机器学习-第十章 聚类

黄亮 副教授

2023年秋

http://www.homepage.zjut.edu.cn/lianghuang/

本章目录

1.无监督学习概述

1.无监督学习方法概述

监督学习和无监督学习的区别

监督学习

在一个典型的监督学习中，训练集有标签y，我们的目标是找到能够区分正样本和负样本的决策边界，需要据此拟合一个假设函数。

无监督学习

与此不同的是，在无监督学习中，我们的数据没有附带任何标签y，无监督学习主要分为聚类、降维、关联规则、推荐系统等方面。

1.无监督学习方法概述

主要的无监督学习方法

- ✓ 聚类（Clustering）
  - ◆ 如何将教室里的学生按爱好、身高划分为5类?
- ✓ 降维（ Dimensionality Reduction ）
  - ◆ 如何将将原高维空间中的数据点映射到低维度的空间中?
- ✓ 关联规则（ Association Rules）
  - ◆ 很多买尿布的男顾客，同时买了啤酒，可以从中找出什么规律来提高超市销售额?
- ✓ 推荐系统（ Recommender systems）
  - ◆ 很多客户经常上网购物，根据他们的浏览商品的习惯，给他们推荐什么商品呢?

1.无监督学习方法概述

聚类

主要算法：
K-means、密度聚类、层次聚类

主要应用：

市场细分、文档聚类、图像分割、图像压缩、聚类分析、特征学习或者词典学习、确定犯罪易发地区、保险欺诈检测、公共交通数据分析、IT资产集群、客户细分、识别癌症数据、搜索引擎应用、医疗应用、药物活性预测……

1.无监督学习方法概述

聚类案例

1.医疗

医生可以使用聚类算法来发现疾病。

以甲状腺疾病为例。当我们对包含甲状腺疾病和非甲状腺疾病的数据集应用无监督学习时，可以使用聚类算法来识别甲状腺疾病数据集。

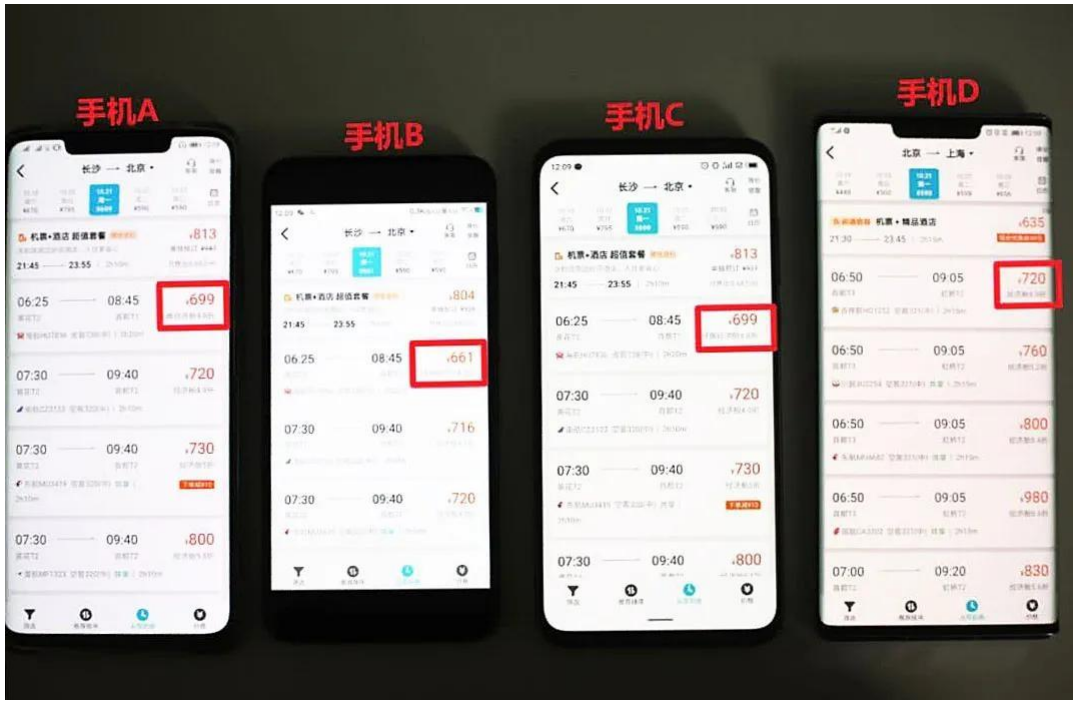
1.无监督学习方法概述

聚类案例

2.市场细分

为了吸引更多的客户，每家公司都在开发易于使用的功能和技术。公司需要了解客户，发现客户之间的相似之处，并对他们进行分组。

为了了解客户，公司可以使用聚类。聚类将帮助公司了解用户群，然后对每个客户进行归类。



1.无监督学习方法概述

聚类案例

3.金融业

银行可以观察到可能的金融欺诈行为，就此向客户发出警告。

在聚类算法的帮助下，保险公司可以发现某些客户的欺诈行为，并调查类似客户的保单是否有欺诈行为。

1.无监督学习方法概述

聚类案例

4.搜索引擎

百度是人们使用的搜索引擎之一。

举个例子，当我们搜索一些信息，如在某地的超市，百度将为我们提供不同的超市的选择。这是聚类的结果，提供给你的结果就是聚类的相似结果。

1.无监督学习方法概述

聚类案例

5.社交网络

比如在社交网络的分析上。已知你朋友的信息，比如经常发email的联系人，或是你的微博好友、微信的朋友圈，我们可运用聚类方法自动地给朋友进行分组，做到让每组里的人们彼此都熟识。

2.K-means聚类

2.K-means聚类

K-均值算法(K-means)算法概述

K-means算法是一种**无监督学习**方法，是最普及的聚类算法，算法使用一个**没有标签**的数据集，然后将数据聚类成不同的组。

K-means算法具有一个迭代过程，在这个过程中，数据集被分组成若干个预定义的不重叠的聚类或子组，使簇的内部点尽可能相似，同时试图保持簇在不同的空间，它将数据点分配给簇，以便**簇的质心和数据点之间的平方距离之和最小**，在这个位置，簇的质心是簇中数据点的算术平均值。

距离度量

闵可夫斯基距离(Minkowski distance)

p取1或2时的闵氏距离是最为常用的

$p = 2$即为欧氏距离

$p = 1$时则为曼哈顿距离

$p = \infty$，可以得到切比雪夫距离

$$d(x,y) = \left(\sum_i |x_i - y_i|^p \right)^{\frac{1}{p}}$$

欧氏距离：$d(x,y) = \left(\sum_i |x_i - y_i|^2 \right)^{\frac{1}{2}}$

2.K-means聚类

K-means算法流程

1：选择K个点作为初始质心。

2：将每个点指派到最近的质心，形成K个簇。

3：对于上一步聚类的结果，进行平均计算，得出该簇的新的聚类中心。

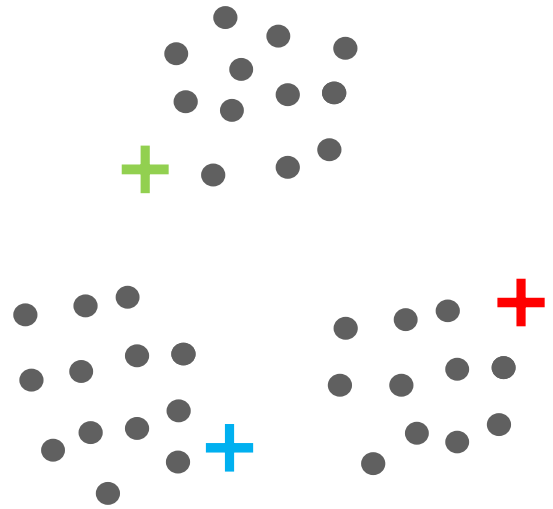
4：重复上述两步/直到迭代结束：质心不发生变化。

2.K-means聚类

K-means算法流程

首先，初始化称为簇质心的任意点。

初始化时，必须注意簇的质心必须小于训练数据点的数目。因为该算法是一种迭代算法，接下来的两个步骤是迭代执行的。

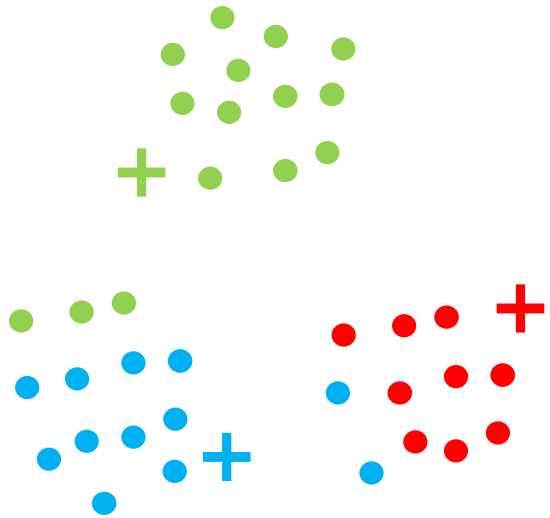


初始化质心

2.K-means聚类

K-means算法流程

初始化后，遍历所有数据点，计算所有质心与数据点之间的距离。现在，这些簇将根据与质心的最小距离而形成。在本例中，数据分为3个簇($K = 3$)。



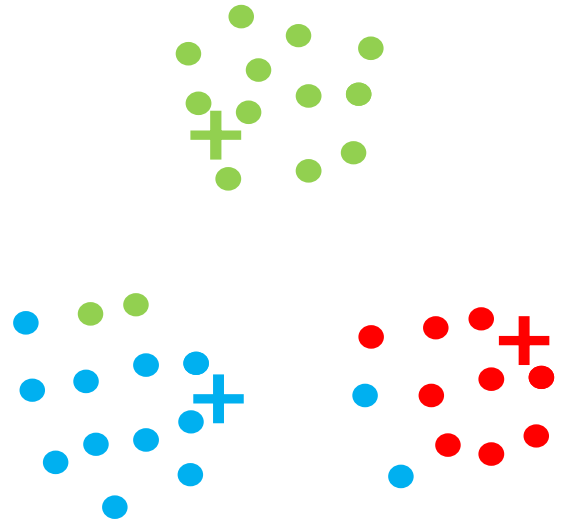
簇赋值

2.K-means聚类

K-means算法流程

第三步：移动质心，因为上面步骤中形成的簇没有优化，所以需要形成优化的簇。为此，我们需要迭代地将质心移动到一个新位置。

1. 取一个簇的数据点，
2. 计算它们的平均值，
3. 然后将该簇的质心移动到这个新位置。
4. 对所有其他簇重复相同的步骤。



迭代更新

2.K-means聚类

K-means算法流程

优化

上述两个步骤是迭代进行的，直到质心停止移动，即它们不再改变自己的位置，并且成为静态的。一旦这样做，k-均值算法被称为收敛。

K-均值的代价函数（又称**畸变函数 Distortion function**）为：

$$J(c^{(1)},\ldots,c^{(m)},\mu_1,\ldots,\mu_K)=\frac{1}{m}\sum_{i=1}^{m}\|X^{(i)}-\mu_{c^{(i)}}\|^2$$

设训练集为：$\{x^{(1)},x^{(2)},x^{(3)},\ldots,x^{(m)}\}$，簇划分$C=\{C_1,C_2,\cdots,C_K\}$，用$\mu_1,\mu_2,\ldots,\mu_K$来表示聚类中心

其中$\mu_{c^{(i)}}$代表与$x^{(i)}$最近的聚类中心点。

我们的的优化目标便是找出使得代价函数最小的 $c^{(1)}$,$c^{(2)}$,...,$c^{(m)}$和 $\mu_1,\mu_2,\ldots,\mu_K$ 。

2.K-means聚类

K-means优化过程

记k个簇中心为$\mu_1, \mu_2, \ldots, \mu_k$，每个簇的样本数目为$N_1$, N_2,..., N_k

使用平方误差作为目标函数:

$$J(\mu_1, \mu_2, \cdots \mu_k) = \frac{1}{2}\sum_{j=1}^{K}\sum_{i=1}^{N_j}(x_i - \mu_j)^2$$

对关于从$\mu_1, \mu_2, \cdots \mu_k$的函数求偏导，这里的求偏导是对第$j$个簇心$\mu_j$求的偏导。故而其驻点为:

$$\frac{\partial J}{\partial \mu_j} = -\sum_{i=1}^{N_j}(x_i - \mu_j) \overset{令}{\rightarrow} 0 \Rightarrow \mu_j = \frac{1}{N_j}\sum_{i=1}^{N_j} x_i$$

推导:

$$\frac{\partial J}{\partial \mu_j} = \frac{\partial \frac{1}{2}\sum_{j=1}^{k}\sum_{i=1}^{N_j}(x_i - \mu_j)^2}{\partial \mu_j}$$

$$= \frac{\partial \frac{1}{2}\sum_{i=1}^{N_j}(x_i - \mu_j)^2}{\partial \mu_j}$$

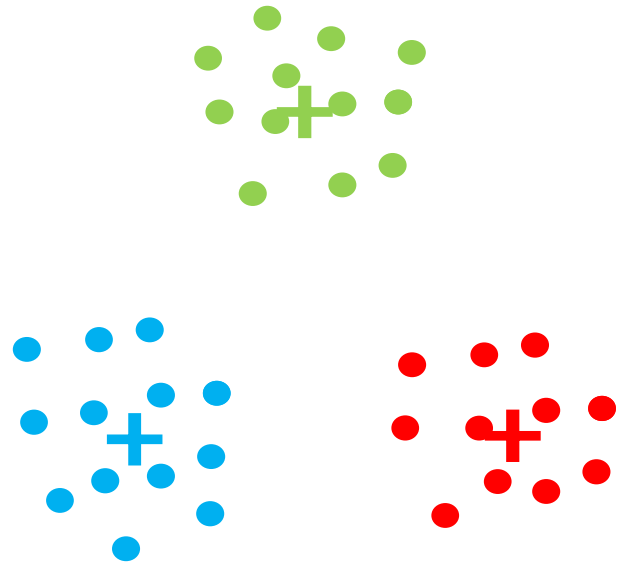
$$= \sum_{i=1}^{N_j}(x_i - \mu_j)\cdot(-1)$$

$$= -\sum_{i=1}^{N_j}(x_i - \mu_j)$$

2.K-means聚类

K-means算法流程

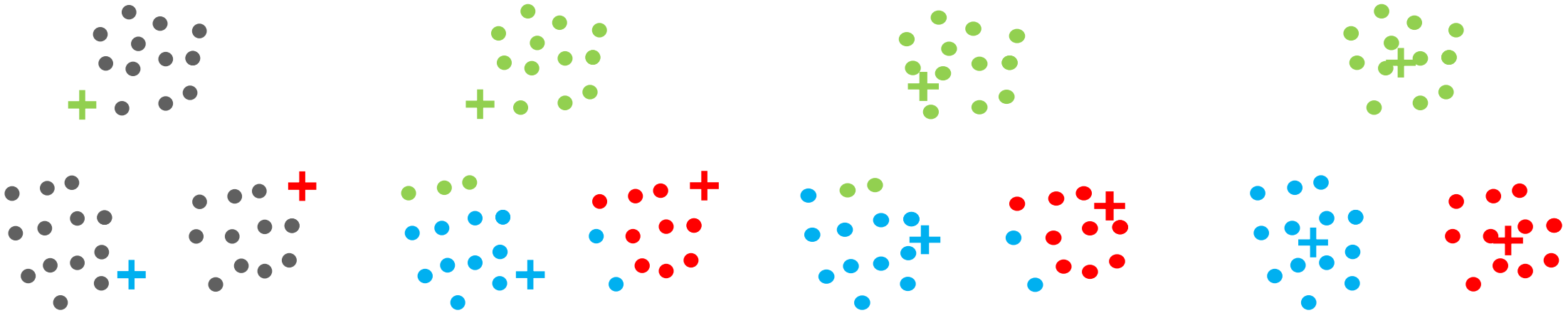
现在，这个算法已经收敛，形成了清晰可见的不同簇。该算法可以根据簇在第一步中的初始化方式给出不同的结果。



收敛

2.K-means聚类

K-means算法流程总结

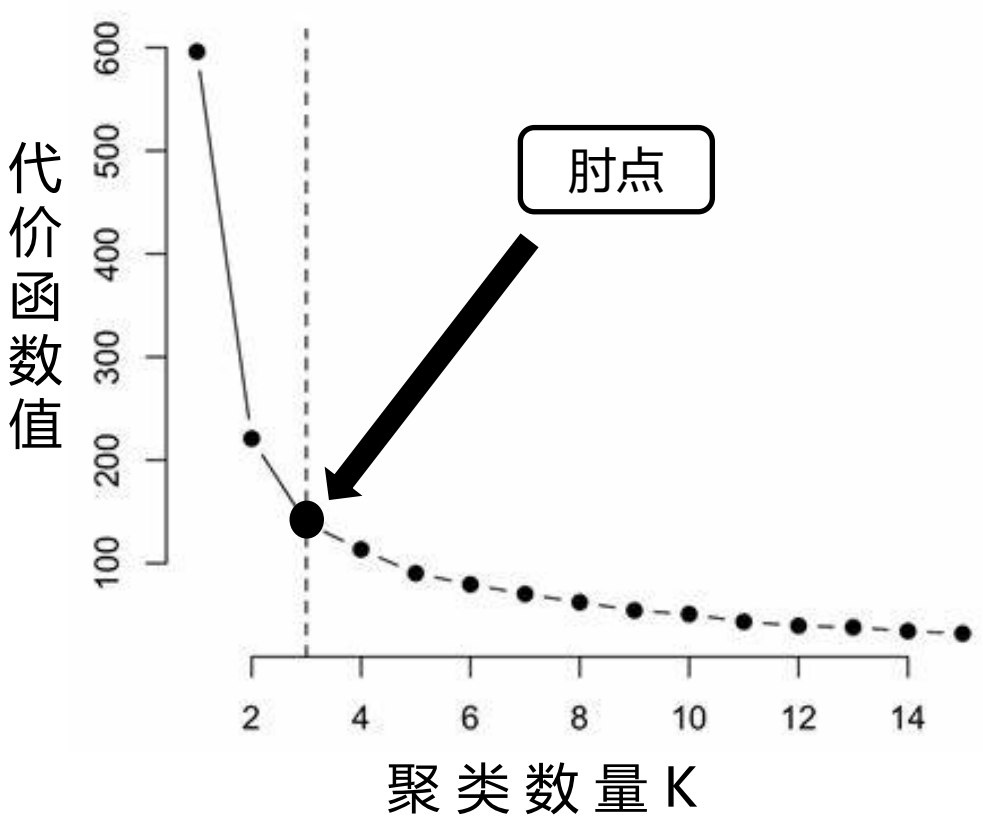


初始化质心　　簇赋值　　迭代更新　　收敛

2.K-means聚类

K值的选择

现在我们需要找到簇的数量。通常通过“肘部法则”进行计算。我们可能会得到一条类似于人的肘部的曲线。右图中，代价函数的值会迅速下降，在$K = 3$的时候达到一个肘点。在此之后，代价函数的值会就下降得非常慢，所以，我们选择$K = 3$。这个方法叫“肘部法则”。



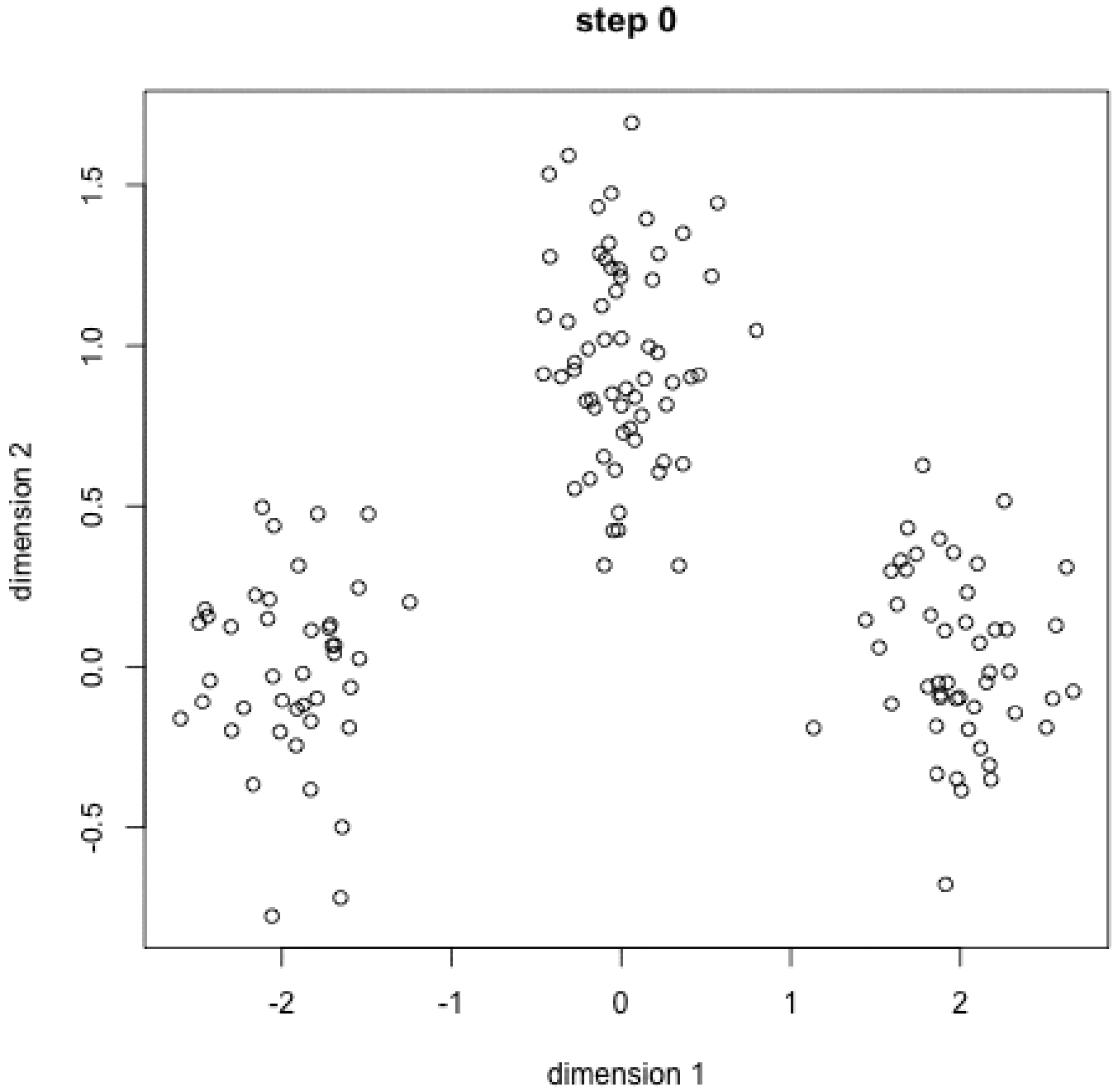
***K-均值**的一个问题在于，它有可能会停留在一个局部最小值处，而这取决于初始化的情况。*

*为了解决这个问题，我们通常需要多次运行**K-均值**算法，每一次都重新进行随机初始化，最后再比较多次运行**K-均值**的结果，选择代价函数最小的结果。*

2.K-means聚类

K-means的优点

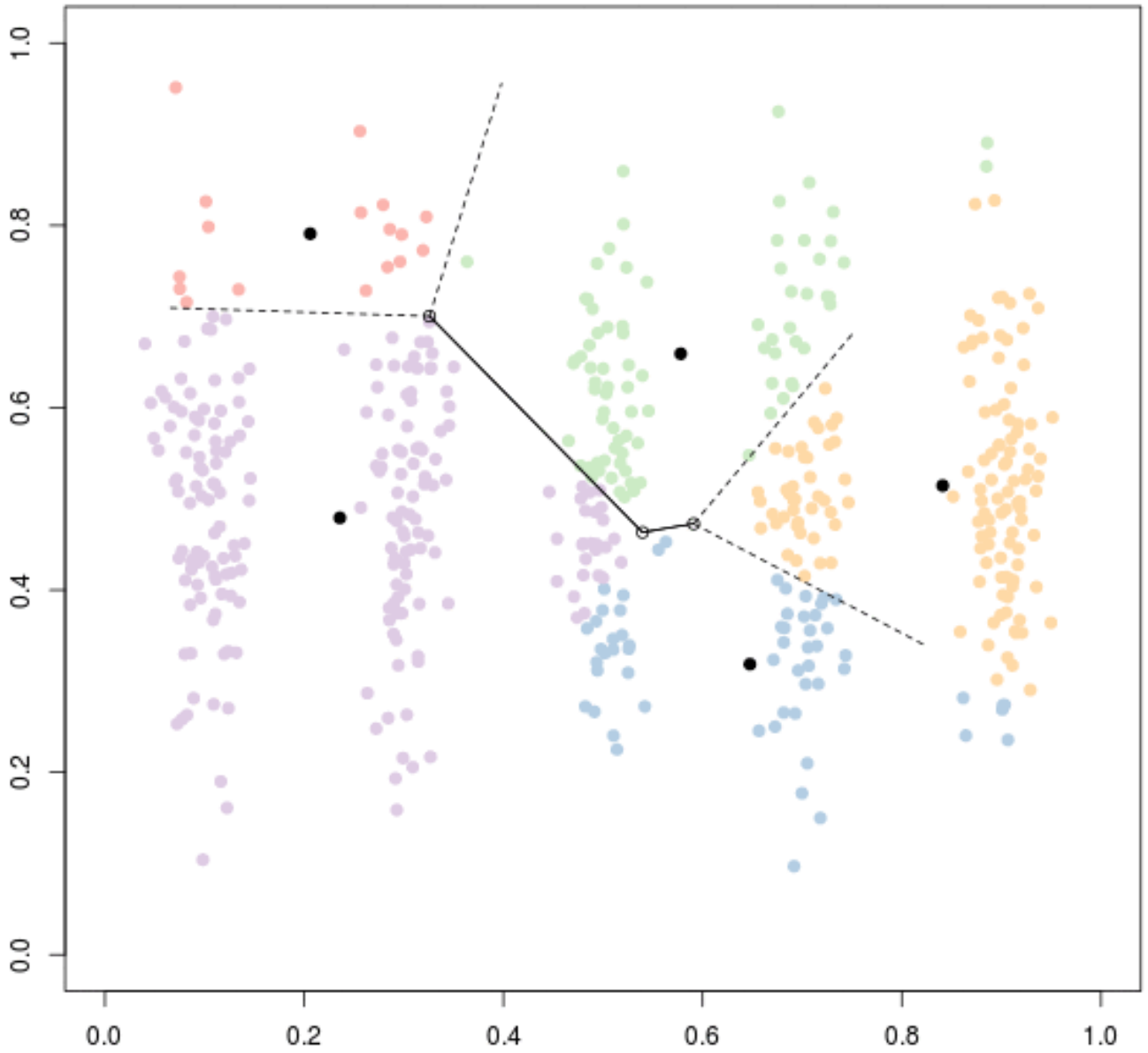
- 鲁棒性高；
- 速度快、易于理解、效率高；
- 计算成本低、灵活性高；
- 如果数据集是不同的，则结果更好；
- 可以产生更紧密的簇；
- 重新计算质心时，簇会发生变化。



2.K-means聚类

K-means的缺点

- 需要预先指定簇的数量；
- 如果有两个高度重叠的数据，那么它就不能被区分，也不能判断有两个簇；
- 欧几里德距离可以不平等的权重因素，限制了能处理的数据变量的类型；
- 有时随机选择质心并不能带来理想的结果；
- 无法处理异常值和噪声数据；
- 不适用于非线性数据集；
- 对特征尺度敏感；
- 如果遇到非常大的数据集，那么计算机可能会崩溃。



3.密度聚类和层次聚类

3.密度聚类和层次聚类

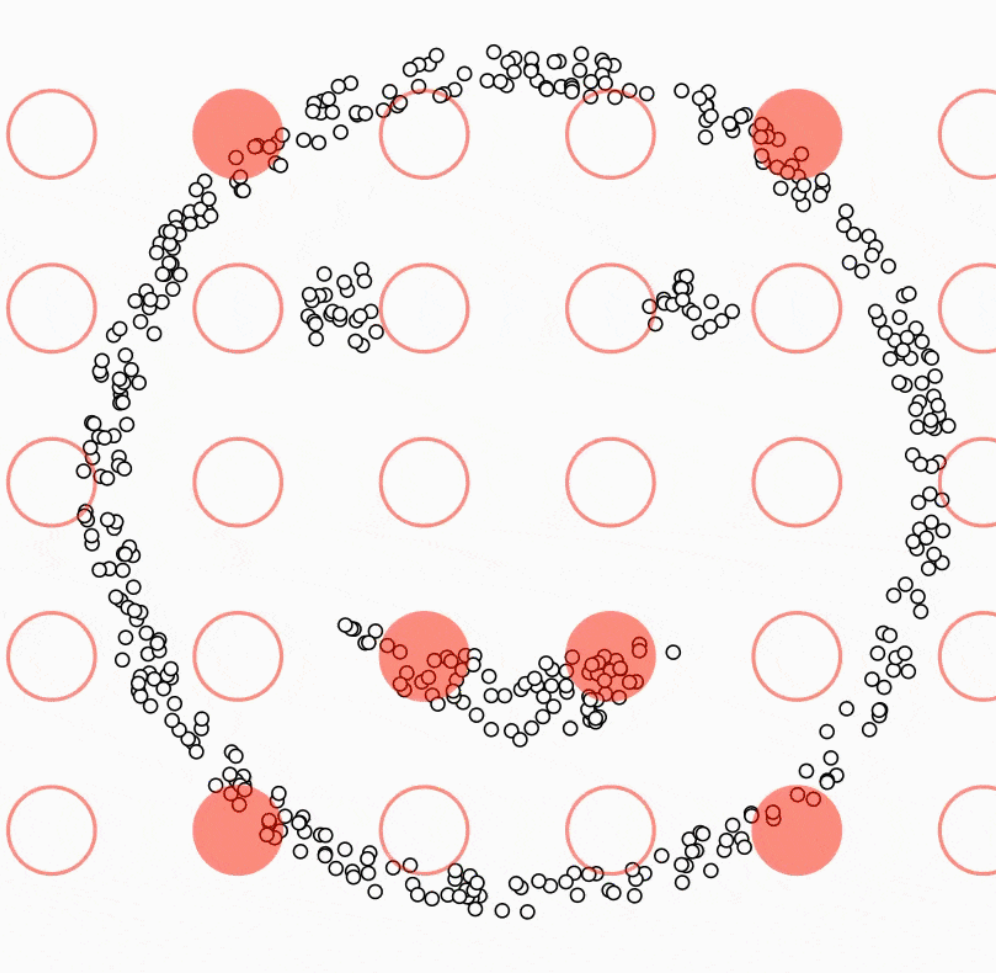
DBSCAN密度聚类

DBSCAN(Density-Based Spatial Clustering of Applications with Noise)是一个比较有代表性的基于密度的聚类算法。

与划分和层次聚类方法不同，它将簇定义为密度相连的点的最大集合，能够把具有足够高密度的区域划分为簇，并可在噪声的空间数据库中发现任意形状的聚类。

3.密度聚类和层次聚类

DBSCAN密度聚类

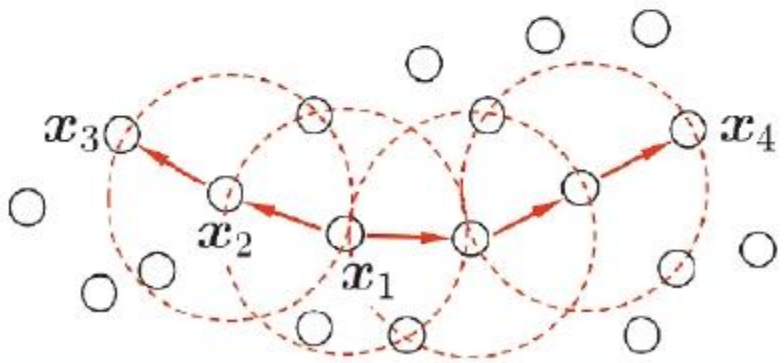


关键概念：

- 核心对象(core object)：若 $\boldsymbol{x}_j$ 的 ϵ-邻域至少包含 $MinPts$ 个样本，即 $|N_\epsilon(\boldsymbol{x}_j)| \geqslant MinPts$，则 $\boldsymbol{x}_j$ 是一个核心对象；
- 密度直达(directly density-reachable)：若 $\boldsymbol{x}_j$ 位于 $\boldsymbol{x}_i$ 的 ϵ-邻域中，且 $\boldsymbol{x}_i$ 是核心对象，则称 $\boldsymbol{x}_j$ 由 $\boldsymbol{x}_i$ 密度直达；
- 密度可达(density-reachable)：对 $\boldsymbol{x}_i$ 与 $\boldsymbol{x}_j$，若存在样本序列 $\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_n$，其中 $\boldsymbol{p}_1 = \boldsymbol{x}_i$，$\boldsymbol{p}_n = \boldsymbol{x}_j$ 且 $\boldsymbol{p}_{i+1}$ 由 $\boldsymbol{p}_i$ 密度直达，则称 $\boldsymbol{x}_j$ 由 $\boldsymbol{x}_i$ 密度可达；
- 密度相连(density-connected)：对 $\boldsymbol{x}_i$ 与 $\boldsymbol{x}_j$，若存在 $\boldsymbol{x}_k$ 使得 $\boldsymbol{x}_i$ 与 $\boldsymbol{x}_j$ 均由 $\boldsymbol{x}_k$ 密度可达，则称 $\boldsymbol{x}_i$ 与 $\boldsymbol{x}_j$ 密度相连.

令$MinPts = 3$，
虚线显示出ϵ 邻域
x_1 是核心对象

x_2 由 x_1 密度直达
x_3 由 x_1 密度可达
x_3 与 x_4 密度相连



3.密度聚类和层次聚类

在DBSCAN使用两个超参数:

扫描半径 (eps)和最小包含点数(minPts)来获得簇的数量，而不是猜测簇的数目。

扫描半径 (eps)：

用于定位点/检查任何点附近密度的距离度量，即扫描半径。

最小包含点数(minPts)：

聚集在一起的最小点数（阈值），该区域被认为是稠密的。

3.密度聚类和层次聚类

DBSCAN

k-means

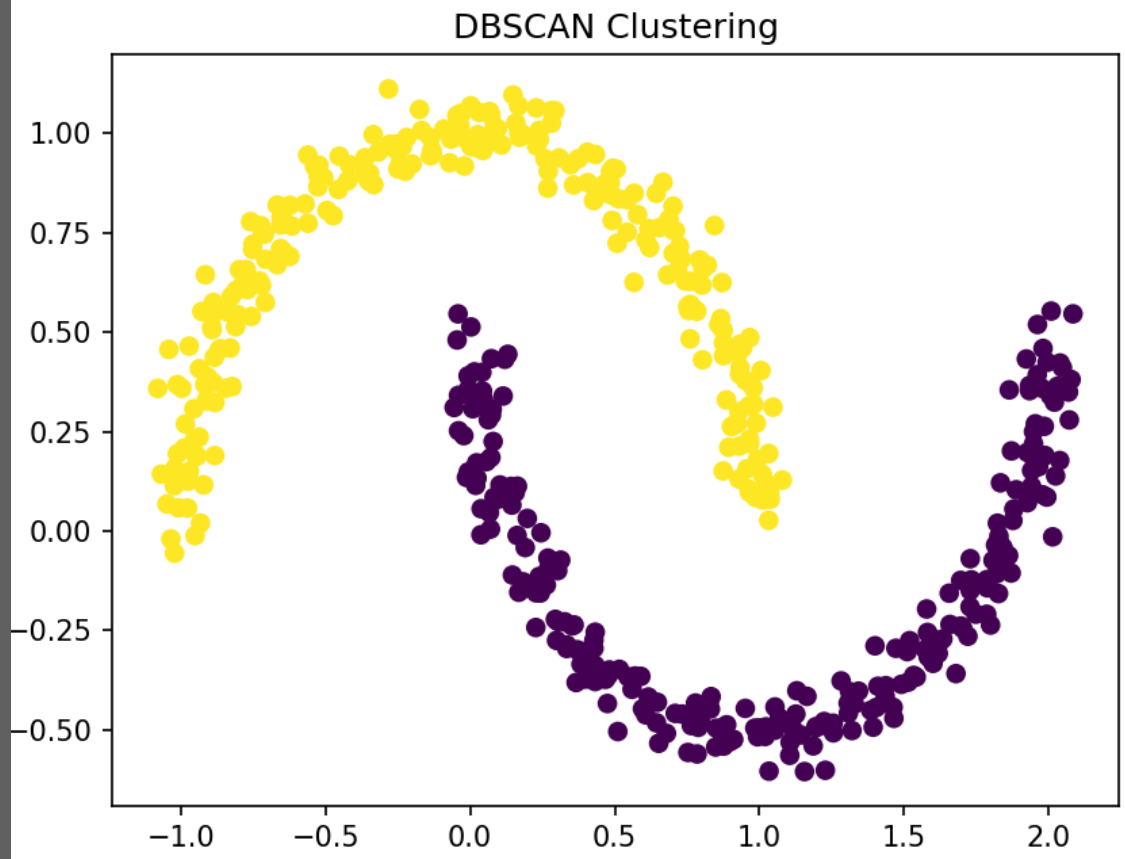
3.密度聚类和层次聚类


```
# DBSCAN example code
from sklearn.cluster import DBSCAN
from sklearn.datasets import make_moons
import matplotlib.pyplot as plt

# 生成样本数据
X, y = make_moons(n_samples=500, noise=0.05,
random_state=42)

# DBSCAN
dbscan = DBSCAN(eps=0.3, min_samples=5)
clusters = dbscan.fit_predict(X)

# 绘制结果
plt.scatter(X[:, 0], X[:, 1], c=clusters,
cmap='viridis')
plt.title("DBSCAN Clustering")
plt.show()
```



3.密度聚类和层次聚类

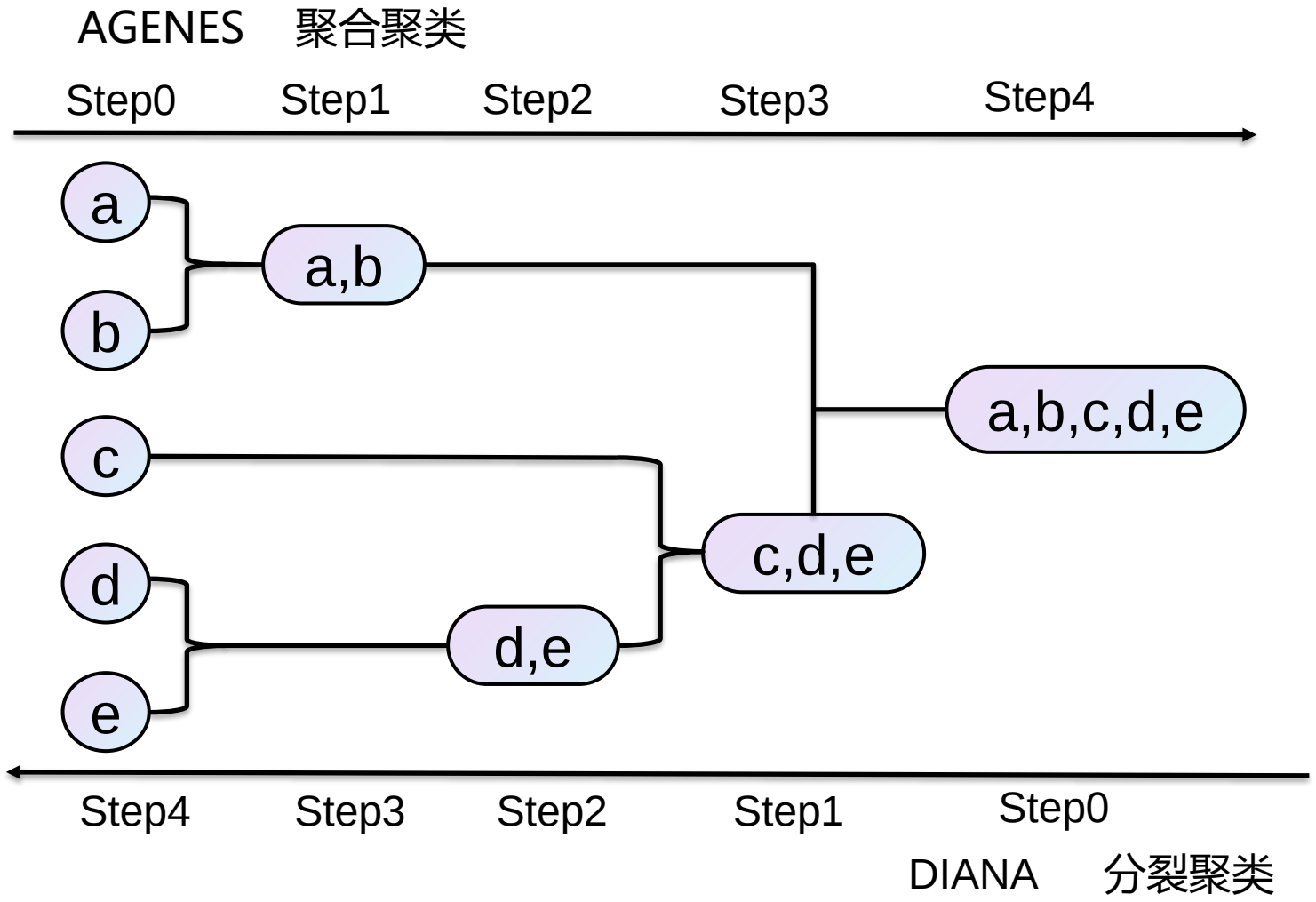
层次聚类

- 层次聚类假设簇之间存在层次结构，将样本聚到层次化的簇中。
- 层次聚类又有聚合聚类（自下而上）、分裂聚类（自上而下）两种方法。
- 因为每个样本只属于一个簇，所以层次聚类属于硬聚类。

背景知识：如果一个聚类方法假定一个样本只能属于一个簇，或簇的交集为空集，那么该方法称为硬聚类方法。

如果一个样本可以属于多个簇，或簇的交集不为空集，那么该方法称为软聚类方法。

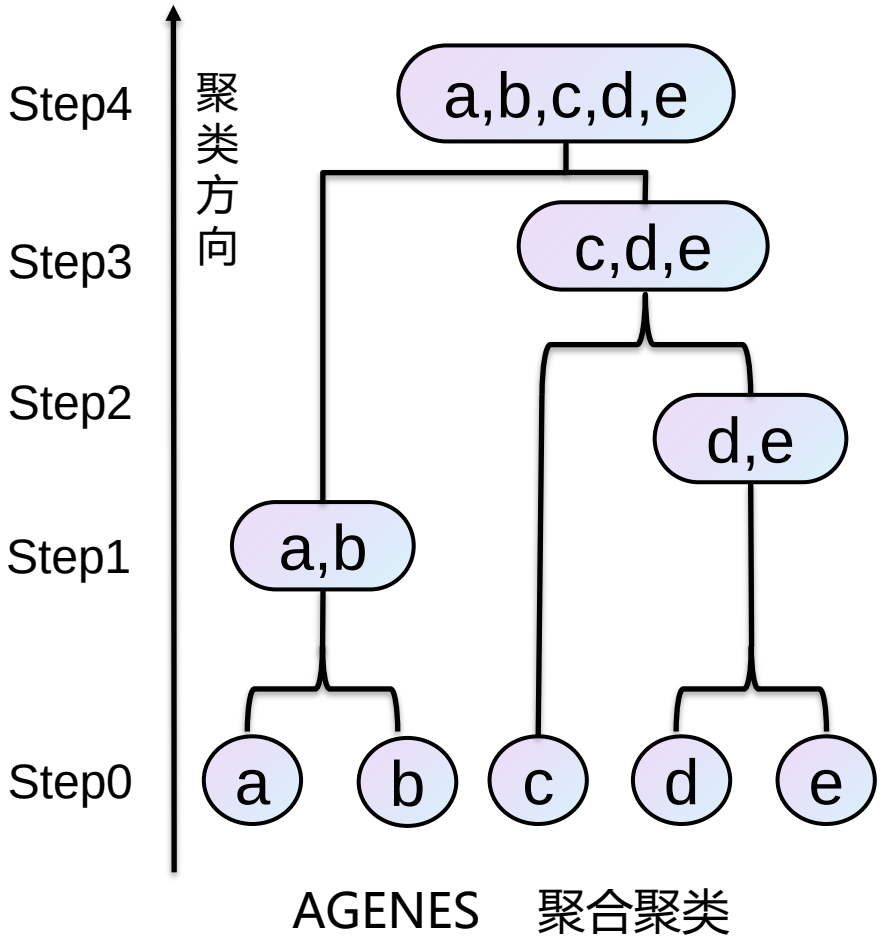
3.密度聚类和层次聚类


AGENES 聚合聚类
Step0
Step1
Step2
Step3
Step4
a
b
a,b
c
d
e
d,e
c,d,e
a,b,c,d,e
Step4
Step3
Step2
Step1
Step0
DIANA 分裂聚类

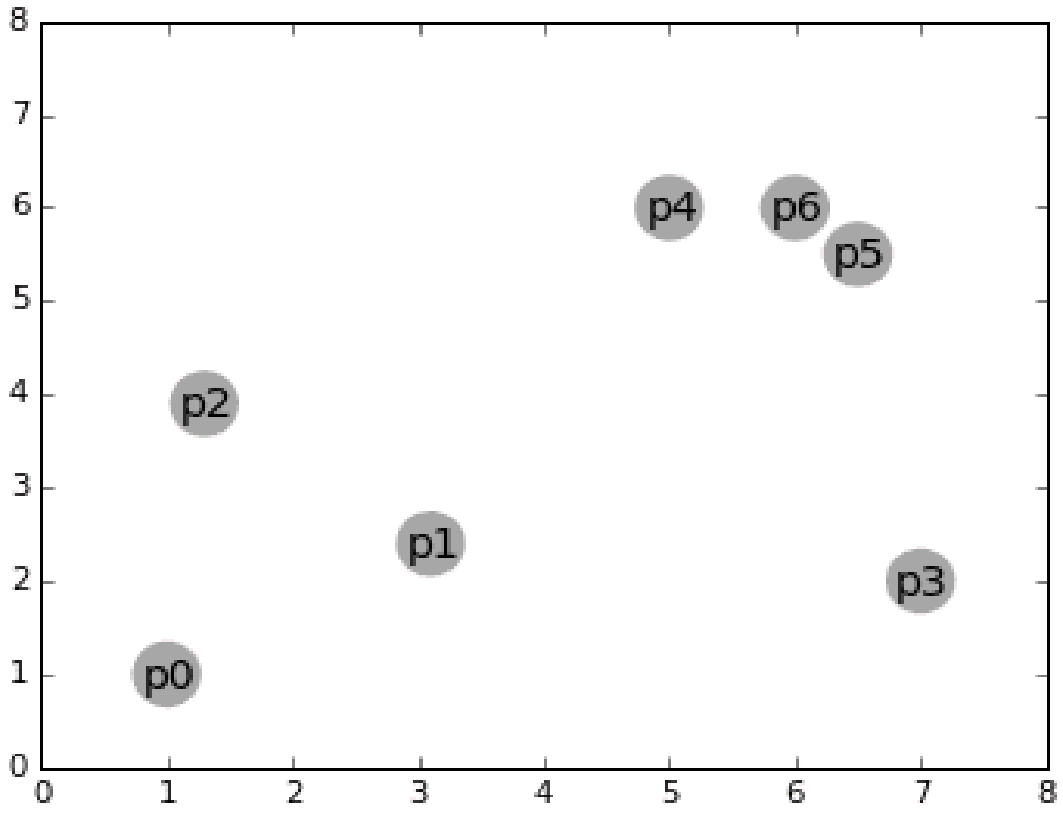
3.密度聚类和层次聚类

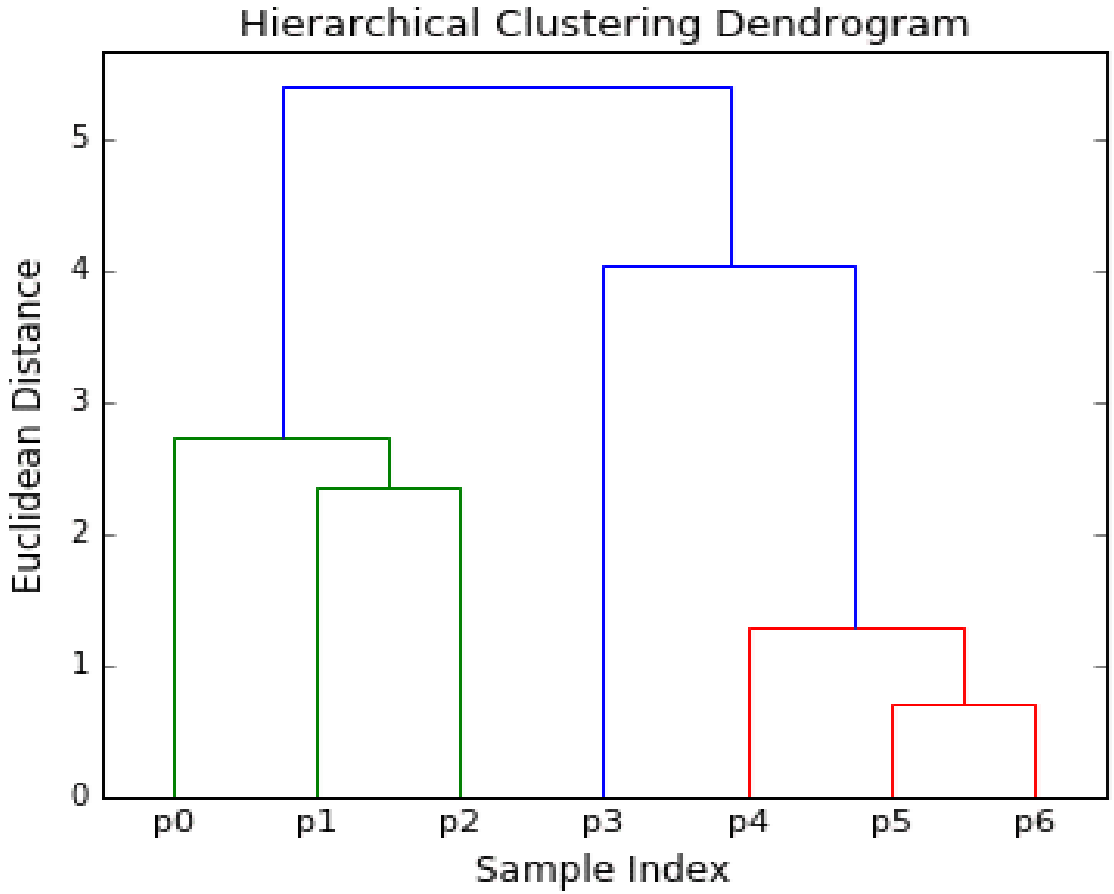
聚合聚类

- 开始将每个样本各自分到一个簇；
- 之后将相距最近的两簇合并，建立一个新的簇；
- 重复此操作直到满足停止条件；
- 得到层次化的类别。



3.密度聚类和层次聚类



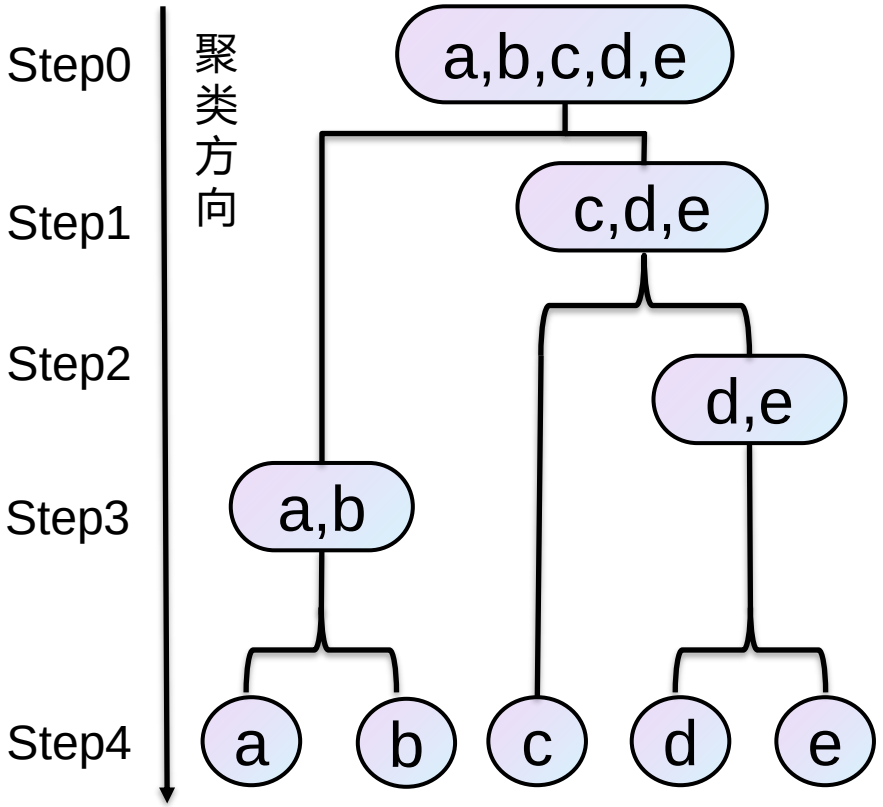


AGENES　聚合聚类

3.密度聚类和层次聚类

分裂聚类

- 开始将所有样本分到一个簇；
- 之后将已有类中相距最远的样本分到两个新的簇；
- 重复此操作直到满足停止条件；
- 得到层次化的类别。



DIANA 分裂聚类

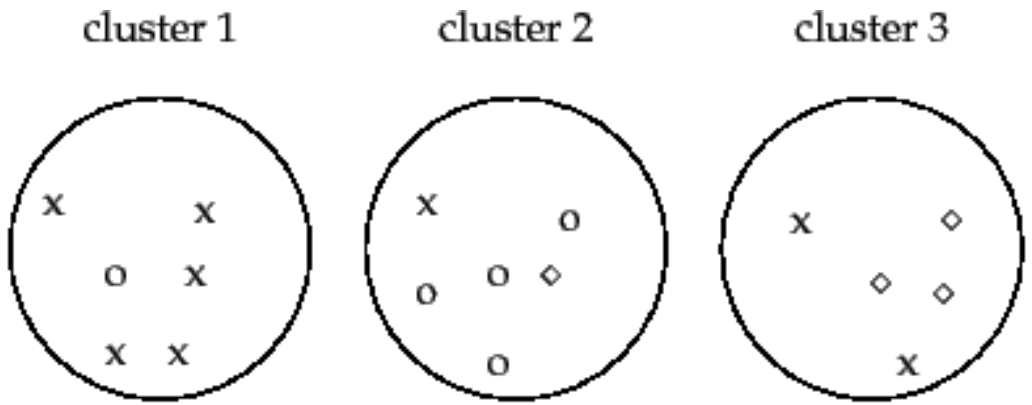
4. 聚类的评价指标

4. 聚类的评价指标

(1). 均一性 purity：p

类似于精确率，一个簇中只包含一个类别的样本，则满足均一性。其实也可以认为就是正确率(每个聚簇中正确分类的样本数占该聚簇总样本数的比例和)

$$p = \frac{1}{k}\sum_{i=1}^{k}\frac{N(C_i == K_i))}{N(K_i)}$$



► **Figure 16.1** Purity as an external evaluation criterion for cluster quality. Majority class and number of members of the majority class for the three clusters are: x, 5 (cluster 1); o, 4 (cluster 2); and ⋄, 3 (cluster 3). Purity is $(1/17) \times (5+4+3) \approx 0.71$.

4. 聚类的评价指标

(2). 兰德系数Rand index [Rand 1971]:

$$RI = \frac{TP + TN}{TP + FP + FN + TN}$$

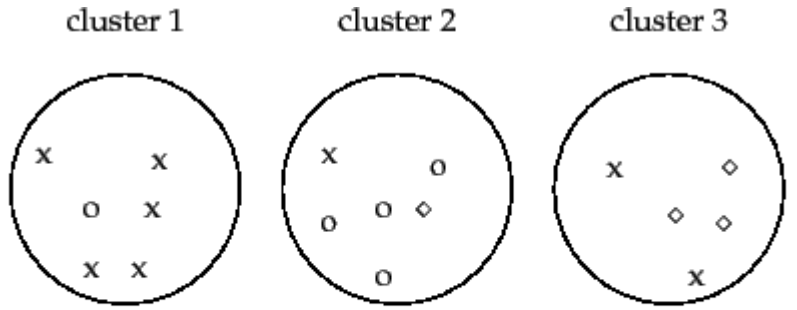
- 俩俩配对，都“对”的组合数：

$$TP + FP = \binom{6}{2} + \binom{6}{2} + \binom{5}{2} = 40$$

- 俩俩配对，实际都对的组合数，包括：
  - cluster 1 中的‘x’
  - cluster 2中的‘o’，
  - cluster 3中的‘◇’和‘x’

$$TP = \binom{5}{2} + \binom{4}{2} + \binom{3}{2} + \binom{2}{2} = 20$$

- 最终得到RI= $(20 + 72)/(20 + 20 + 24 + 72) \approx 0.68$



► **Figure 16.1** Purity as an external evaluation criterion for cluster quality. Majority class and number of members of the majority class for the three clusters are: x, 5 (cluster 1); o, 4 (cluster 2); and ◇, 3 (cluster 3). Purity is $(1/17) \times (5 + 4 + 3) \approx 0.71$.

	Same cluster	Different clusters
Same class	TP = 20	FN = 24
Different classes	FP = 20	TN = 72

4. 聚类的评价指标

(2). F-measure:

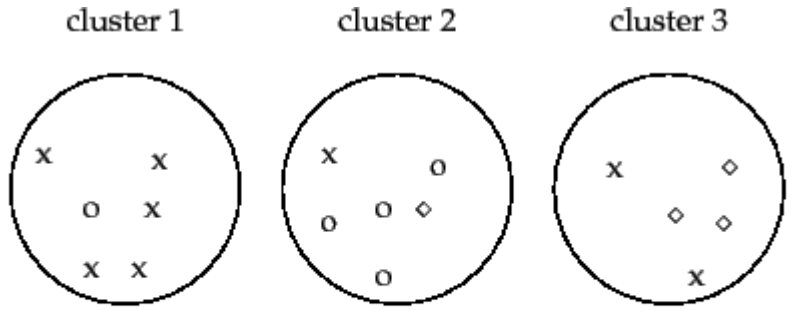
$$P = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}} \qquad R = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}} \qquad F_\beta = \frac{(\beta^2+1)PR}{\beta^2 P + R}$$

当$\beta > 1$时，对 FN 惩罚比FP大：

$$\mathrm{F}_\beta = \frac{1+\beta^2}{\frac{\beta^2}{R}+\frac{1}{P}}$$

上面例子中，$\mathrm{P} = \frac{20}{40} = 0.5, R = 20/44 \approx 0.455$，

- 当$\beta = 1$则 $\mathrm{F}_1 \approx 0.48$；
- 当$\beta = 5$则 $\mathrm{F}_5 \approx 0.456$



► **Figure 16.1** Purity as an external evaluation criterion for cluster quality. Majority class and number of members of the majority class for the three clusters are: x, 5 (cluster 1); o, 4 (cluster 2); and ◇, 3 (cluster 3). Purity is $(1/17) \times (5+4+3) \approx 0.71$.

	Same cluster	Different clusters
Same class	TP = 20	FN = 24
Different classes	FP = 20	TN = 72

4. 聚类的评价指标

(5).调整兰德系数(ARI, Adjusted Rand Index)

[Hubert and Arabie,1985]

数据集S共有N个元素， 两个聚类结果分别是：

$$X = \{X_1, X_2, \ldots, X_r\}, Y = \{Y_1, Y_2, \ldots, Y_s\}$$

X和Y的元素个数为：

$$a = \{a_1, a_2, \ldots, a_r\}, b = \{b_1, b_2, \ldots, b_s\}$$

C	Y_1	Y_2	$\cdots$	Y_s	sum
X_1	n_{11}	n_{12}	$\cdots$	n_{1s}	a_1
X_2	n_{21}	n_{22}	$\cdots$	n_{2s}	a_2
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
X_r	n_{r1}	n_{r2}	$\cdots$	n_{rs}	a_r
sum	b_1	b_2	$\cdots$	b_s	N

$$\overbrace{ARI}^{\text{Adjusted Index}} = \frac{\overbrace{\sum_{ij}\binom{n_{ij}}{2}}^{\text{Index}} - \overbrace{[\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2}]/\binom{n}{2}}^{\text{Expected Index}}}{\underbrace{\frac{1}{2}[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}]}_{\text{Max Index}} - \underbrace{[\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2}]/\binom{n}{2}}_{\text{Expected Index}}}$$

ARI取值范围为[−1,1]，值越大意味着聚类结果与真实情况越吻合。从广义的角度来讲，ARI衡量的是两个数据分布的吻合程度

参考文献

1. 《统计学习方法》，清华大学出版社，李航著，2019年出版
2. 《机器学习》，清华大学出版社，周志华著，2016年出版
3. Christopher M. Bishop, Pattern Recognition and Machine Learning, Springer-Verlag, 2006
4. 《人工智能概论》，北京联合大学，彭涛
5. 《机器学习》，邹伟


谢 谢!