聚类算法的应用场景
3. 知道聚类算法的分类

聚类算法 - 概念

- 什么是聚类算法?
  - 根据样本之间的相似性，将样本划分到不同的类别中；不同的相似度计算方法，会得到不同的聚类结果，常用的相似度计算方法有欧式距离法。
  - 聚类算法的目的是在没有先验知识的情况下，自动发现数据集中的内在结构和模式。
  - 无监督学习算法
- 使用不同的聚类准则，产生的聚类结果不同

繁衍方式 (胎生、卵生)

呼吸方式 (肺、腮)

生活环境(陆地、两栖、水中)

繁衍后台方式及呼吸方式

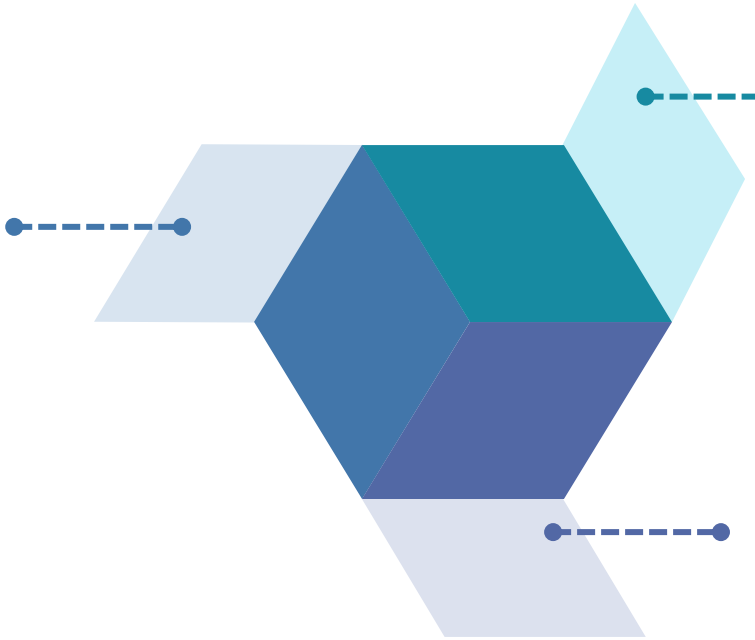
聚类算法在现实生活中的应用

用户画像，广告推荐

Data Segmentation

搜索引擎的流量推荐

恶意流量识别



基于位置信息的商业推送

新闻聚类，筛选排序

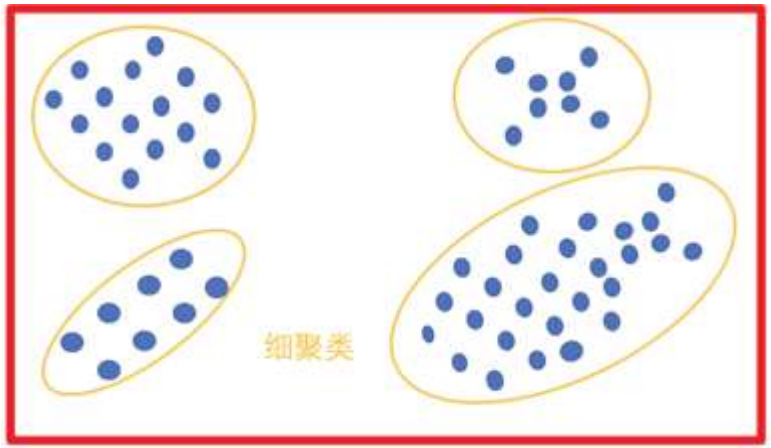
图像分割，降维，识别

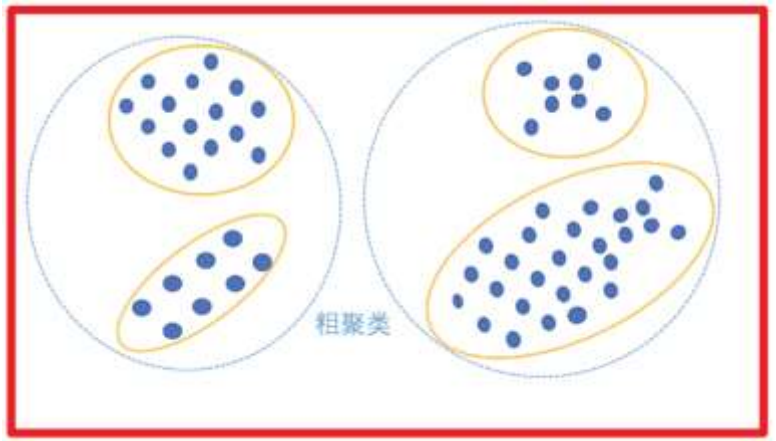
离群点检测，信用卡异常消费

发掘相同功能的基因片段

聚类算法分类

1.根据聚类颗粒度分类





2.根据实现方法分类

- K-means：按照质心分类，主要介绍K-means，通用、普遍
- 层次聚类：对数据进行逐层划分，直到达到聚类的类别个数
- DBSCAN聚类（Density-Based Spatial Clustering of Applications with Noise）是一种基于密度的聚类算法
- 谱聚类是一种基于图论的聚类算法

总结

1 聚类概念

无监督学习算法，主要用于将相似的样本自动归到一个类别中；

计算样本和样本之间的相似性，一般使用欧式距离

2 聚类分类

- 颗粒度：粗聚类、细聚类。
- 实现方法： K-means聚类、层次聚类、 DBSCAN聚类、谱聚类

练习

1、下列关于聚类算法的描述错误的是？

A）聚类算法是一种无监督的机器学习算法

B）聚类算法通过计算样本之间的相似度来确定它是属于哪一个聚集类别

C）在聚类算法中样本之间的相似度只能通过欧式距离来衡量

D）不同的聚类准则产生的聚类效果也不同

答案解析：衡量样本间相似度的方法不止欧式距离一种，它只不过是常用的一种。

答案：C

目录

Contents

学习目标

Learning Objectives

1. 了解Kmeans算法的API
2. 动手实践Kmeans算法

聚类算法API

- sklearn.cluster.KMeans(n_clusters=8)
  - 参数: n_clusters:开始的聚类中心数量

    整型，缺省值=8，生成的聚类数，即产生的质心（centroids）数。
  - 方法

    estimator.fit(x)

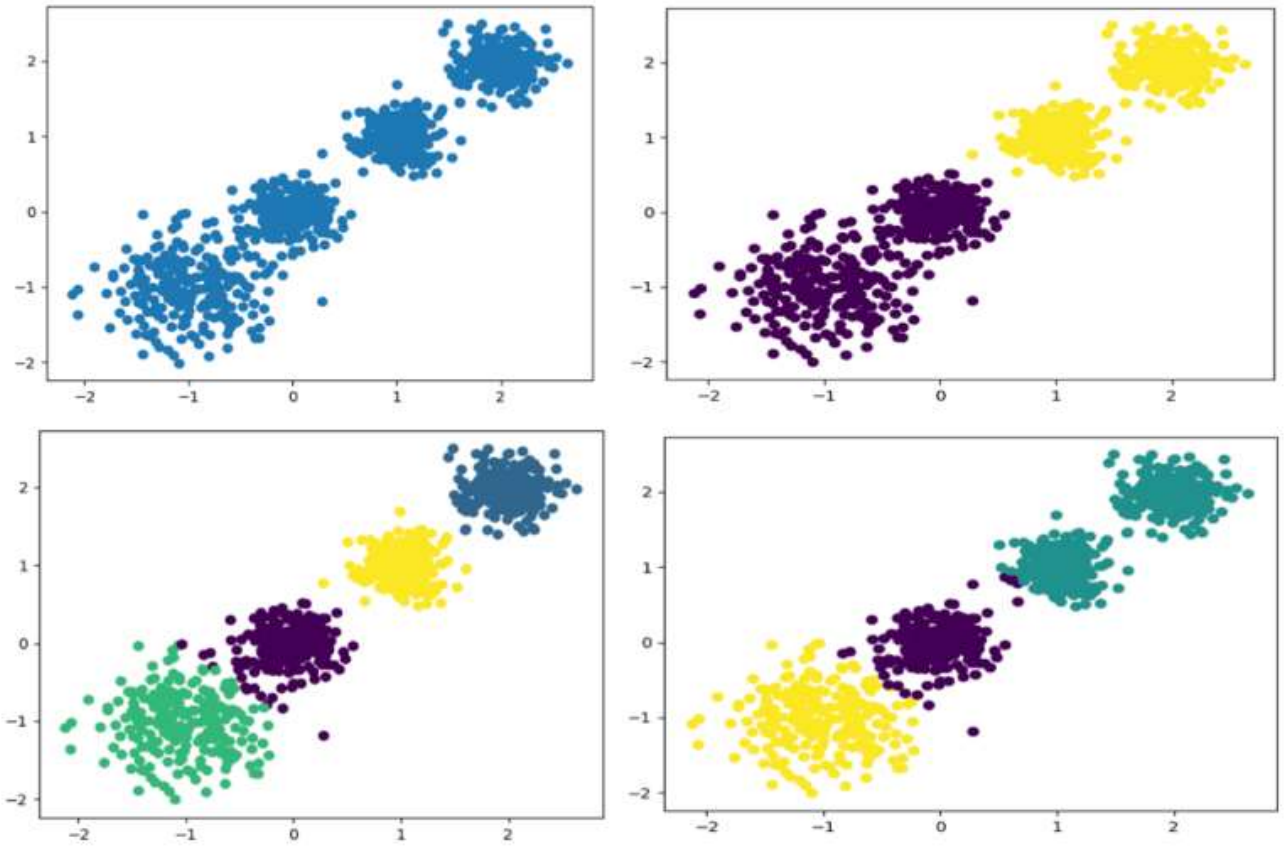
    estimator.predict(x)

    estimator.fit_predict(x)

    计算聚类中心并预测每个样本属于哪个类别,相当于先调用fit(x),然后再调用predict(x)

案例 使用KMeans模型数据探索聚类

随机创建不同二维数据集作为训练集，并结合k-means算法将其聚类，尝试分别聚类不同数量的簇，并观察聚类效果：



案例 使用KMeans模型数据探索聚类

1.导入依赖包

sklearn.cluster.KMeans
sklearn.datasets.make_blobs

2.构建数据集

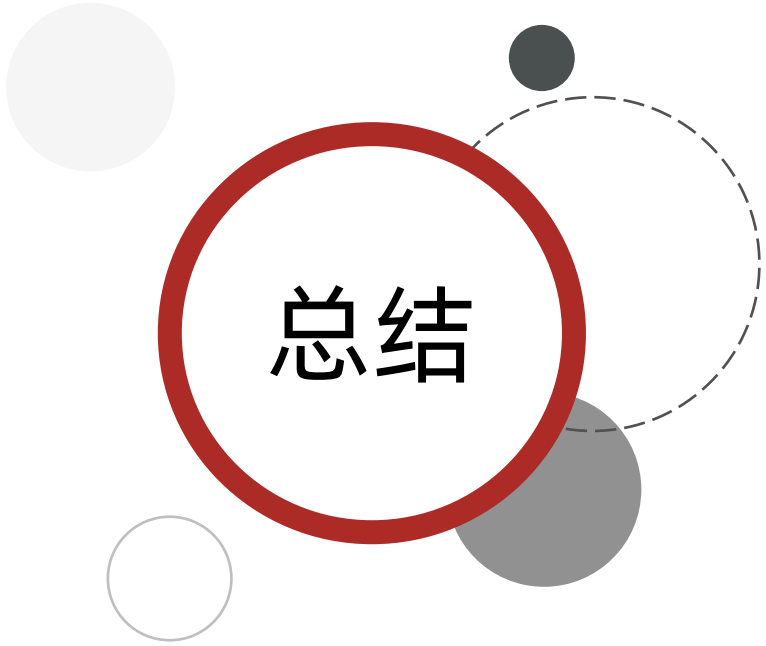
3.模型训练并预测（实例化Kmeans）

4.展示聚类效果

5.评估聚类效果好坏

案例 使用KMeans模型数据探索聚类

```
# 1.导入工具包
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt
from sklearn.datasets import make_blobs
from sklearn.metrics import calinski_harabasz_score  # calinski_harabaz_score 废弃

# 2.构建数据集1000个样本,每个样本2个特征4个质心簇数据标准差[0.4, 0.2, 0.2, 0.2]
x, y = make_blobs(n_samples=1000, n_features=2, centers=[[-1,-1], [0,0], [1,1], [2,2]],cluster_std = [0.4, 0.2, 0.2, 0.2], random_state=22)
plt.figure()
plt.scatter(x[:, 0], x[:, 1], marker='o')
plt.show()

# 3.使用k-means进行聚类
y_pred = KMeans(n_clusters=3, random_state=22).fit_predict(x)

# 4.展示聚类效果
plt.scatter(x[:, 0], x[:, 1], c=y_pred)
plt.show()

# 5.模型评估, 使用CH方法评估
print(calinski_harabasz_score(x, y_pred))
```



1 聚类算法API

- sklearn.cluster.KMeans(n_clusters=8)
- 参数：n_clusters:开 始的聚类中心数量
- 方法：estimator.fit_predict(x)

  计算聚类中心并预测每个样本属于哪个类别,相当于先调用fit(x),

  然后再调用predict(x)
- calinski_harabasz_score(x, y_pred) 用来评估聚类效果，数值越大越好

练习

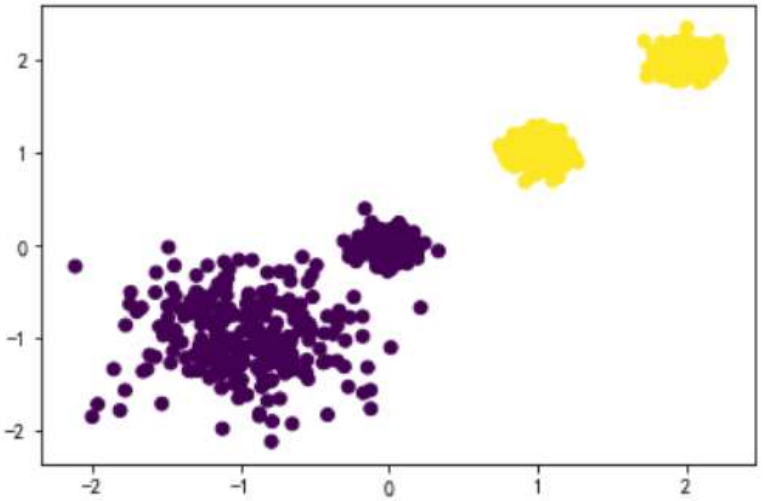
1、下列关于聚类算法API的描述正确的是？（**多选**）

A）它是通过sklearn.cluster.Kmeans来实现的

B）可以通过n_clusters参数指定样本最终被归为多少个聚类

C）右图中的样本一共被分为4个类

D）右图中的样本一共被分为2个类



答案解析：c错误，聚类成2个类别。

答案：ABD

目录

Contents

学习目标

Learning Objectives

1. 能掌握K-means聚类的实现步骤

KMeans算法实现流程

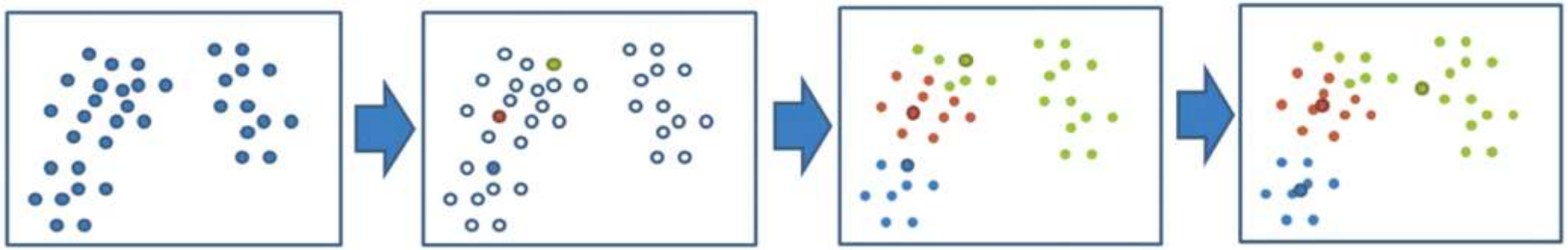
1、事先确定常数K ， 常数K意味着最终的聚类类别数

2、随机选择 K 个样本点作为初始聚类中心

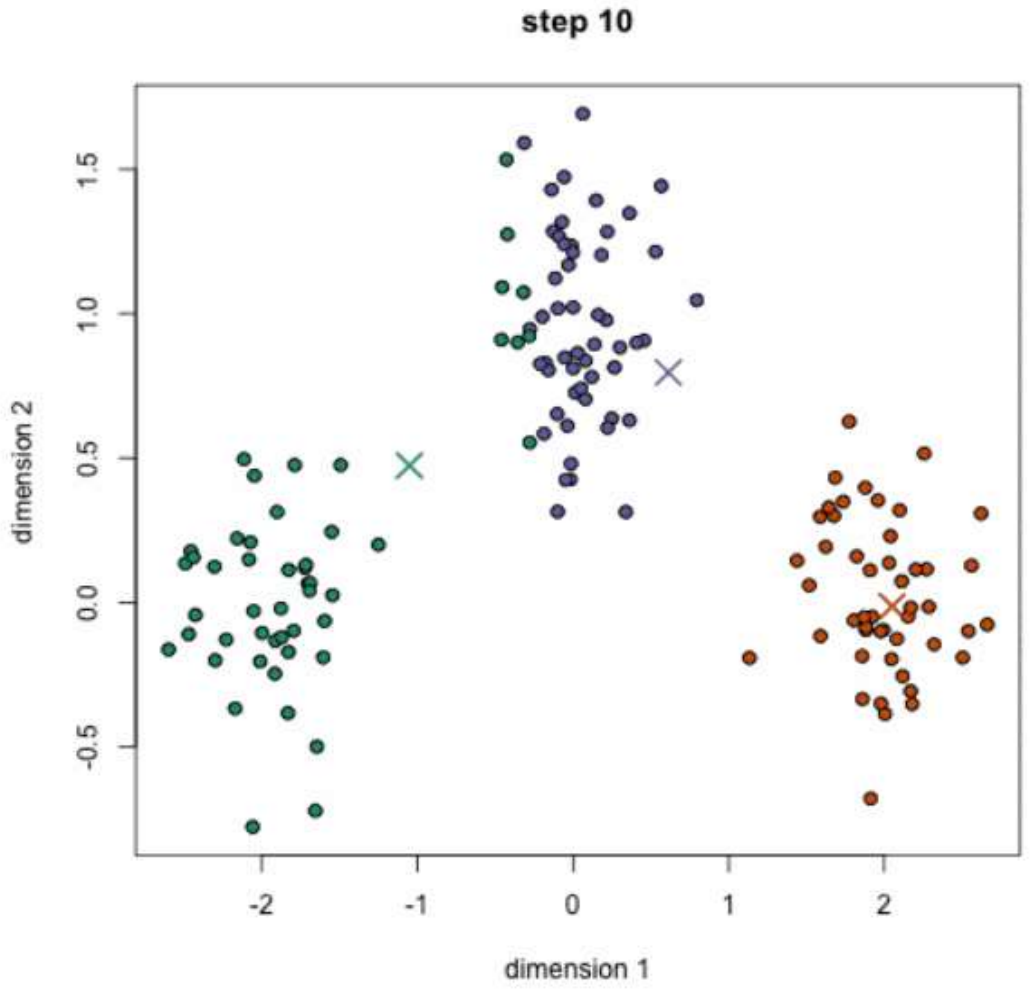
3、计算每个样本到 K 个中心的距离，选择最近的聚类中心点作为标记类别

4、根据每个类别中的样本点，重新计算出新的聚类中心点（平均值），如果计算得出的新中心点与原中心点一样则停止聚类，否则重新进行第 2 步过程，直到聚类中心不再变化

KMeans算法实现流程



KMeans算法实现流程



KMeans算法实现流程 - 举例

已知数据如下表所示，按照KMeans算法实现流程进行聚类：

	X值	Y值
P1	7	7
P2	2	3
P3	6	8
P4	1	4
P5	1	2
P6	3	1
P7	8	8

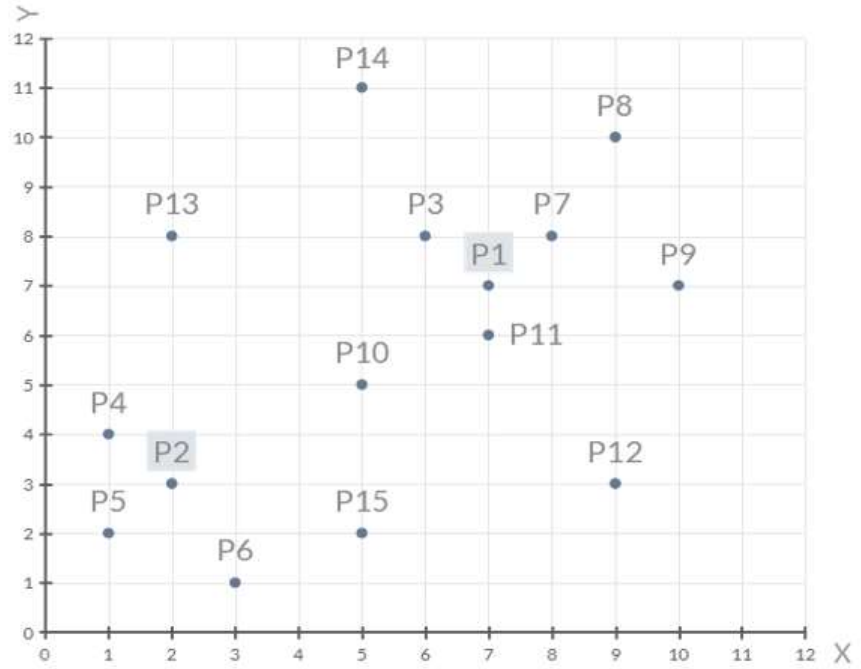
	X值	Y值
P8	9	10
P9	10	7
P10	5	5
P11	7	6
P12	9	3
P13	2	8
P14	5	11
P15	5	2

KMeans算法实现流程 - 举例

1、随机设置K个特征空间内的点作为初始的聚类中心（本案例中设置p1和p2）

Euclidean $\sqrt{\sum_{i=1}^{k}(x_i - y_i)^2}$

$$d = \sqrt{(x_n - x_1)^2 + (y_n - y_1)^2}$$



KMeans算法实现流程 - 举例

2、对于其他每个点计算到K个中心的距离，选择最近的一个聚类中心点作为标记类别

	P1（7,7）	P2（2,3）
P3	1.41	6.40
P4	6.71	1.41
P5	7.81	1.41
P6	7.21	2.24
P7	1.41	7.81
P8	3.61	9.90

	P1（7,7）	P2（2,3）
P9	3	8.94
P10	2.83	3.61
P11	1	5.83
P12	4.47	7.00
P13	5.10	5.00
P14	4.47	8.54
P15	5.39	3.16

P1	P3	P7	P8	P9	P10	P11	P12	P14
P2	P4	P5	P6	P13	P15			

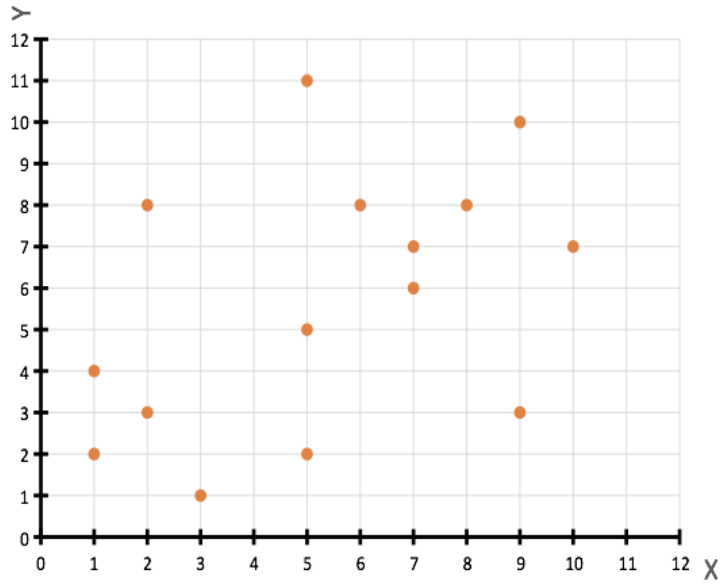
KMeans算法实现流程 - 举例

3、接着对标记的聚类中心，重新计算每个聚类的新中心点（平均值）

$$P_y = \sum_{i=1}^{n} p_{iy} / n \qquad P_x = \sum_{i=1}^{n} p_{ix} / n$$

$$P_1' = (7.3, 7.2)$$

$$P_2' = (1.8, 4.6)$$



KMeans算法实现流程 - 举例

4、如果计算得出的新中心点与原中心点一样（质心不再移动），那么结束，否则重新进行第二步过程

【经过判断，需要重复上述步骤，开始新一轮迭代】

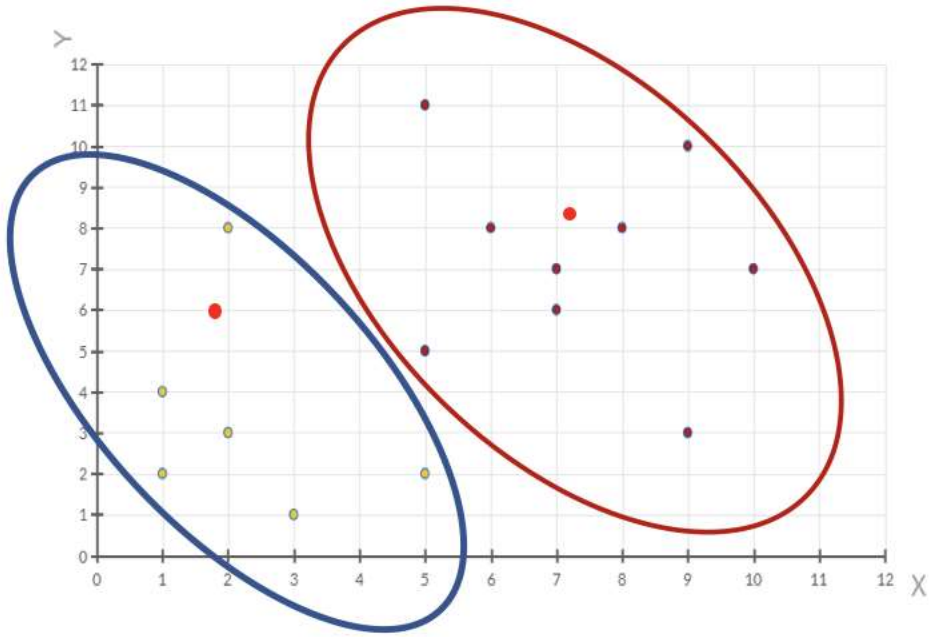
	P'_1 (7.3,7.2)	P'_2 (1.8,4.6)
P1	0.36	5.73
P2	6.75	1.61
P3	1.39	5.40
P4	7.02	1.00
P5	8.16	2.72
P6	7.57	3.79
P7	1.06	7.07

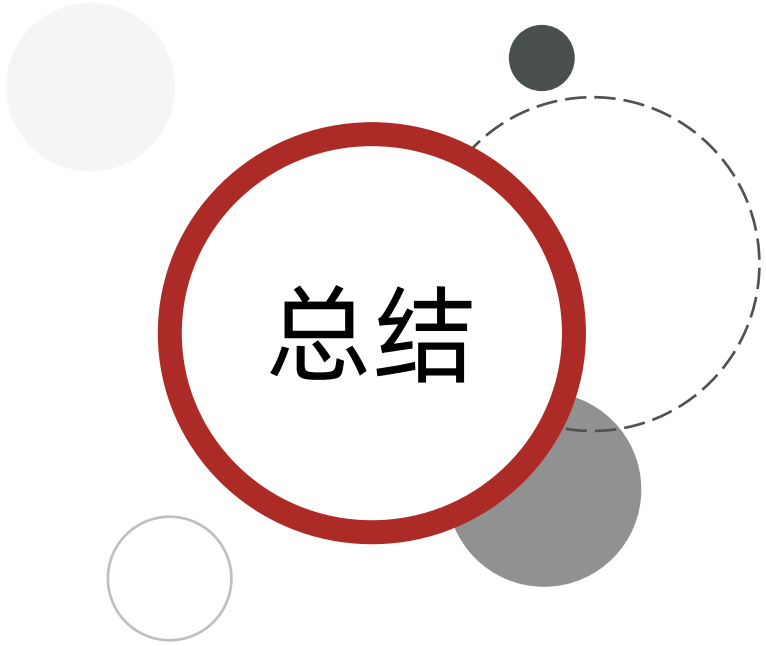
	P'_1 (7.3,7.2)	P'_2 (1.8,4.6)
P8	3.24	9.00
P9	2.82	8.54
P10	3.18	3.22
P11	1.32	5.39
P12	4.66	7.38
P13	5.25	3.41
P14	4.30	7.16
P15	5.25	3.41

P'_1	P1	P3	P7	P8	P9	P10	P11	P12	P14
P'_2	P2	P4	P5	P6	P13	P15			

KMeans算法实现流程 - 举例

5、当每次迭代结果不变时，认为算法收敛，聚类完成





1 KMeans算法实现流程

1 、事先确定常数K ， 常数K意味着最终的聚类类别数

2、随机选择 K 个样本点作为初始聚类中心

3、计算每个样本到 K 个中心的距离，选择最近的聚类中心点作为标记类别

4、根据每个类别中的样本点，重新计算出新的聚类中心点（平均值），如果计算得出的新中心点与原中心点一样则停止聚类，否则重新进行第 2 步过程，直到聚类中心不再变化

练习

1、下列是Kmeans算法的实现流程，请对它们进行排序：

A） 将该未知样本点归类为与D值最小时的中心点相同的类别；

B） 计算未知样本点分别到这K个中心点的距离D；

C） 重复上述过程，直至新的中心点与旧的中心点一致，则迭代停止，将最后这次的聚类作为最优聚类结果。

D） 随机初始化K个中心点；

E） 计算这K个分类簇的均值分别作为这K个簇新的中心点；

答案解析 D在开始 C在最后

答案：D→B→A→E→C

目录

Contents

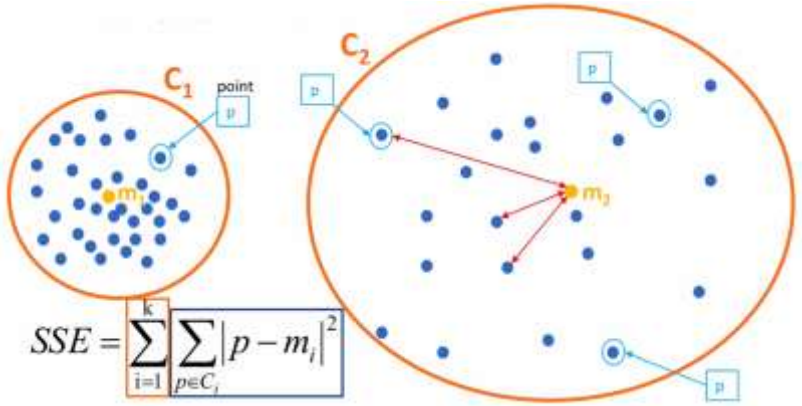
学习目标

Learning Objectives

1. 了解 SSE 聚类评估指标
2. 了解 SC 聚类评估指标
3. 了解 CH 聚类评估指标
4. 了解肘方法的作用

误差平方和SSE (The sum of squares due to error)

$$SSE = \sum_{i=1}^{k} \sum_{p \in C_i} |p - m_i|^2$$

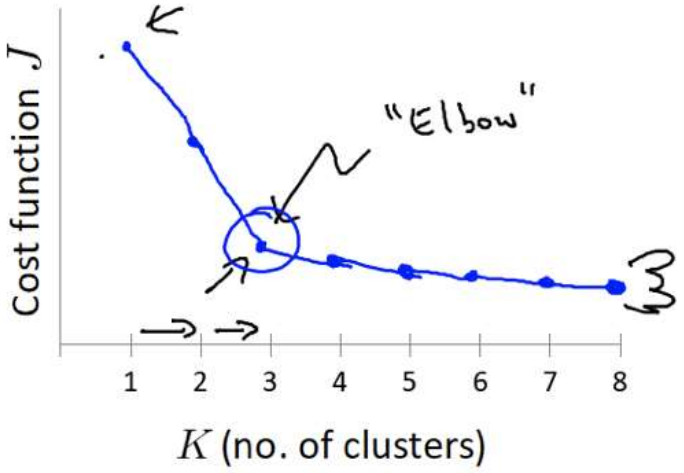


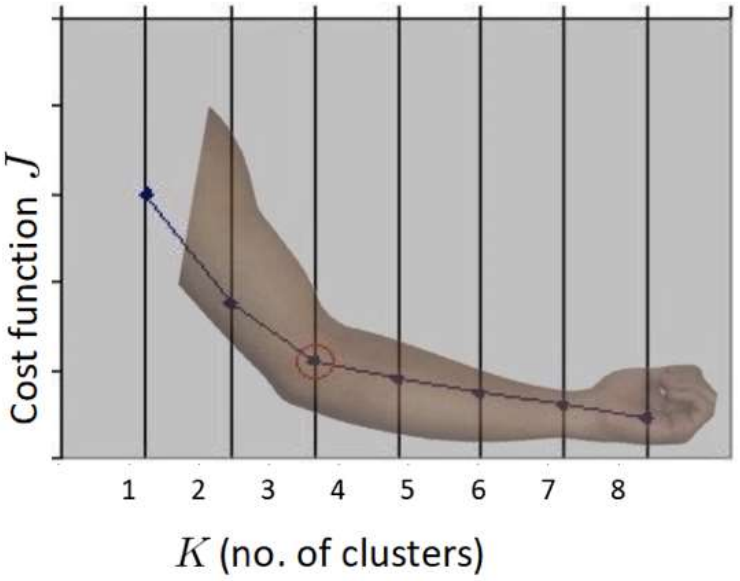
1. C_i 表示簇
2. k 表示聚类中心的个数
3. p 表示某个簇内的样本
4. m 表示质心点

结论：SSE 越小，表示数据点越接近它们的中心，聚类效果越好

“肘”方法 (Elbow method) - K值确定

- "肘" 方法通过 SSE 确定 n_clusters 的值
  - 对于n个点的数据集，迭代计算 k （from 1 to n），每次聚类完成后计算 SSE
  - SSE 是会逐渐变小的，因为每个点都是它所在的簇中心本身。
  - SSE 变化过程中会出现一个拐点，下降率突然变缓时即认为是最佳 n_clusters 值。
  - 在决定什么时候停止训练时，肘形判据同样有效，数据通常有更多的噪音，在增加分类无法带来更多回报时，我们停止增加类别。



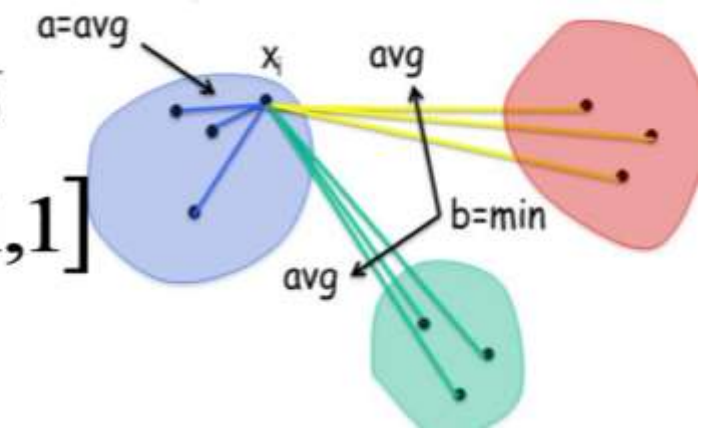


SC轮廓系数法（Silhouette Coefficient）

- 轮廓系数法考虑簇内的内聚程度(Cohesion)，簇外的分离程度(Separation)。其计算过程如下
  - 对计算每一个样本 i 到同簇内其他样本的平均距离a_i，该值越小，说明簇内的相似程度越大
  - 计算每一个样本 i 到最近簇 j 内的所有样本的平均距离 bij，该值越大，说明该样本越不属于其他簇 j
  - 根据下面公式计算该样本的轮廓系数：$S = \frac{b-a}{\max(a,b)}$
  - 计算所有样本的平均轮廓系数
  - 轮廓系数的范围为：[-1, 1]

$$S = \frac{(b-a)}{\max(a,b)}$$

$$S \in [-1,1]$$



a： 样本i到簇内其他点的距离平均值

b：样本i到其他簇间的距离平均值的 最小值

结论：SC值越大，聚类效果越好

聚类效果评估 - CH轮廓系数法（Calinski-Harabasz Index）

- CH 系数考虑簇内的内聚程度、簇外的离散程度、质心的个数
  - 类别内部数据的距离平方和越小越好，类别之间的距离平方和越大越好。聚类的种类数越少越好

$$\mathrm{CH(k)} = \frac{SSB}{SSW}\frac{m-k}{k-1}$$

$$SSW = \sum_{i=1}^{m} \|x_i - C_{pi}\|^2$$

$$SSB = \sum_{j=1}^{k} n_j \left\|C_j - \bar{X}\right\|^2$$

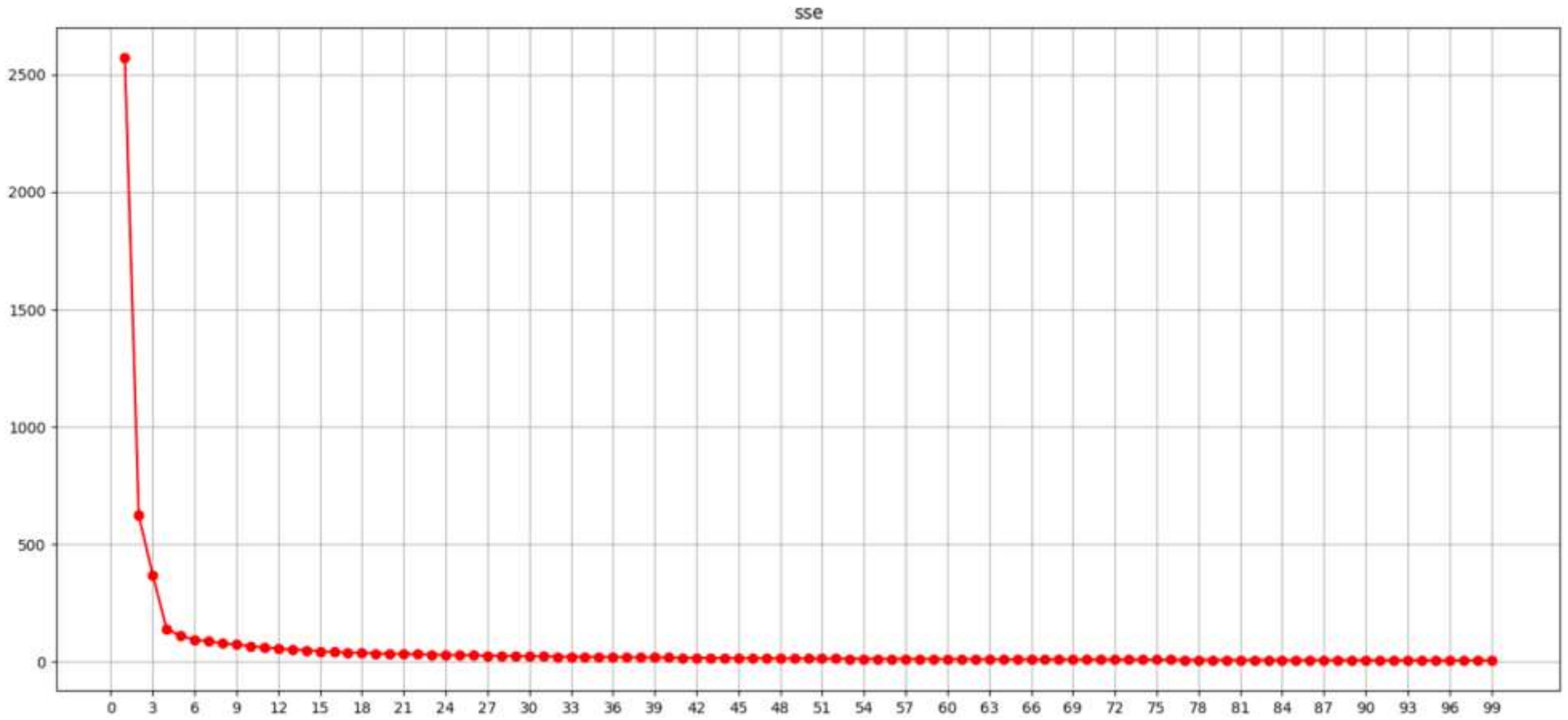
- **SSW 的含义**：相当于SSE，簇内距离
  - C_{pi}表示质心
  - x_i表示某个样本
  - SSW 值是计算每个样本点到质心的距离，并累加起来
  - SSW 表示表示簇内的内聚程度，越小越好
- **SSB 的含义**：簇间距离
  - C_j表示质心，$\bar{X}$ 表示质心与质心之间的中心点，n_j表示样本的个数
  - SSB 表示簇与簇之间的分离程度，SSB 越大越好
  - m 表示样本数量
  - k 表示质心个数

结论：CH值越大，聚类效果越好

聚类效果评估 - 代码效果展示SSE误差平方和

```
# 1.导入依赖包
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt
from sklearn.datasets import make_blobs
from sklearn.metrics import calinski_harabasz_score

def dm01_SSE误差平方和求模型参数():
    # 2.构建数据，产生数据random_state=22固定好
    x, y = make_blobs(n_samples=1000, n_features=2, centers=[[-1, -1], [0, 0], [1, 1], [2, 2]],
                      cluster_std=[0.4, 0.2, 0.2, 0.2], random_state=22)
    # 3.模型训练及SSE
    sse_list = []
    for clu_num in range(1, 100):
        my_kmeans = KMeans(n_clusters=clu_num, max_iter=100, random_state=0)
        my_kmeans.fit(x)
        sse_list.append(my_kmeans.inertia_ ) # 获取SSE的值
    # 4.展示效果
    plt.figure(figsize=(18, 8), dpi=100)
    plt.xticks(range(0, 100, 3), labels=range(0, 100, 3))
    plt.grid()
    plt.title('sse')
    plt.plot(range(1, 100), sse_list, 'or-')
    plt.show()
```

聚类效果评估 - 代码效果展示SSE误差平方和

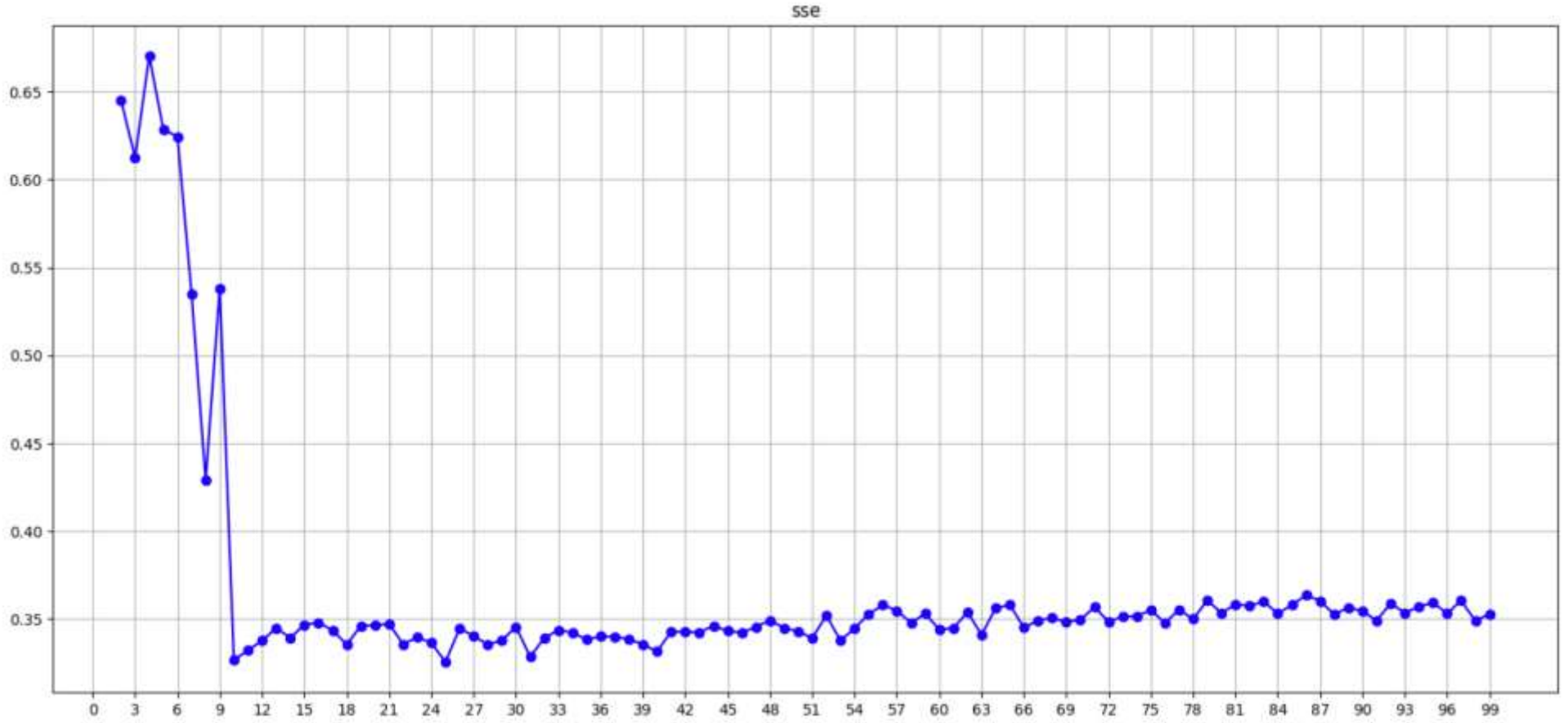


通过图像可观察到 n_clusters=4 sse开始下降趋缓, 最佳值4

聚类效果评估 - 代码效果展示 - SC系数

```
# 1.导入依赖包
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt
from sklearn.datasets import make_blobs
from sklearn.metrics import silhouette_score

def dm02_轮廓系数SC():
    # 2.构建数据，产生数据random_state=22固定好
    x, y = make_blobs(n_samples=1000, n_features=2, centers=[[-1, -1], [0, 0], [1, 1], [2, 2]],
                      cluster_std=[0.4, 0.2, 0.2, 0.2], random_state=22)
    # 3.模型训练及SC系数
    tmp_list = []
    for clu_num in range(2, 100):
        my_kmeans = KMeans(n_clusters=clu_num, max_iter=100, random_state=0)
        my_kmeans.fit(x)
        ret = my_kmeans.predict(x)
        tmp_list.append(silhouette_score(x, ret))  # SC系数
    # 4.效果展示
    plt.figure(figsize=(18, 8), dpi=100)
    plt.xticks(range(0, 100, 3), labels=range(0, 100, 3))
    plt.grid()
    plt.title('sse')
    plt.plot(range(2, 100), tmp_list, 'ob-')
    plt.show()
```

聚类效果评估 - 代码效果展示 - SC系数

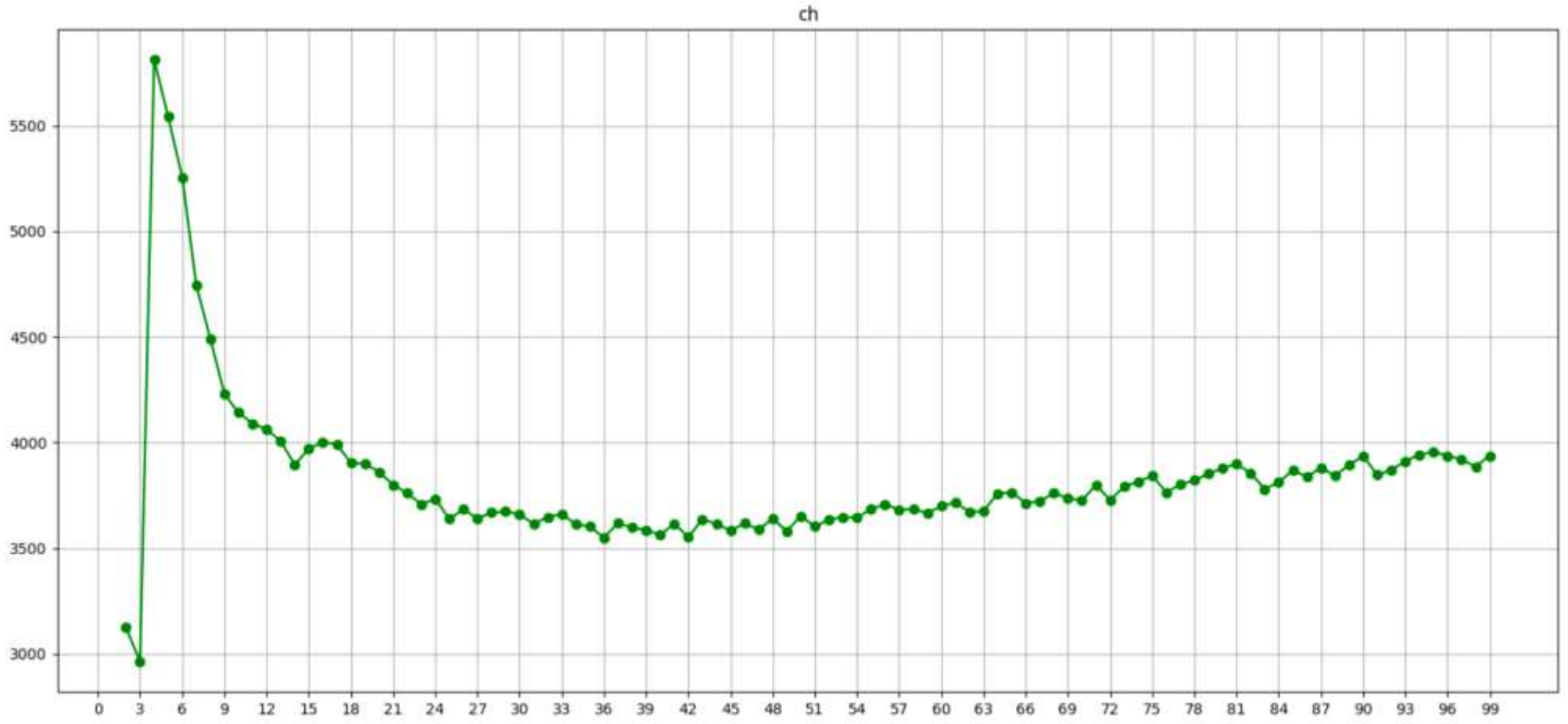


通过图像可观察到 n_clusters=4 取到最大值； 最佳值4

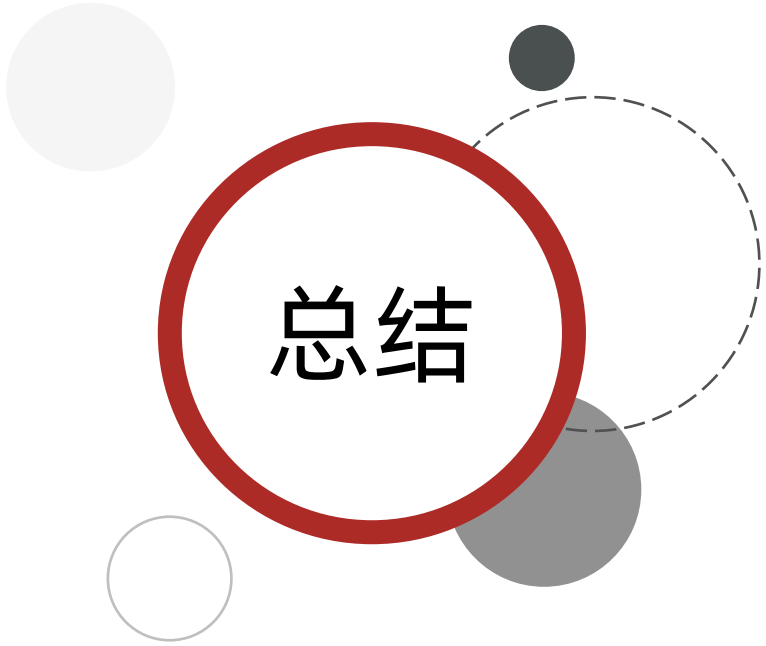
聚类效果评估 - 代码效果展示 - CH系数

```
# 1.导入依赖包
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt
from sklearn.datasets import make_blobs
from sklearn.metrics import calinski_harabasz_score

def dm03_ch系数():
    # 2.构建数据，产生数据random_state=22固定好
    x, y = make_blobs(n_samples=1000, n_features=2, centers=[[-1, -1], [0, 0], [1, 1], [2, 2]],
                      cluster_std=[0.4, 0.2, 0.2, 0.2], random_state=22)
    # 3.模型训练及CH
    tmp_list = []
    for clu_num in range(2, 100):
        my_kmeans = KMeans(n_clusters=clu_num, max_iter=100, random_state=0)
        my_kmeans.fit(x)
        ret = my_kmeans.predict(x)
        tmp_list.append(calinski_harabasz_score(x, ret))  # CH
    # 4.展示效果
    plt.figure(figsize=(18, 8), dpi=100)
    plt.xticks(range(0, 100, 3), labels=range(0, 100, 3))
    plt.grid()
    plt.title('ch')
    plt.plot(range(2, 100), tmp_list, 'og-')
    plt.show()
```

聚类效果评估 - 代码效果展示 - CH系数



通过图像可观察到 n_clusters=4 取到最大值； 最佳值4



1 误差平方和SSE

- 误差平方和的值越小，聚类效果越好
- 主要考量：簇内聚程度

2 肘部法

- 下降率突然变缓时即认为是最佳的k值

3 SC系数

- 取值为[-1, 1]，SC系数值越大，聚类效果越好
- 主要考量：簇内聚程度、簇间分离程度

4 CH系数

- CH分数越高，则聚类效果越好
- CH达到的目的：用尽量少的类别聚类尽量多的样本，同时获得较好的聚类效果
- 主要考量：簇内聚程度、簇间分离程度、质心个数

练习

1、下列可用于评估聚类算法的方法或指标的是：（**多选**）

A） SSE：误差平方和

B） 肘部法

C） Silhouette Coefficient： SC系数

D） Calinski-Harabasz Index： CH指数

答案：ACD

目录

Contents

学习目标

Learning Objectives

1. 能使用聚类算法完成客户案例分析
2. 知道怎么求最佳K值

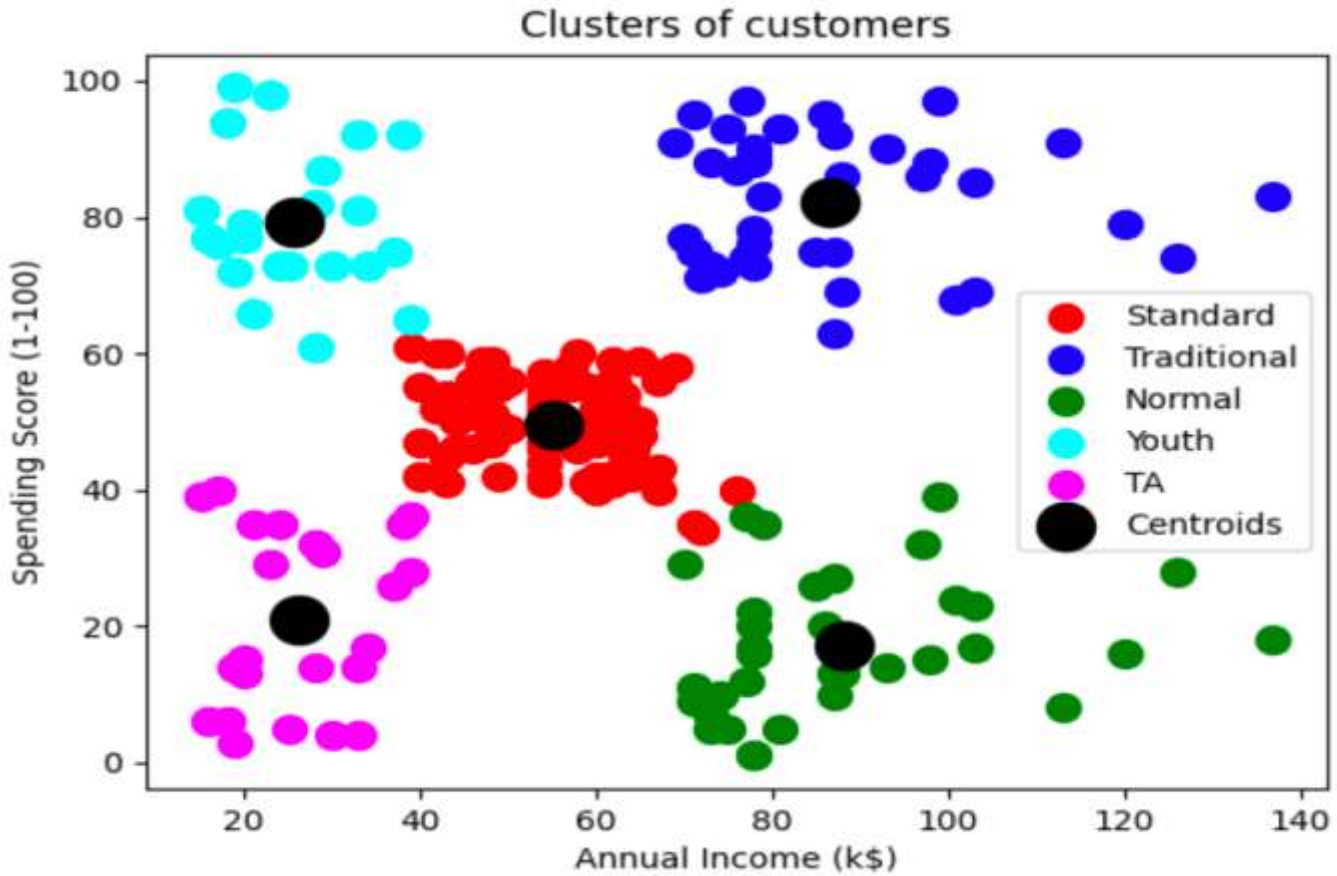
案例：顾客数据聚类分析

- 已知：客户性别、年龄、年收入、消费指数
- 需求：对客户进行分析，找到业务突破口，寻找黄金客户

	A	B	C	D	E
1	CustomerID	Gender	Age	Annual Income (k$)	Spending Score (1-100)
2	1	Male	19	15	39
3	2	Male	21	15	81
4	3	Female	20	16	6
195	194	Female	38	113	91
196	195	Female	47	120	16
197	196	Female	35	120	79
198	197	Female	45	126	28
199	198	Male	32	126	74
200	199	Male	32	137	18
201	200	Male	30	137	83

案例：顾客数据聚类分析

- 客户分群效果展示：



从图中可以看出，聚成5类，右上角属于挣的多，消费的也多黄金客户群

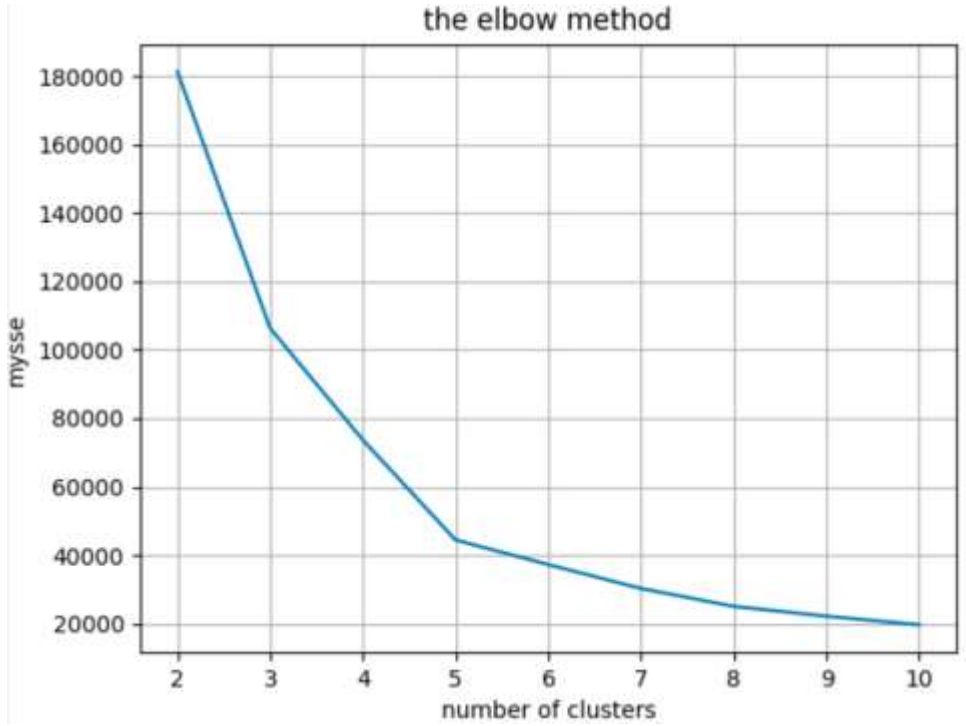
案例：顾客数据聚类分析

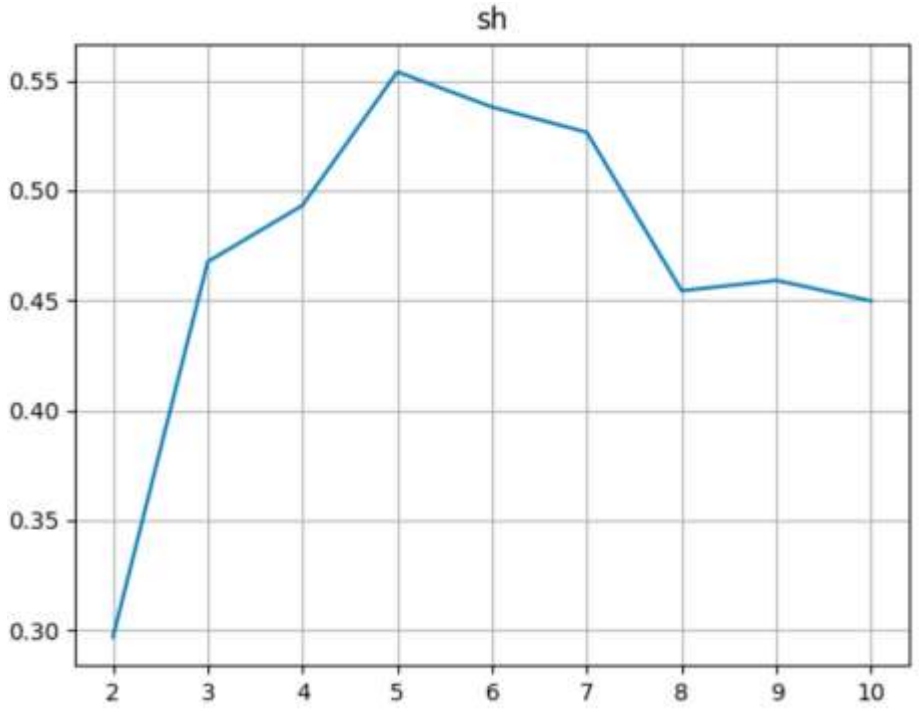
```
# 1.导入依赖包
import pandas as pd
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt
from sklearn.metrics import silhouette_score

# 聚类分析用户分群
def dm01_聚类分析用户群():
    # 2.数据读取及预处理
    # 2.1 数据读取
    dataset = pd.read_csv('data/customers.csv')
    print(dataset.head)
    # 2.2 特征选择
    X = dataset.iloc[:, [3, 4]]
    print('X-->\n', X)

    # 3.模型训练，评估聚类个数K值选择
    mysse = []
    mysscore = []
    for i in range(2, 11):
        mykeans = KMeans(n_clusters=i)
        mykeans.fit(X)
        mysse.append(mykeans.inertia_)     # inertia 簇内误差平方和
        ret = mykeans.predict(X)
        mysscore.append(silhouette_score(X, ret))   # SC系数 聚类需要1个以上的类别
```

```
    # 效果展示
    plt.plot(range(2, 11), mysse)
    plt.title('the elbow method')
    plt.xlabel('number of clusters')
    plt.ylabel('mysse')
    plt.grid()
    plt.show()

    plt.title('sh')
    plt.plot(range(2, 11), mysscore)
    plt.grid(True)
    plt.show()
```

案例：顾客数据聚类分析

- 效果分析：





通过肘方法、SH系数都可以看出，聚成5类效果最好

案例：顾客数据聚类分析

```
def dm02_聚类分析用户群():
    # 2.读取数据及数据预处理
    dataset = pd.read_csv('data/customers.csv')
    X = dataset.iloc[:, [3, 4]]
    # 3.模型训练及预测
    mykeans = KMeans(n_clusters=5)
    mykeans.fit(X)
    y_kmeans = mykeans.predict(X)
    # 4.聚类效果展示
    # 把类别是0的, 第0列数据,第1列数据, 作为x/y, 传给plt.scatter函数
    plt.scatter(X.values[y_kmeans == 0, 0], X.values[y_kmeans == 0, 1], s=100, c='red', label='Standard')
    # 把类别是1的, 第0列数据,第1列数据, 作为x/y, 传给plt.scatter函数
    plt.scatter(X.values[y_kmeans == 1, 0], X.values[y_kmeans == 1, 1], s=100, c='blue', label='Traditional')
    # 把类别是2的, 第0列数据,第1列数据, 作为x/y, 传给plt.scatter函数
    plt.scatter(X.values[y_kmeans == 2, 0], X.values[y_kmeans == 2, 1], s=100, c='green', label='Normal')
    plt.scatter(X.values[y_kmeans == 3, 0], X.values[y_kmeans == 3, 1], s=100, c='cyan', label='Youth')
    plt.scatter(X.values[y_kmeans == 4, 0], X.values[y_kmeans == 4, 1], s=100, c='magenta', label='TA')
    plt.scatter(mykeans.cluster_centers_[:, 0], mykeans.cluster_centers_[:, 1], s=300, c='black', label='Centroids')

    plt.title('Clusters of customers')
    plt.xlabel('Annual Income (k$)')
    plt.ylabel('Spending Score (1-100)')
    plt.legend()
    plt.show()
```


黑马程序员
www.itheima.com
传智教育旗下高端IT教育品牌