

第八课（第22-24课时）

邻近度和聚类

- 从数据挖掘角度看数据
- 数据点之间的距离
- KNN
- 聚类分析方法及评价

- 分析单个变量：各种方法
- 分析多个变量：各种方法
- 回归分析和广义线性模型：确认变量之间的关系
 - 解释和预测
- 分类分析：预测类别型因变量，有监督学习
- 基于重抽样：
 - 统计量的显著性检验和区间估计（permutation test, Bootstrap）
 - 增强训练效果和评价的稳定性（CV，Bagging，Boost..）
- 模型选择：
 - 拟合度，查准率，查全率，ROC

预测与分类：任务描述

- 理解预测和分类的目的
- 了解各种预测和分类算法
- 掌握如何根据因变量和自变量的类型来确定模型和算法
- 掌握对模型的评价方法
- 理解和了解对数据集的操作

数据挖掘视角的数据

- **数据集**：用属性描述的数据对象的集合
- **属性**：刻画对象基本特征
 - 如：眼睛的颜色、温度
 - 属性=变量、字段、特性, or 特征、维
- **数据对象**：记录，点，案例，样本，事件、实例

	team	coach	play	ball	score	game	win	lost	timeout	season
Document 1	3	0	5	0	2	6	0	2	0	2
Document 2	0	7	0	2	1	0	0	3	0	0
Document 3	0	1	0	0	1	2	2	0	3	0

对象
Objects

属性
Attributes

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

➤ 数据分析：

- 关心变量本身，以及变量之间的关系（列）
- 不符合预先假设模式的：异常点

➤ 数据挖掘：

- 目的为发现模式
- 开始关心数据点（行）
- 甚至关心点：如，某个人是否会喜欢某本书？

➤ 数据的相似性和相异性

- Similarity 相似度：度量对象之间的相似程度
 - $[0, 1]$ ：越大表示越相似
- Dissimilarity 相异度 (距离)：度量对象的差异程度
 - 最小为0，表示两者相同
 - 最大无上限
- Proximity refers to a similarity or dissimilarity
- 邻近性（泛指相似性和相异性）

➤ p and q are the attribute values for two data objects.

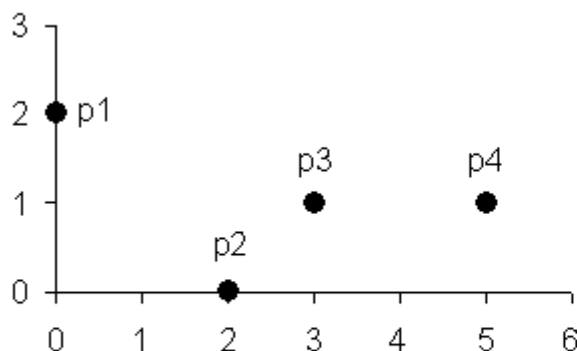
Attribute Type	Dissimilarity	Similarity
Nominal	$d = \begin{cases} 0 & \text{if } p = q \\ 1 & \text{if } p \neq q \end{cases}$	$s = \begin{cases} 1 & \text{if } p = q \\ 0 & \text{if } p \neq q \end{cases}$
Ordinal	$d = \frac{ p-q }{n-1}$ (values mapped to integers 0 to $n-1$, where n is the number of values)	$s = 1 - \frac{ p-q }{n-1}$
Interval or Ratio	$d = p - q $	$s = -d, s = \frac{1}{1+d} \text{ or } s = 1 - \frac{d - \min_d}{\max_d - \min_d}$

Table 5.1. Similarity and dissimilarity for simple attributes

➤ 欧几里得距离

$$dist = \sqrt{\sum_{k=1}^n (p_k - q_k)^2}$$

➤ 一般需要对不同属性进行标准化（规范化）



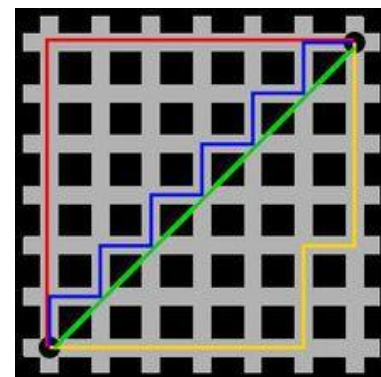
point	x	y
p1	0	2
p2	2	0
p3	3	1
p4	5	1

	p1	p2	p3	p4
p1	0	2.828	3.162	5.099
p2	2.828	0	1.414	3.162
p3	3.162	1.414	0	2
p4	5.099	3.162	2	0

➤ 闵可夫斯基距离

$$dist = \left(\sum_{k=1}^n |p_k - q_k|^r \right)^{\frac{1}{r}}$$

- $r = 1$. City block (曼哈顿, L_1 norm) 距离.
- $r = 2$ 欧几里得距离
- $r \rightarrow \infty$ “supremum” (L_{\max} norm, L_{∞} norm) 距离
 - 两个向量间的最大距离



数据的邻近度

point	x	y
p1	0	2
p2	2	0
p3	3	1
p4	5	1

L1	p1	p2	p3	p4
p1	0	4	4	6
p2	4	0	2	4
p3	4	2	0	2
p4	6	4	2	0

L2	p1	p2	p3	p4
p1	0	2.828	3.162	5.099
p2	2.828	0	1.414	3.162
p3	3.162	1.414	0	2
p4	5.099	3.162	2	0

L_{∞}	p1	p2	p3	p4
p1	0	2	3	5
p2	2	0	1	3
p3	3	1	0	2
p4	5	3	2	0

➤ Binary向量

– 对象 p , q 只有0,1属性

- M_{01} = p 为0 且 q 为 1的属性个数
- M_{10} = p 为1 且 q 为 0的属性个数
- M_{00} = p 为0 且 q 为 0的属性个数
- M_{11} = p 为1 且 q 为 1的属性个数

– Simple Matching Coefficients

- $(M_{11} + M_{00}) / (M_{01} + M_{10} + M_{11} + M_{00})$

– Jaccard Coefficients

- $(M_{11}) / (M_{01} + M_{10} + M_{11})$ 处理非对称的二元属性（稀疏数据）

➤ SMC v.s. Jaccard:

– $p = 1000000000$

– $q = 0000001001$

- $M01 = 2$

- $M10 = 1$

- $M00 = 7$

- $M11 = 0$

– $SMC = (M11 + M00) / (M01 + M10 + M11 + M00) = (0 + 7) / (2 + 1 + 0 + 7) = 0.7$

– $J = (M11) / (M01 + M10 + M11) = 0 / (2 + 1 + 0) = 0$

数据的邻近度

➤ 余弦相似性

—如果 d_1 , d_2 为两个文档向量

$$\cos(d_1, d_2) = (d_1 \cdot d_2) / \|d_1\| \|d_2\|$$

$$d_1 = 3 \ 2 \ 0 \ 5 \ 0 \ 0 \ 0 \ 2 \ 0 \ 0$$

$$d_2 = 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 2$$

	team	coach	play	ball	score	game	win	lost	timeout	season
Document 1	3	0	5	0	2	6	0	2	0	2
Document 2	0	7	0	2	1	0	0	3	0	0
Document 3	0	1	0	0	1	2	2	0	3	0

$$d_1 \cdot d_2 = 3*1 + 2*0 + 0*0 + 5*0 + 0*0 + 0*0 + 0*0 + 2*1 + 0*0 + 0*2 = 5$$

$$\|d_1\| = (3*3 + 2*2 + 0*0 + 5*5 + 0*0 + 0*0 + 0*0 + 2*2 + 0*0 + 0*0)^{0.5} = (42)^{0.5} = 6.481$$

$$\|d_2\| = (1*1 + 0*0 + 0*0 + 0*0 + 0*0 + 0*0 + 0*0 + 1*1 + 0*0 + 2*2)^{0.5} = (6)^{0.5} = 2.245$$

$$\cos(d_1, d_2) = .3150$$

➤ 广义Jaccard系数：用于文档数据

$$T(p, q) = \frac{p \bullet q}{\|p\|^2 + \|q\|^2 - p \bullet q}$$

➤ 相关性 Correlation

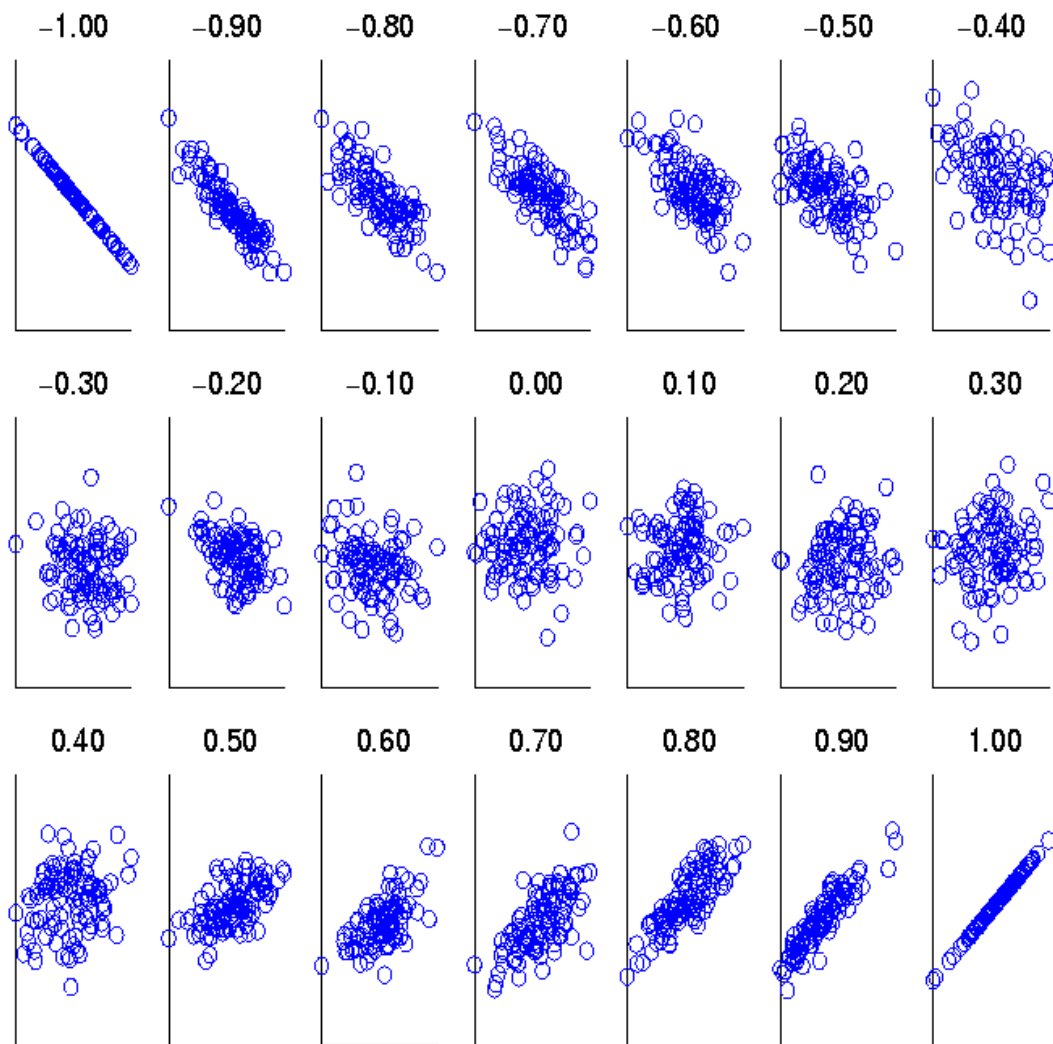
- 度量线性关系
- 标准化后计算

$$p'_k = (p_k - \text{mean}(p)) / \text{std}(p)$$

$$q'_k = (q_k - \text{mean}(q)) / \text{std}(q)$$

$$\text{correlation}(p, q) = p' \bullet q'$$

➤ 相关性

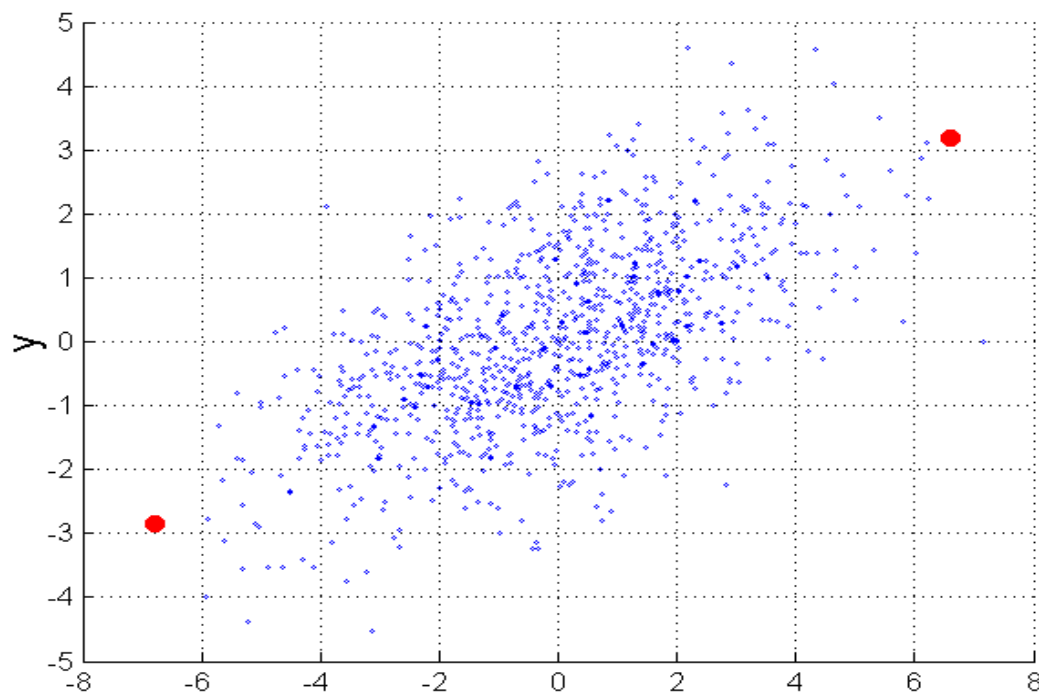


➤ Mahalanobis 距离

– scale不同或相关时

$$mahalanobis(p, q) = (p - q) \Sigma^{-1} (p - q)^T$$

Σ 为数据的协方差矩阵



x,y 相关系数为0.6. 欧几里得距离: 14.7, Mahalanobis distance: 6.

➤ 组合异种属性的邻近度

- 属性类型多样，但是需要一个总体的邻近度量

$$similarity(p, q) = \frac{\sum_{k=1}^n \delta_k s_k}{\sum_{k=1}^n \delta_k}$$

- s_k : 第k个属性度量的邻近度
- δ_k :
 - 0, 如果第k个属性为不均衡的二元属性, 而p,q都为0
 - 0, p或q的第k个属性缺失
 - 1 其他情况

➤ 属性权重不同

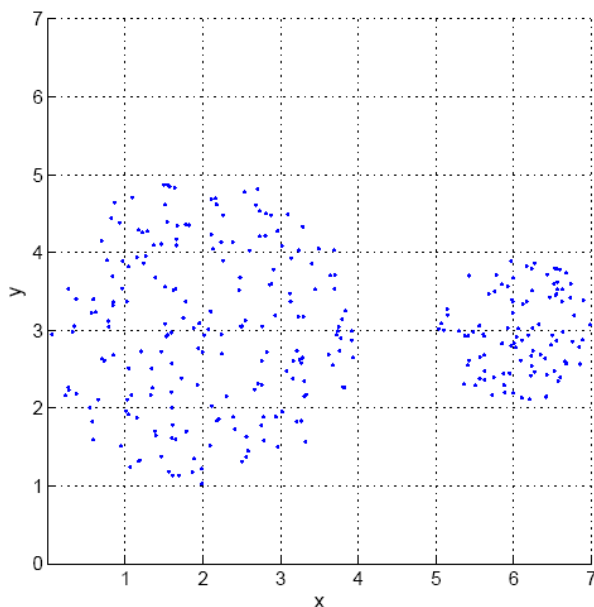
$$\text{similarity}(p, q) = \frac{\sum_{k=1}^n w_k \delta_k s_k}{\sum_{k=1}^n \delta_k}$$

$$\text{distance}(p, q) = \left(\sum_{k=1}^n w_k |p_k - q_k|^r \right)^{1/r}$$

➤ 密度

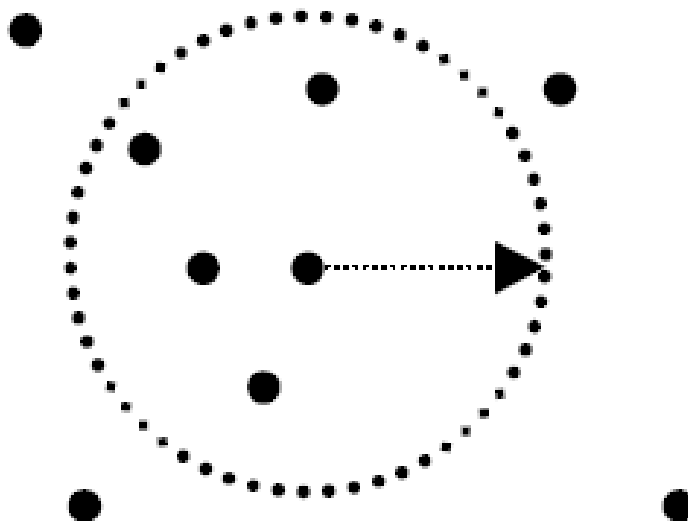
- 欧几里得密度
- 概率密度
- 基于图的密度

➤ 欧几里得密度：基于Cell



0	0	0	0	0	0	0
0	0	0	0	0	0	0
4	17	18	6	0	0	0
14	14	13	13	0	18	27
11	18	10	21	0	24	31
3	20	14	4	0	0	0
0	0	0	0	0	0	0

➤ 欧几里得密度：基于中心划分



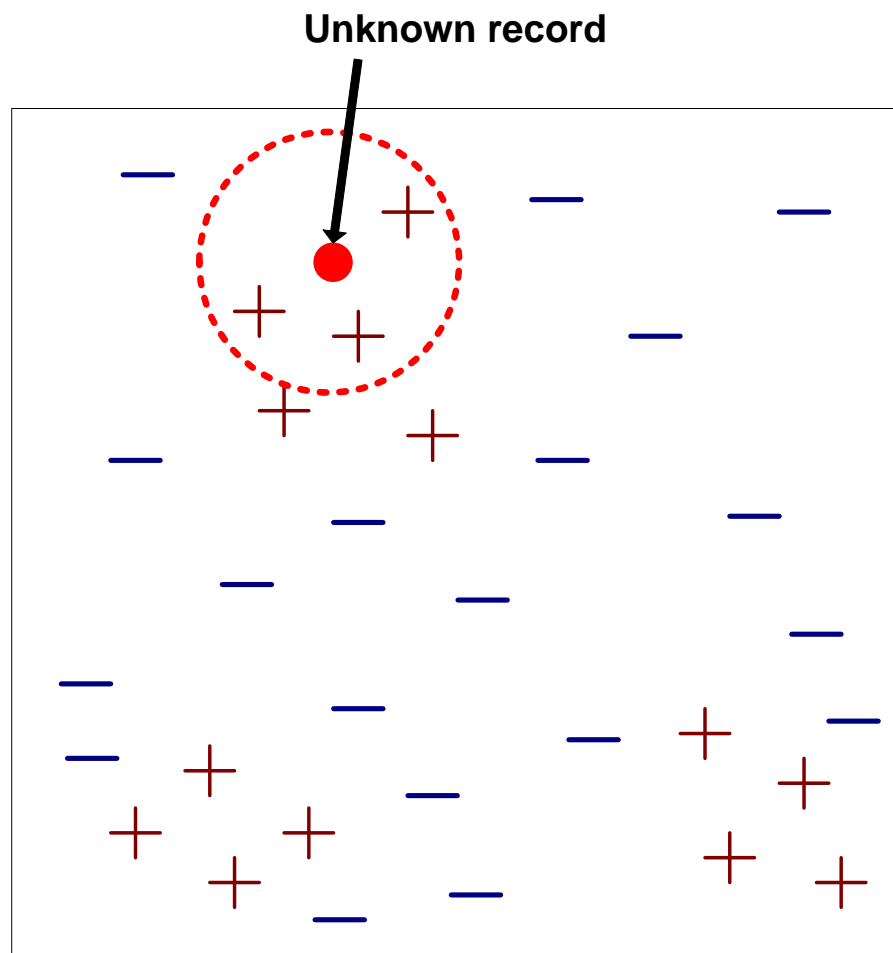
数据的邻近性

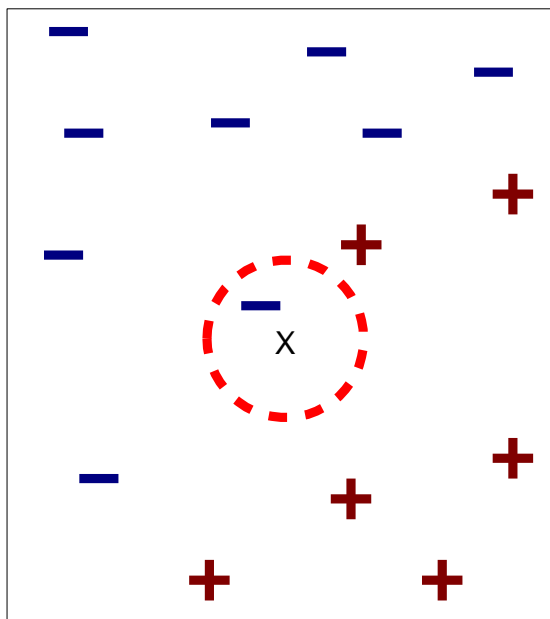
➤ 最微妙的部分

- 现实世界→数据空间
- 数学建模问题
- 基于业务

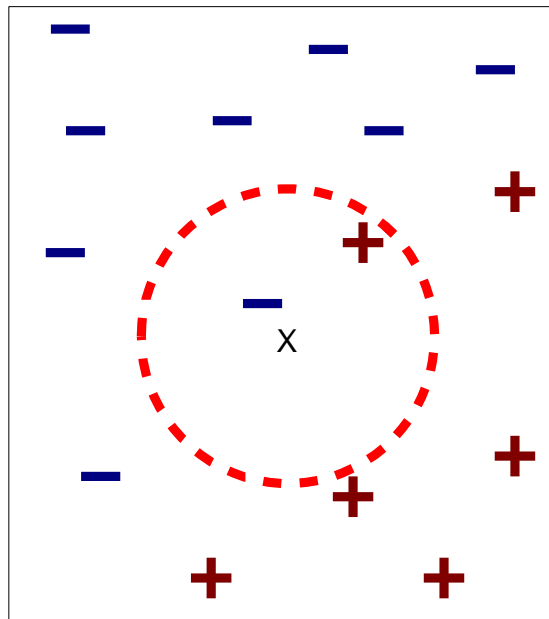
➤ KNN-最邻近分类

- 定义数据集中记录之间的距离
- 定义参数 k ：临近数
- 分类时识别最近的 k 个紧邻
- 由 k 个近邻的类别对未知记录进行投票

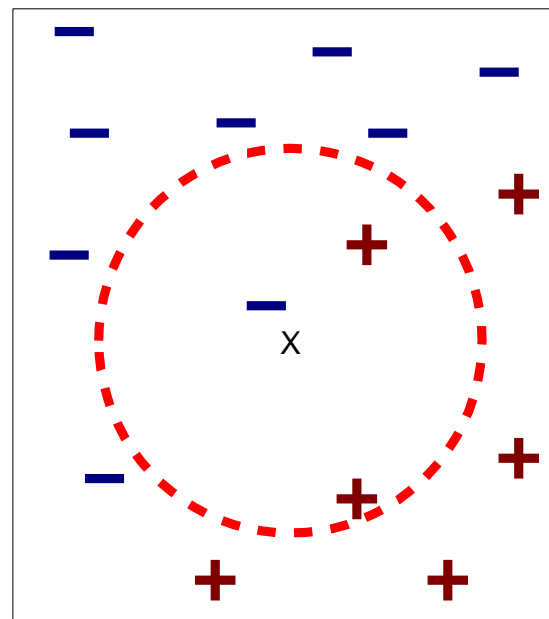




(a) 1-nearest neighbor

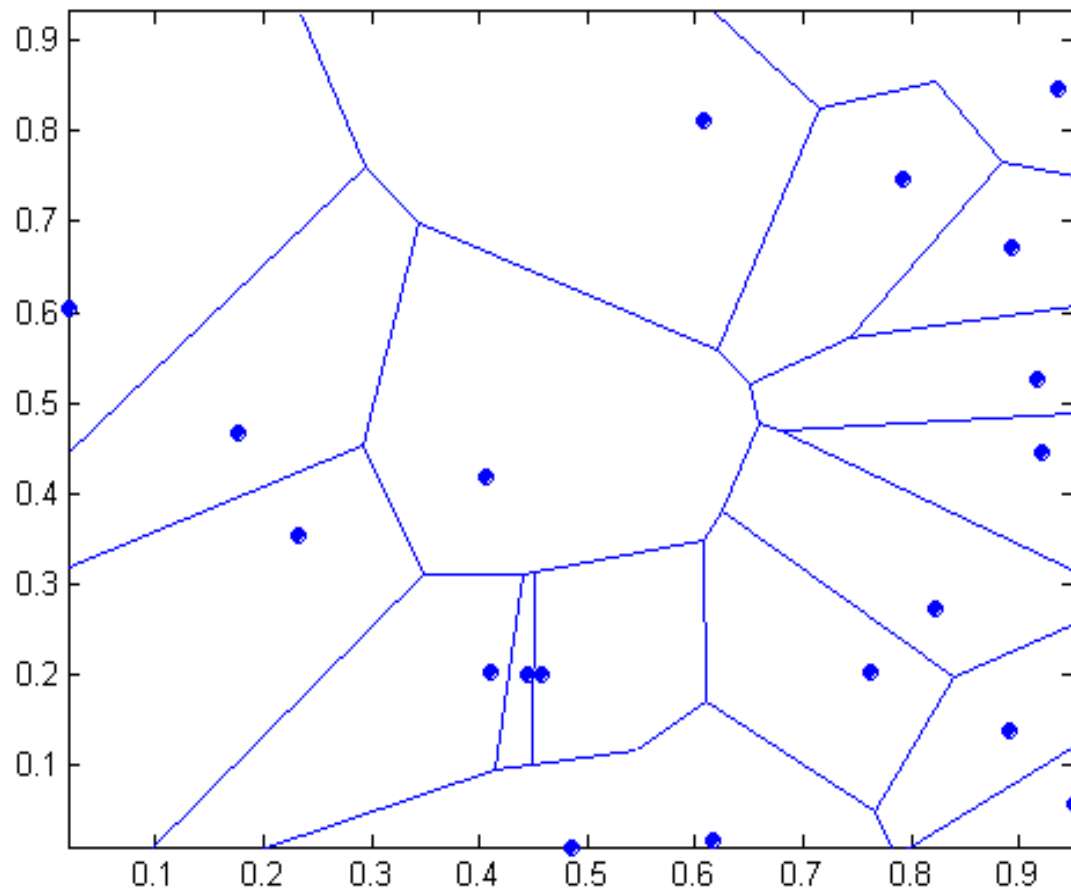


(b) 2-nearest neighbor



(c) 3-nearest neighbor

➤ 1-近邻 Voronoi Diagram



➤ 过程

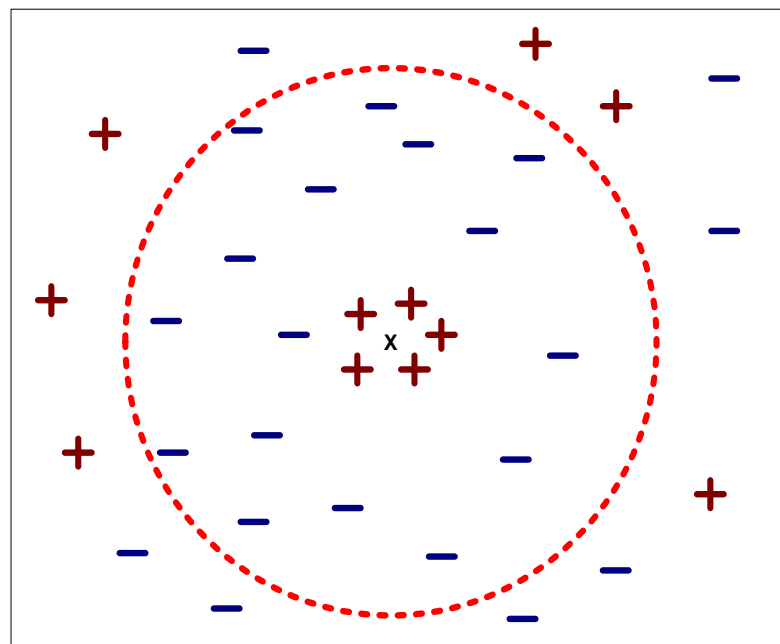
- K个近邻就类别投票
- 用距离作为权重： $w = 1/d^2$ 用权重

➤ k值的选择

- 过小：受噪声影响
- 过大：受非同类数据影响

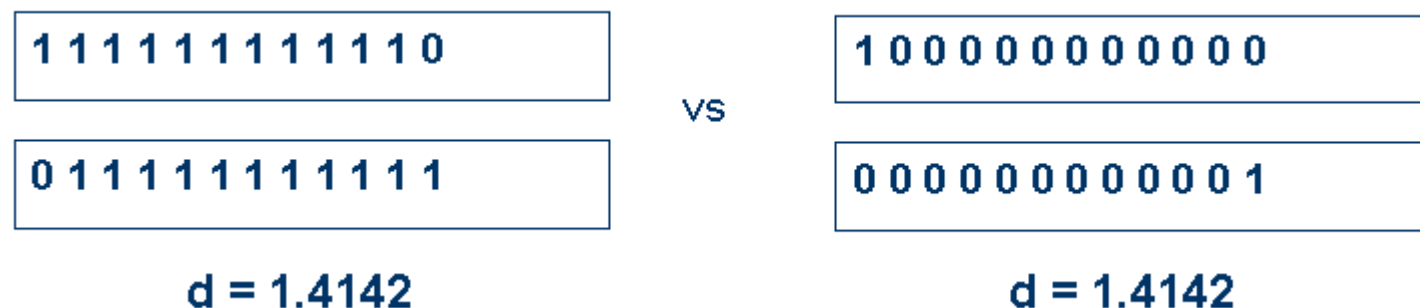
➤ 数据属性的尺度问题

- 归一化
 - 例如，身高 1.5m ~ 1.8m
 - 人的体重 90lb ~ 300lb
 - 收入：\$10K ~ \$1M



➤ 欧几里德距离的问题

- 高维诅咒：不易计算
- 可能产生问题：



- 如何解决？

➤ KNN的问题：并不能建立明确的模型，较高的消耗资源

➤ PEBLS: Parallel Exemplar-Based Learning System (Cost & Salzberg)

- 属性可以是数值型或类别型 (使用value difference metric (MVDM)) ; 每个属性赋予权重 ; $k=1$

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Class	Marital Status		
	Single	Married	Divorced
Yes	2	0	1
No	2	4	1

Class	Refund	
	Yes	No
Yes	0	3
No	3	4

$$d(\text{Single}, \text{Married}) = |2/4 - 0/4| + |2/4 - 4/4| = 1$$

$$d(\text{Single}, \text{Divorced}) = |2/4 - 1/2| + |2/4 - 1/2| = 0$$

$$d(\text{Married}, \text{Divorced}) = |0/4 - 1/2| + |4/4 - 1/2| = 1$$

$$d(\text{Refund}=\text{Yes}, \text{Refund}=\text{No}) = |0/3 - 3/7| + |3/3 - 4/7| = 6/7$$

$$d(V_1, V_2) = \sum_i \left| \frac{n_{1i}}{n_1} - \frac{n_{2i}}{n_2} \right|$$

➤ PEBLS

<i>Tid</i>	Refund	Marital Status	Taxable Income	Cheat
X	Yes	Single	125K	No
Y	No	Married	100K	No

$$\Delta(X, Y) = w_X w_Y \sum_{i=1}^d d(X_i, Y_i)^2$$

$$w_X = \frac{\text{Number of times X is used for prediction}}{\text{Number of times X predicts correctly}}$$

➤ python实现

- sklearn.neighbors.KNeighborsClassifier

Examples

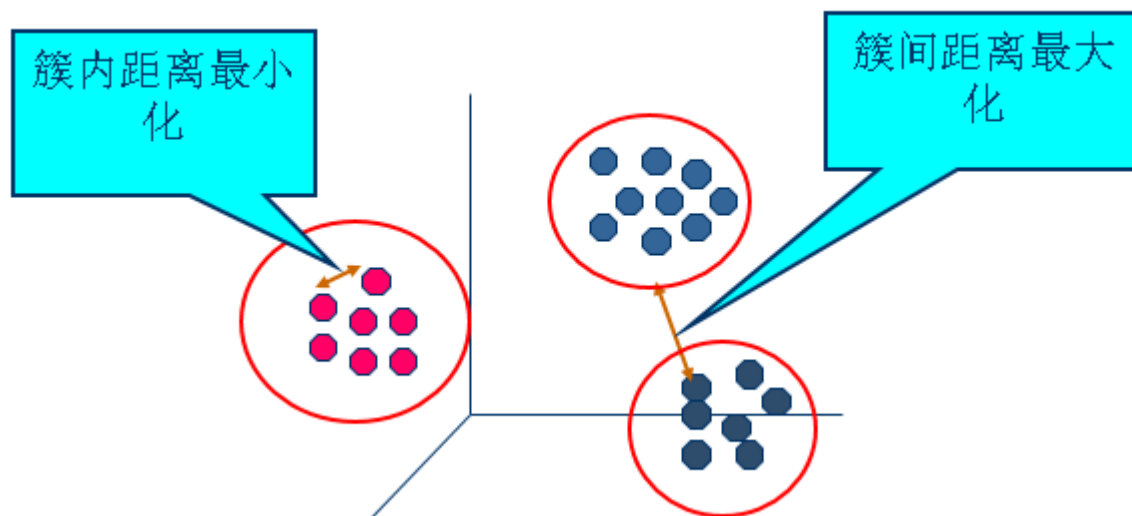
```
>>> X = [[0], [1], [2], [3]]
>>> y = [0, 0, 1, 1]
>>> from sklearn.neighbors import KNeighborsClassifier
>>> neigh = KNeighborsClassifier(n_neighbors=3)
>>> neigh.fit(X, y)
KNeighborsClassifier(...)
>>> print(neigh.predict([[1.1]]))
[0]
>>> print(neigh.predict_proba([[0.9]]))
[[ 0.66666667  0.33333333]]
```

- 更多说明和例子参见官方文档



聚类分析：Cluster Analysis 定义

- 将数据划分成有意义或有用的组（簇）
 - 捕获数据的自然结构（旨在理解）
 - 解决其他问题的起点（旨在实用）
- 根据在数据中发现的描述对象及其关系的信息，将数据分组
 - 目标：组内相似性（同质性、相关性）越大、组间的越小、聚类就越好



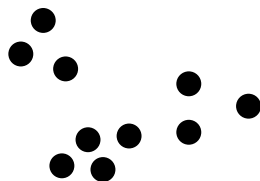
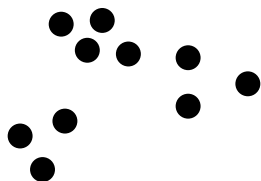
➤ 旨在理解

- 将对象划分成组或类（聚类）
- 指派特定对象到组或类（分类）
- **簇：潜在类**

- 生物学
- 信息检索
- 气候
- 心理学和医学
- 商业

➤ 旨在实用

- 个别数据对象到簇的抽象
- 刻画簇特征的**簇原型**（代表簇中其他对象的数据对象）

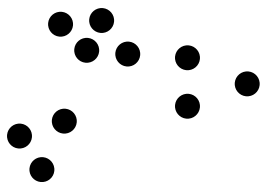


多少个簇？

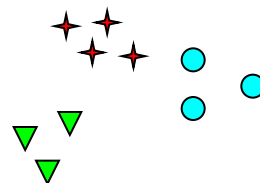
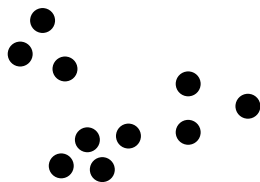
- 汇总：降维
- 压缩：向量量化
- 有效的发现近邻

聚类分析

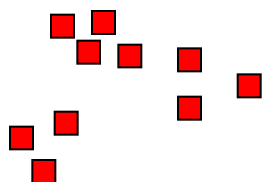
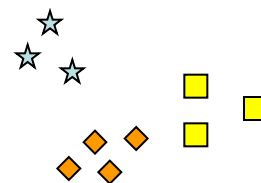
➤ 簇



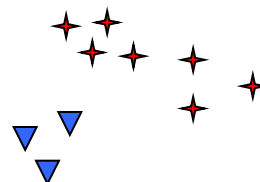
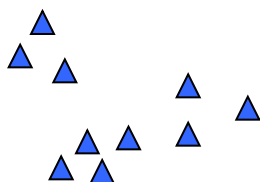
多少个簇?



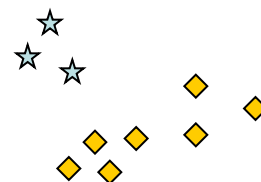
6个簇



2个簇



4个簇



➤ 聚类方式

- 层次的 **hierarchical** 与 划分的 **partitional**

➤ 划分的聚类 **Partitional Clustering**

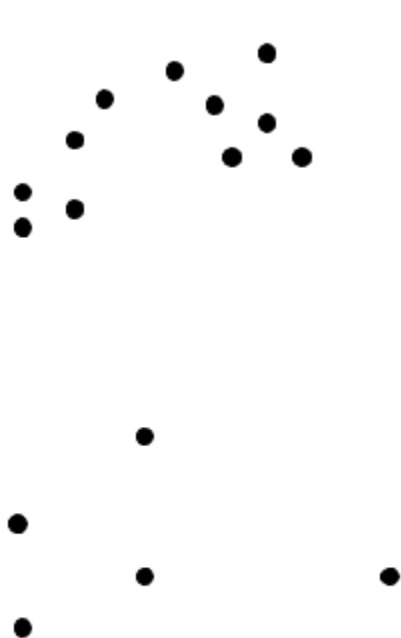
- 数据对象被划入不重叠的子集 (簇, clusters), 使得每个数据对象仅属于一个子集

➤ 层次的聚类 **Hierarchical clustering**

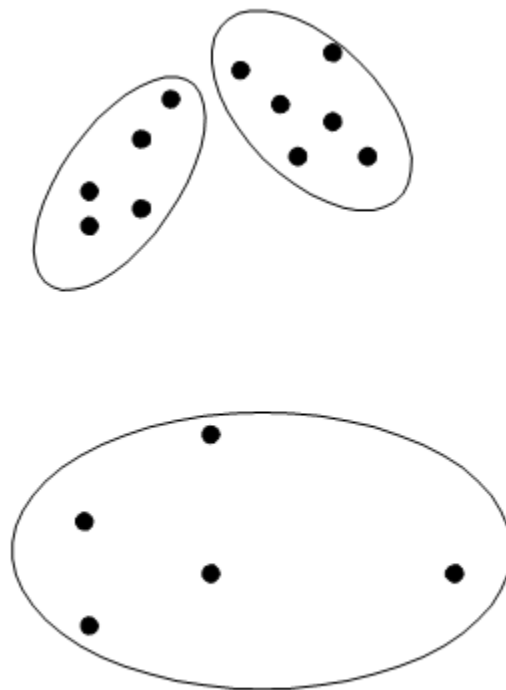
- 簇有子簇, 簇之间存在树形的层次嵌套关系



➤ 划分的聚类 Partitional Clustering

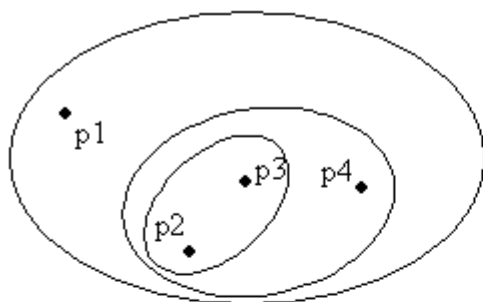


初始数据点

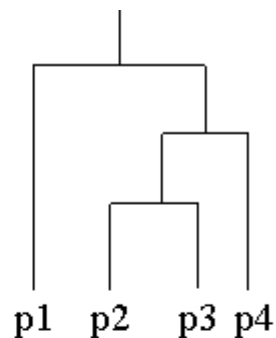


A Partitional Clustering

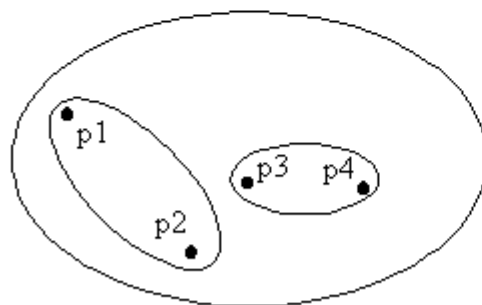
➤ 层次聚类 Hierarchical Clustering



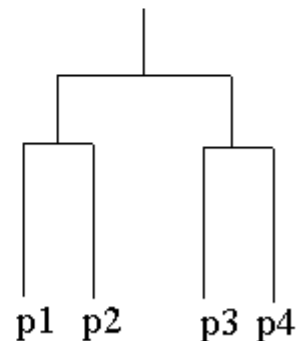
Traditional Hierarchical Clustering



Traditional Dendrogram



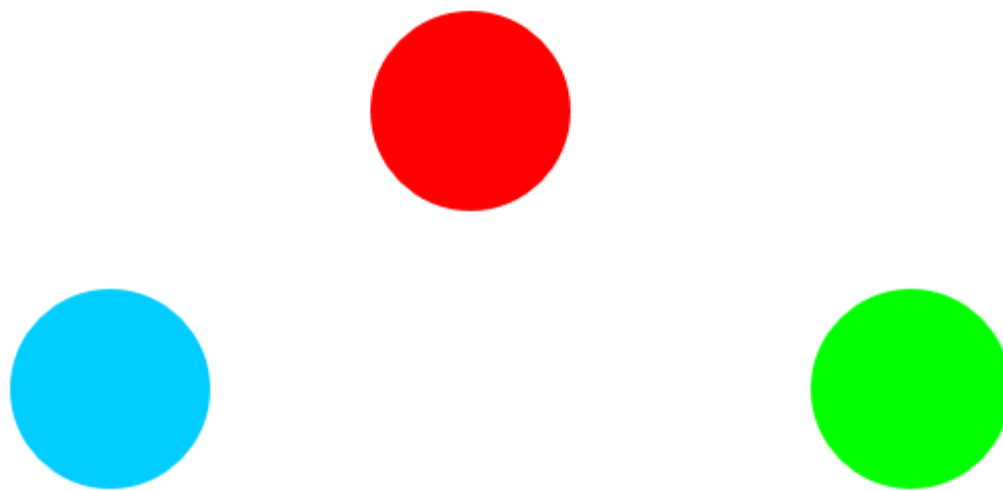
Non-traditional Hierarchical Clustering



Non-traditional Dendrogram

➤ 簇的类型

- 明显分离的 Well-separated clusters
- 基于原型（中心）的 Center-based clusters
- 基于临近的 Contiguous clusters
- 基于密度的 Density-based clusters
- 共同性质的（概念簇） Property or Conceptual



3 well-separated clusters

- 簇的类型: 基于中心 (原型) 的
 - 质心 centroid : 平均点
 - 中心点 medoid : 最有代表性



4 center-based clusters

➤ 簇的类型: 基于临近的

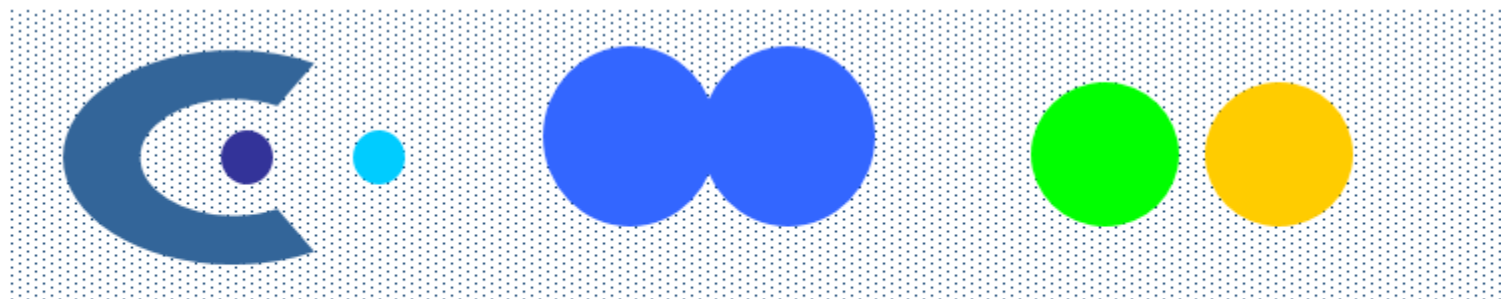
- 簇中的每个对象到该簇的**某个对象**的距离比到不同簇的任意点更近



8 contiguous clusters

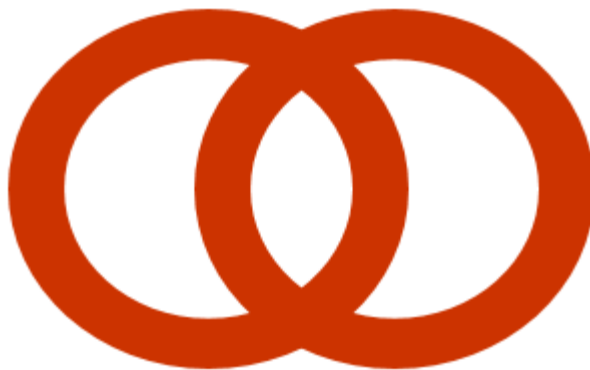
➤ 簇的类型: 基于密度的

- 簇是对象的稠密区域
- 适用于簇形状不规则、或有噪声或离群点时



6 density-based clusters

➤ 簇的类型: 共同性质的 (概念簇)



2 Overlapping Circles

➤ 聚类算法

- K均值 K-means and its variants
 - 基于原型和划分的
- （凝聚的）层次聚类 Hierarchical clustering
- 基于密度的聚类（DBSCAN）Density-based clustering

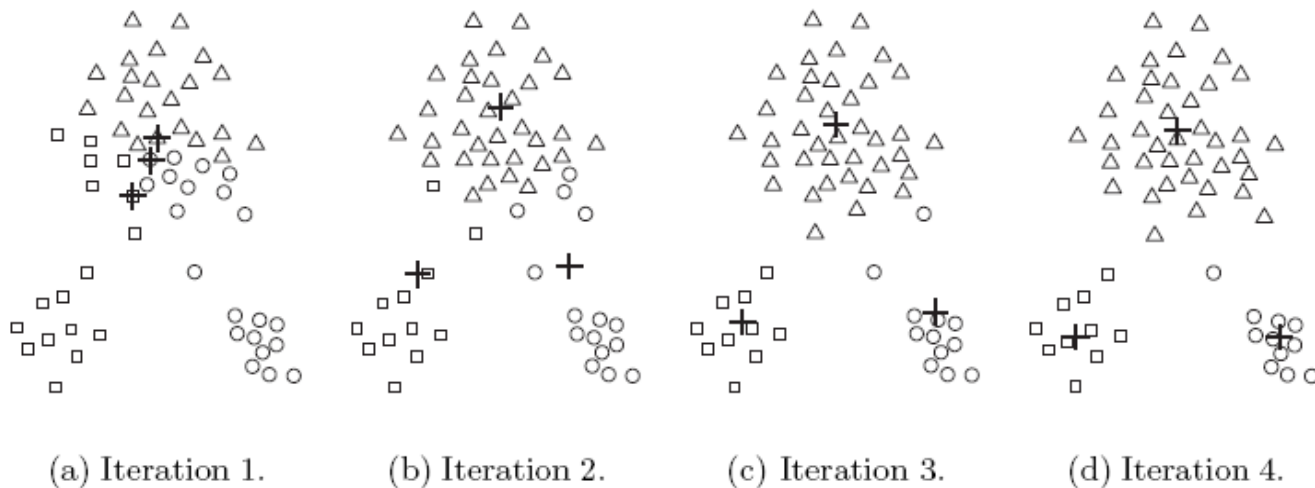
➤ K均值

- 基于划分
- 每个簇有一个质心 **centroid** (center point)
- 每个点被分配给最近的质心
- 需指定簇的数量 K
- 另： K 中心点聚类（中心点必须是一个实际数据点）

-
- 1: Select K points as the initial centroids.
 - 2: **repeat**
 - 3: Form K clusters by assigning all points to the closest centroid.
 - 4: Recompute the centroid of each cluster.
 - 5: **until** The centroids don't change
-

聚类分析：K均值

- 初始质心通常随机设置
- 质心 (通常) 是簇内所有点的均值.
- ‘最近’：欧几里得距离、余弦相似性、相关性、等等.
- 对于某些临近性函数和质心类型，K均值总是快速收敛到某个状态：
 - 质心不变/所有点不会改变簇的归属
 - 较弱的条件，如“直到只有1%的点发生改变”



➤ 1 指派点到最近的质心

– “最近”：邻近性度量

- 欧式空间的点：欧几里得距离（L2），曼哈顿距离(L1)
- 文档：余弦相似性，Jaccard度量

➤ 2 质心和目标函数

– “重新计算每个簇的质心”

- 质心可能随邻近性度量和聚类目标不同而改变

– 欧几里得空间：误差平方和

$$SSE = \sum_{i=1}^K \sum_{x \in C_i} dist^2(c_i, x)$$

– 好的簇：K小，SSE小

➤ 2 质心和目标函数

- 文档数据
- 最大化簇中文档与簇的质心的相似性
- 簇的凝聚度 (cohesion)

$$TotalCohesion = \sum_{i=1}^K \sum_{x \in C_i} \cos ine(x, c_i)$$

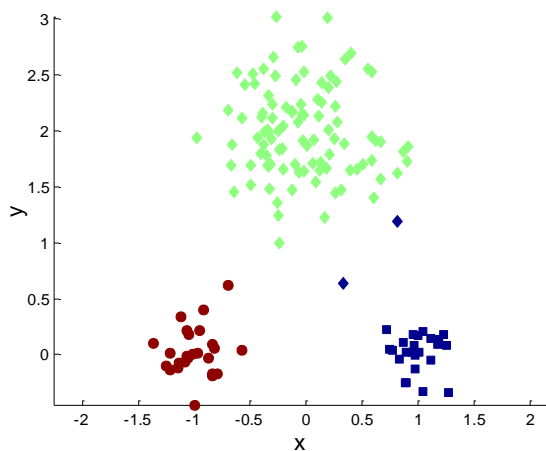
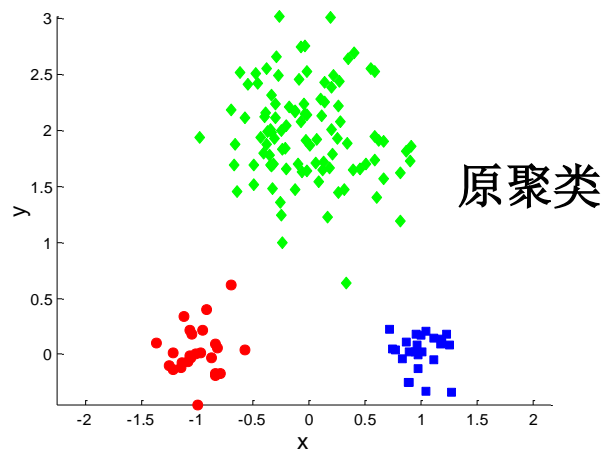
- 曼哈顿距离 L1绝对误差和 SAE

$$SAE = \sum_{i=1}^K \sum_{x \in C_i} dist_{L_1}(c_i, x)$$

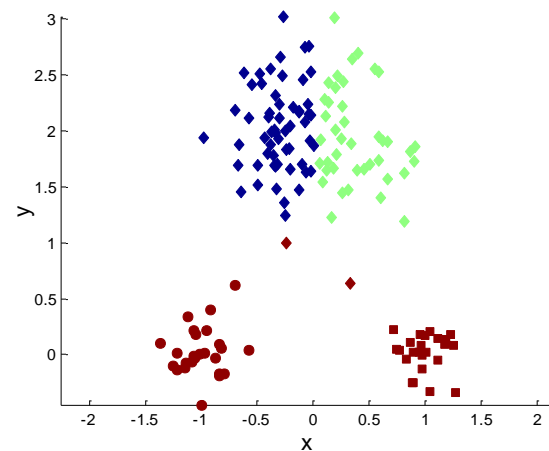
$$dist_{L_1} = |c_i - x|$$

聚类分析：K均值算法

➤ 初始质心的选择

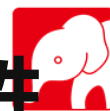


最优聚类

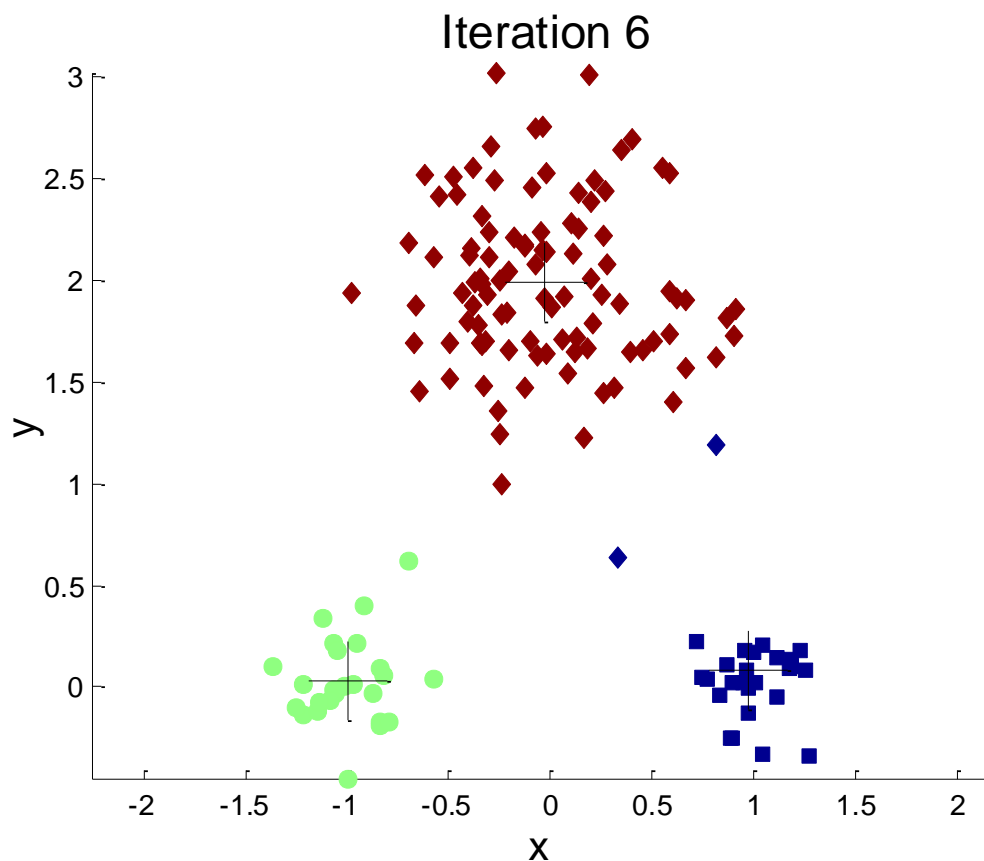


次最优聚类

聚类：K均值，初始质心的选择的重要性



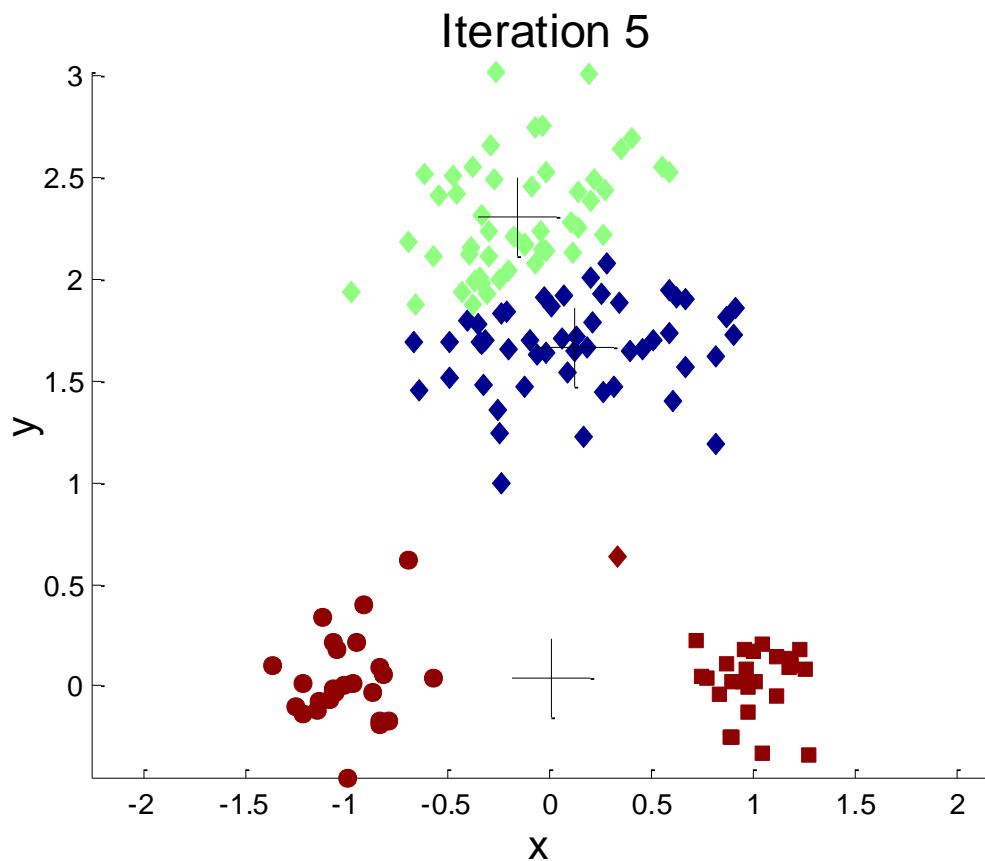
小象学院
ChinaHadoop.cn



聚类：K均值，初始质心的选择的重要性



小象学院
ChinaHadoop.cn



➤ 随机初始化的局限

- 在每个“真正”的簇中选择一个初始点的几率很小
 - 特别是当K很大时
 - 假设簇的大小相同为 n

$$P = \frac{\text{number of ways to select one centroid from each cluster}}{\text{number of ways to select } K \text{ centroids}} = \frac{K!n^K}{(Kn)^K} = \frac{K!}{K^K}$$

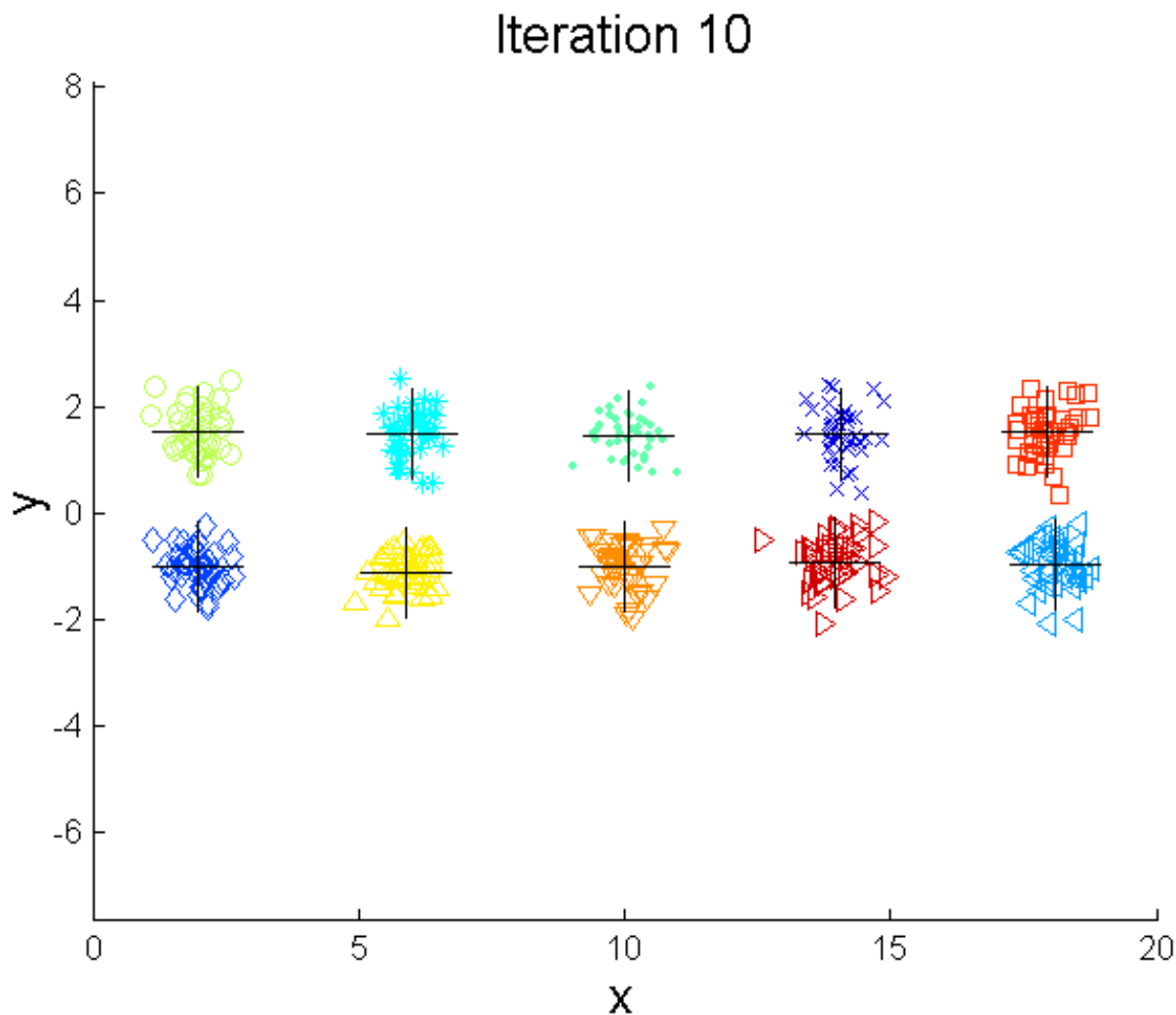
➤ 初始质心问题的解决方案

- 重复运行
 - 帮助有限
- 对样本进行层次聚类技术，提取K个簇的质心作为初始质心
- 后处理 Postprocessing
- 二分K均值 Bisecting K-means
 - 对初始化问题不敏感

➤ 处理空簇

- 简单 K均值算法可能产生空簇（没有点被指派到某个簇），需要选择替补质心
- 解决方法
 - 选择一个距离当前任何质心最远的点
 - 从最大 SSE的簇中选择一个点

聚类：K均值，二分 K均值



➤ 预处理

- 标准化数据
- 离群点可能过度影响所发现的簇
 - 导致原型不够有代表性，SSE较高；可以删除
 - 但是，离群点有可能有趣：数据压缩，异常点
- 识别离群点后聚类
 - 可能需删除很小的簇（常常代表离群点的组）

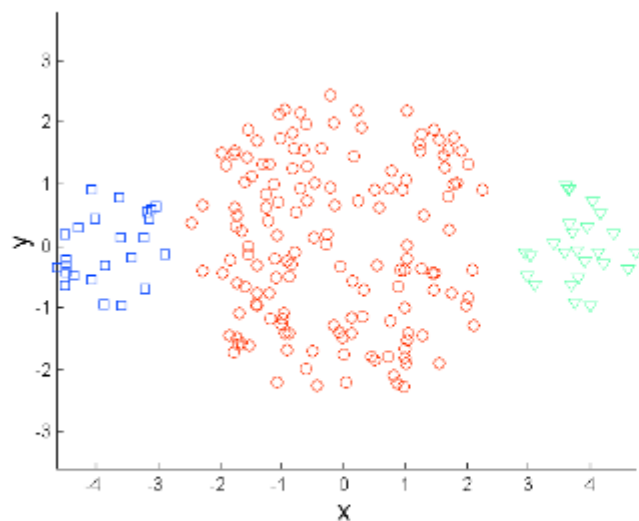
➤ 后处理

- 删除小的簇（可能是离群点的组）
- 分裂簇（高SSE的）
- 合并簇（相近且低SSE）

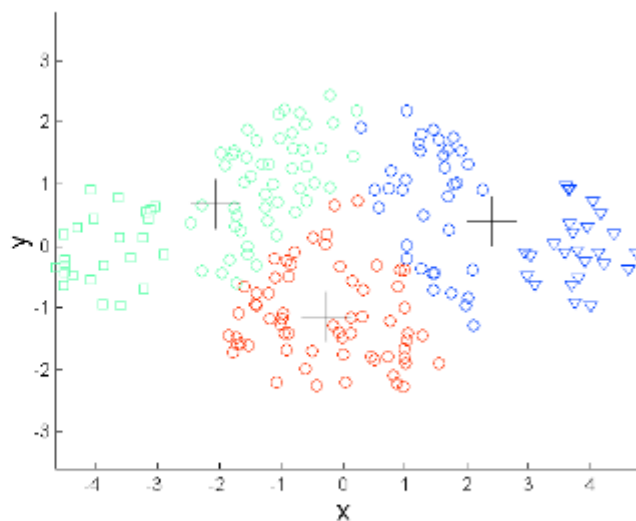
聚类分析：K均值算法

➤ 局限性

- 尺寸
- 密度
- 非球状



Original Points

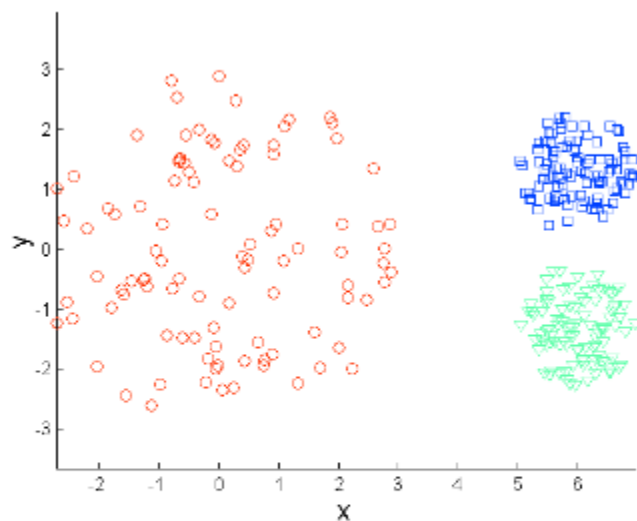


K-means (3 Clusters)

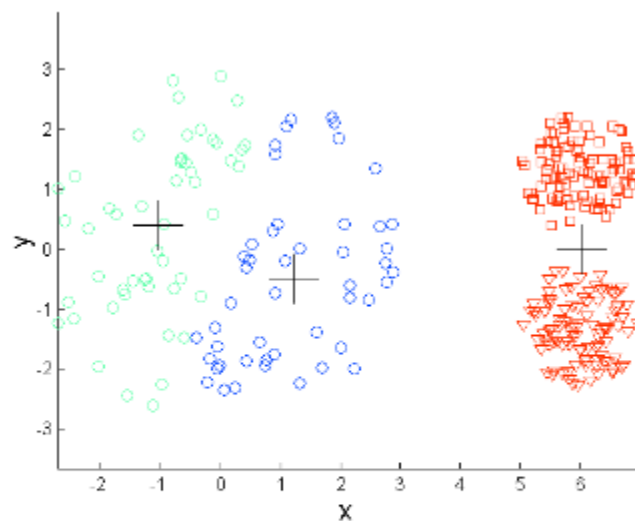
聚类分析：K均值算法

➤ 局限性

- 尺寸
- 密度
- 非球状



Original Points

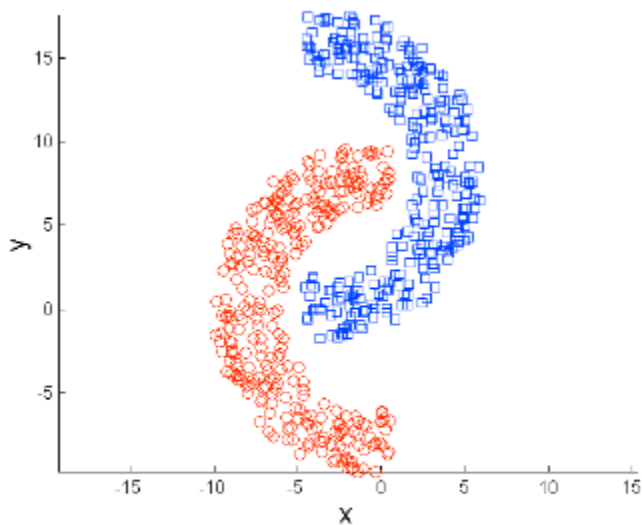


K-means (3 Clusters)

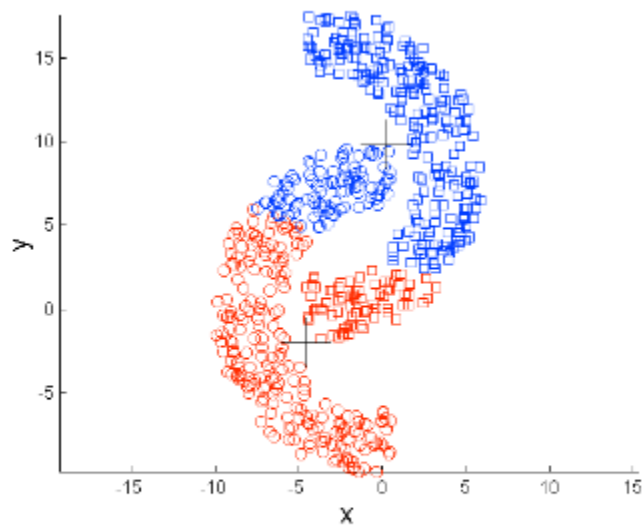
聚类分析：K均值算法

➤ 局限性

- 尺寸
- 密度
- 非球状



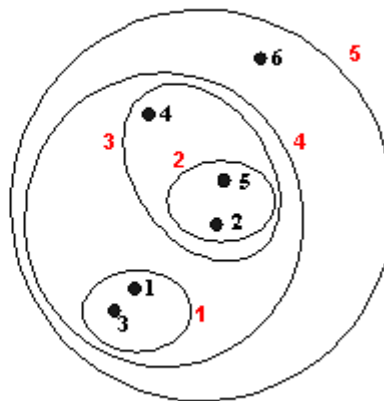
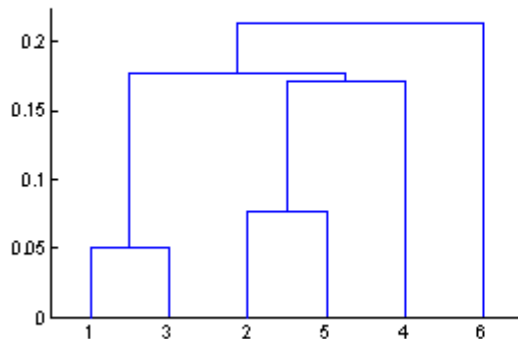
Original Points



K-means (2 Clusters)

聚类分析：层次聚类 Hierarchical

- 生成具有树状层次结构的嵌套簇
- 可被表现为树状图 dendrogram
 - A tree like diagram that records the sequences of merges or splits 也可表现为嵌套簇图



➤ 优点

- 无需预判簇的个数
- 与有实际意义的分类学对应

➤ 两种方法

- 凝聚的 Agglomerative:
 - 从点作为个体簇开始
 - 每一步合并两个最接近的簇，直到一个簇
- 分裂的 Divisive:
 - 从包含所有点的某个簇开始
 - 每一步分裂一个簇，直到（每个簇足够小或者簇足够多）

➤ 使用相似性或距离矩阵 similarity or distance matrix

➤ 每次合并或分裂一个簇 Merge or split one cluster at a time

➤ 基本算法

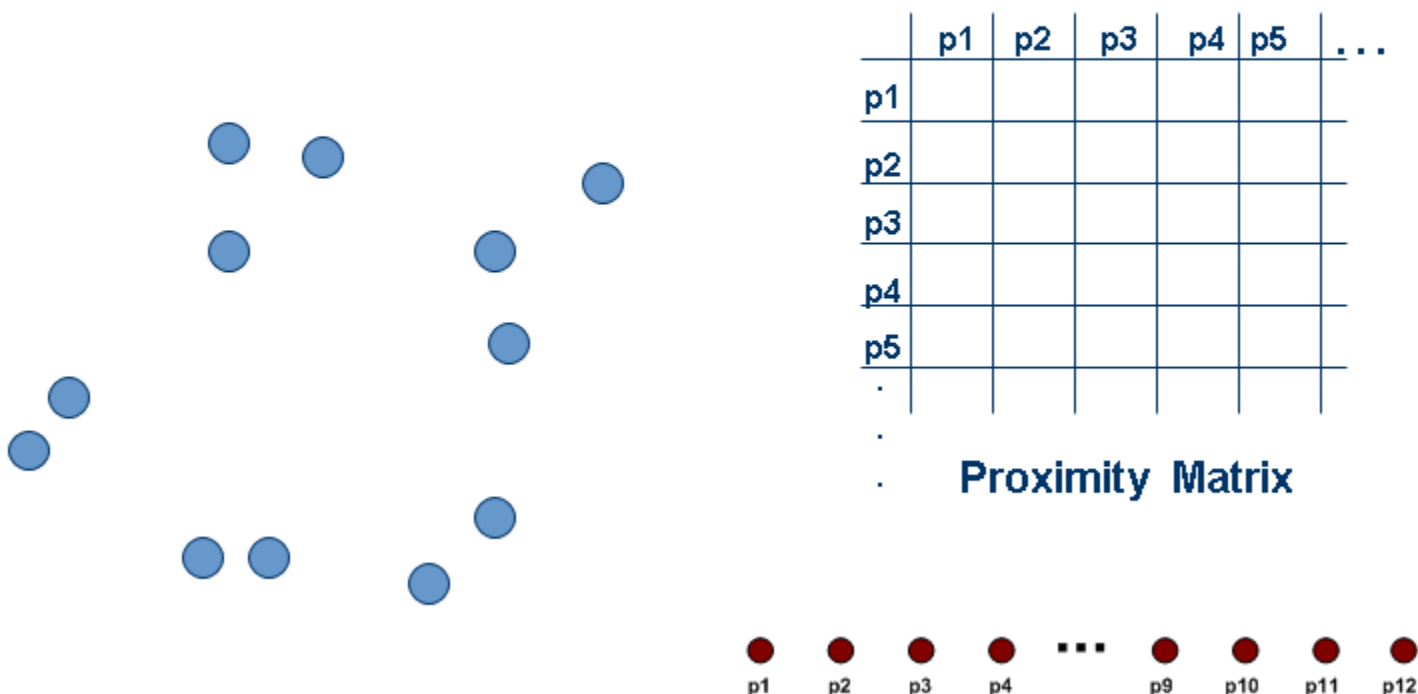
1. 计算邻近性矩阵 proximity matrix
2. Let each data point be a cluster
3. **Repeat**
4. 合并两个最接近的簇
5. 更新邻近性矩阵
6. **Until** only a single cluster remains

➤ 关键点

- 邻近性矩阵的计算

聚类分析：层次聚类

- 从点作为个体簇开始，计算邻近性矩阵



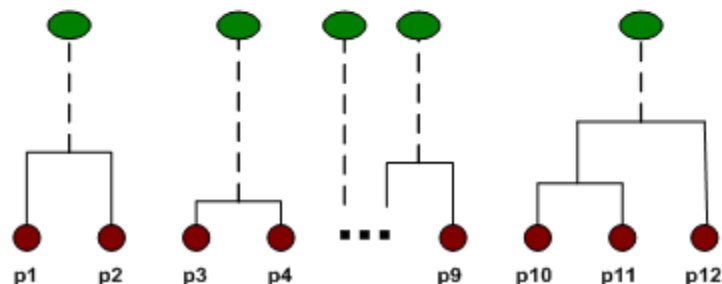
聚类分析：层次聚类

- 合并数次后，得到若干



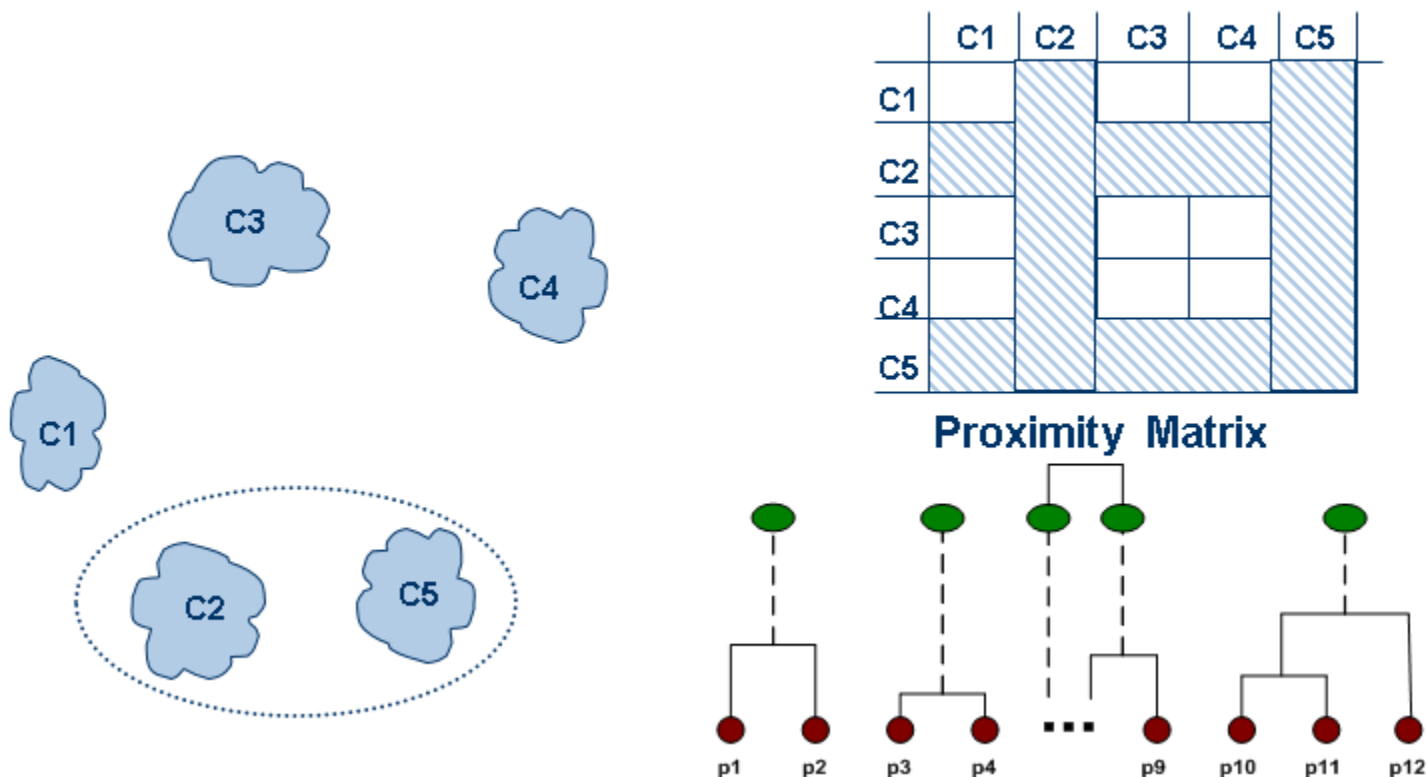
	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

Proximity Matrix



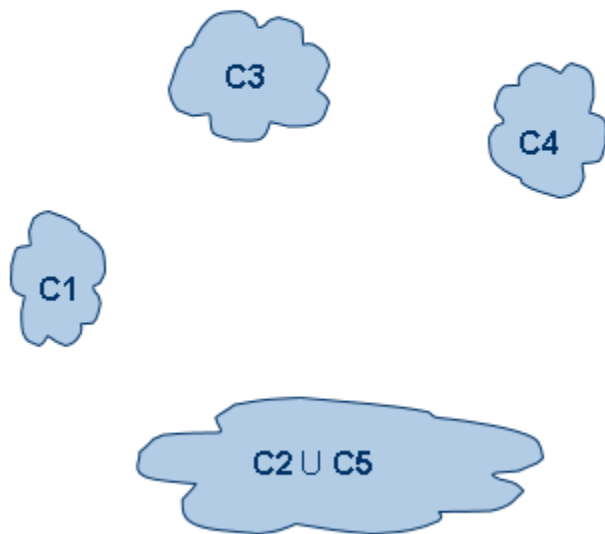
聚类分析：层次聚类

- 合并最接近的两个簇，然后更新邻近性矩阵



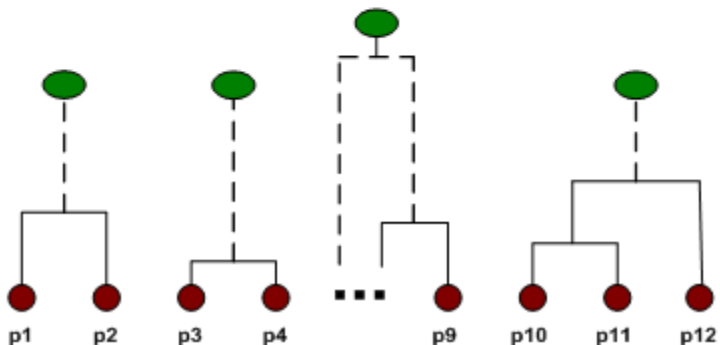
聚类分析：层次聚类

- 如何更新邻近性矩阵？



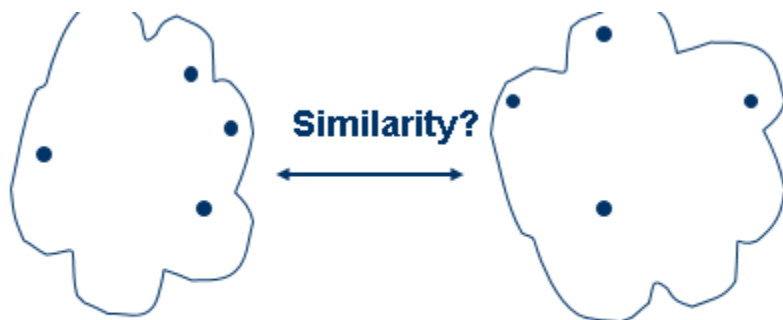
		C2 ∪ C5	C3	C4
C1		?		
C2 ∪ C5	?	?	?	?
C3		?		
C4		?		

Proximity Matrix



聚类分析：层次聚类

➤ 如何定义簇之间的邻近性？



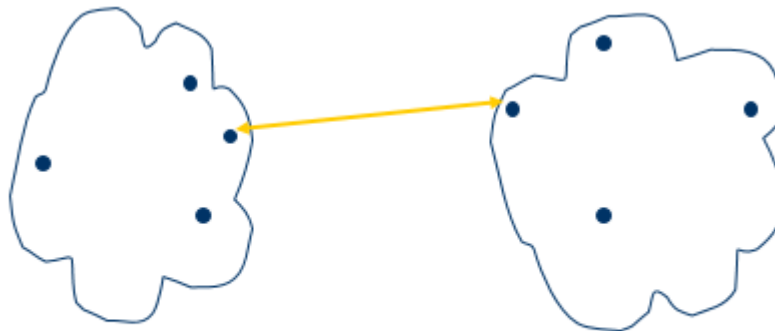
- MIN
- MAX
- 组平均距离
- 质心距离
- 使用目标函数的其他方法
 - **Ward's** 使用误差平方

	p1	p2	p3	p4	p5	...
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

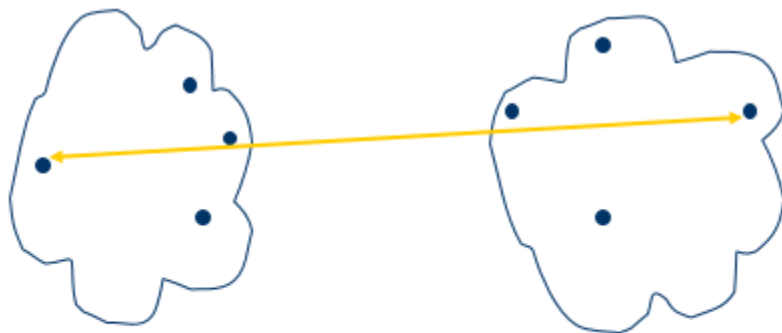
Proximity Matrix

聚类分析：层次聚类

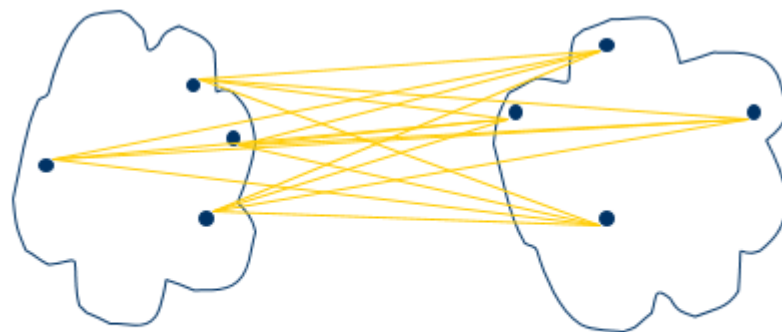
➤ 簇之间的距离



• MIN

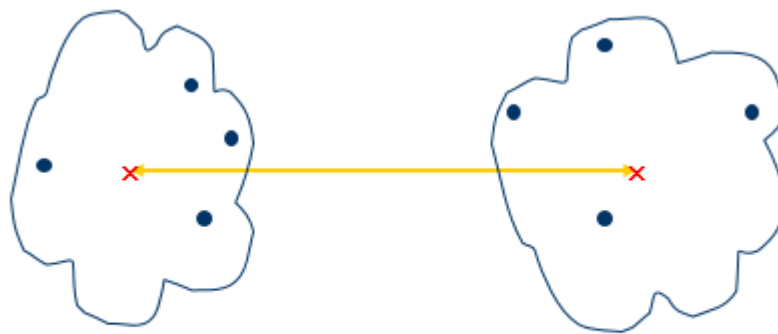


• MAX



• Group Average

➤ 簇之间的距离

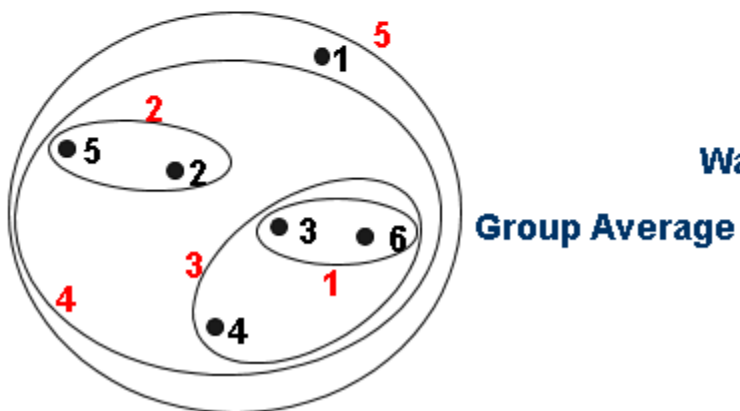
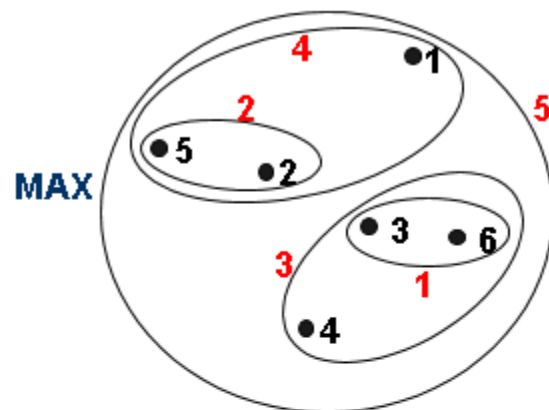
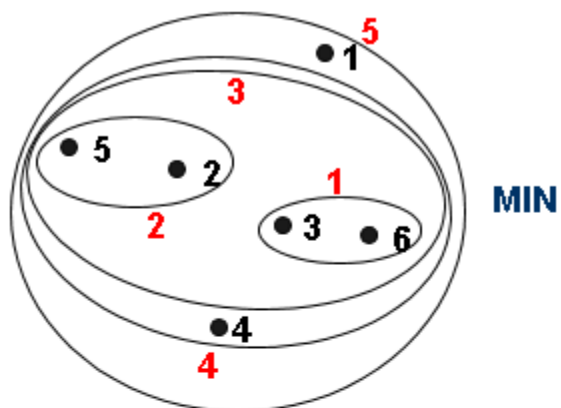


• 质心距离

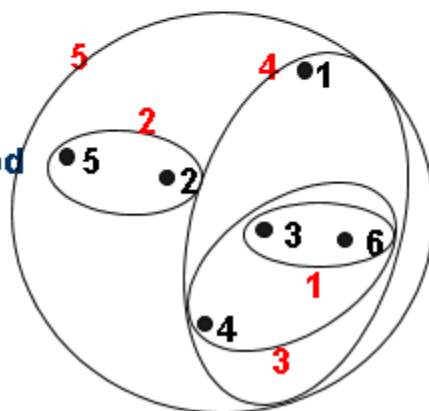
➤ Ward 's 方法

- 两个簇的临近度为两个簇合并时导致的平方误差的增量
- 当两个点的临近度取它们之间距离的平方时，与组平均非常相似

聚类分析：层次聚类



Ward's Method

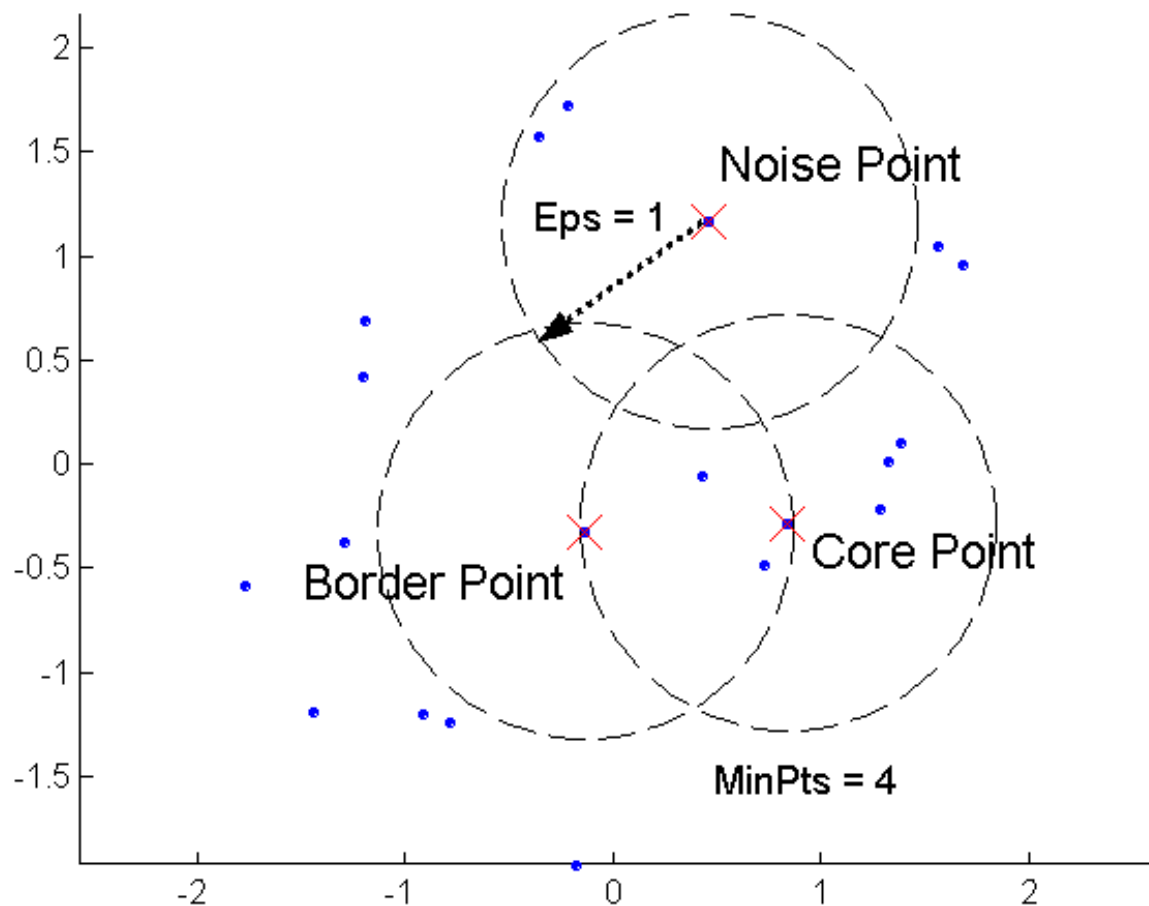


聚类分析：层次聚类

- 过程不可逆
- 目标函数不会被直接优化
- 不同的算法有不同的局限性
 - 对噪声和离群点敏感
 - 不适合凹形簇、尺寸不同的簇
 - 容易分裂大的簇

聚类分析：基于密度的聚类

➤ DBSCAN

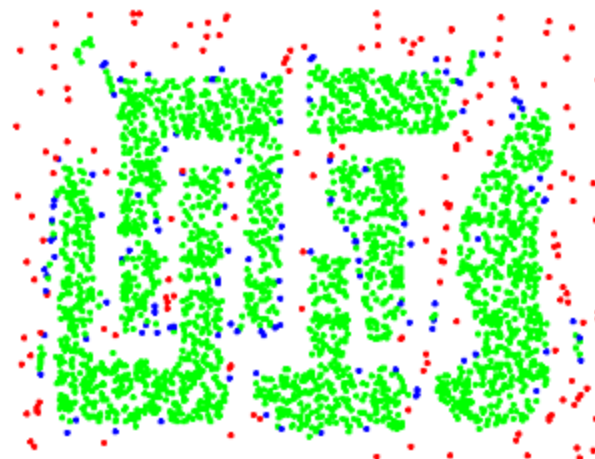


聚类分析：基于密度的聚类

➤ DBSCAN



Original Points



Point types: **core**,
border and **noise**

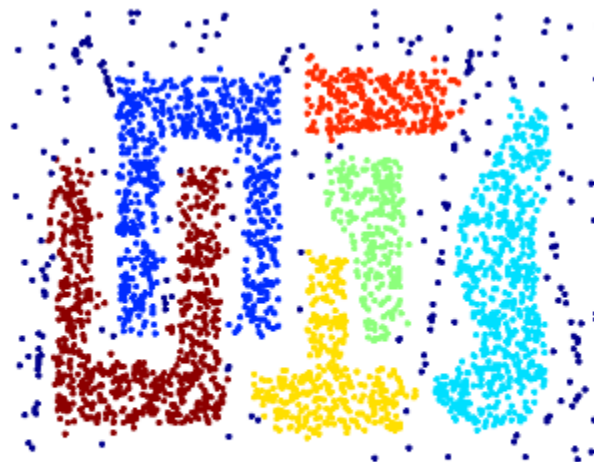
Eps = 10, MinPts = 4

聚类分析：基于密度的聚类

➤ DBSCAN



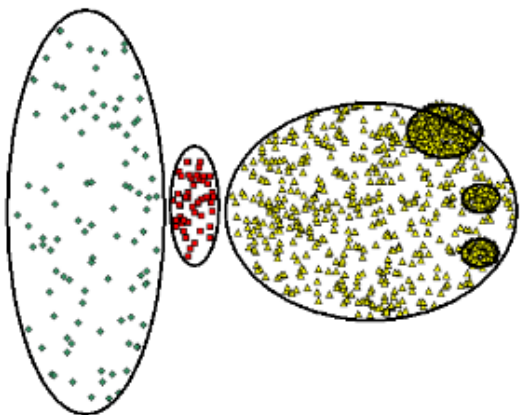
Original Points



Clusters

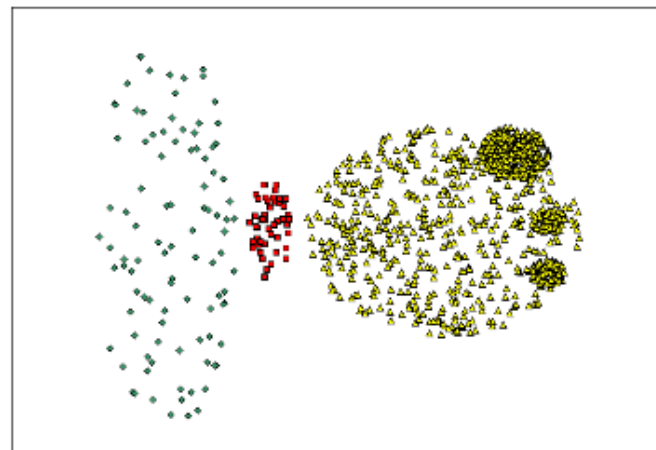
聚类分析：基于密度的

➤ DBSCAN的局限性

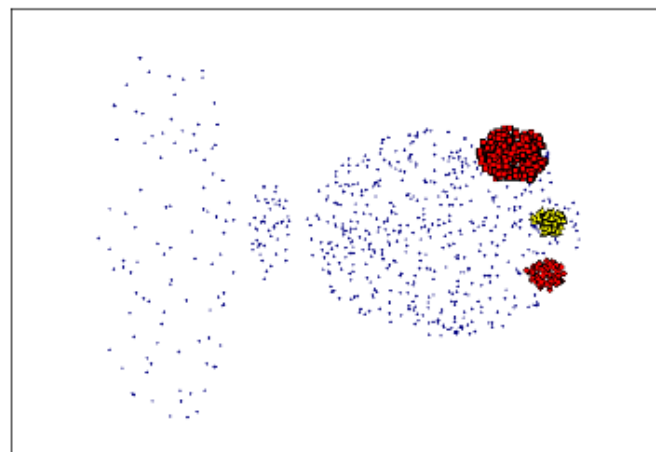


Original Points

- 簇的密度变化过大
- 高维数据



(MinPts=4, Eps=9.75).

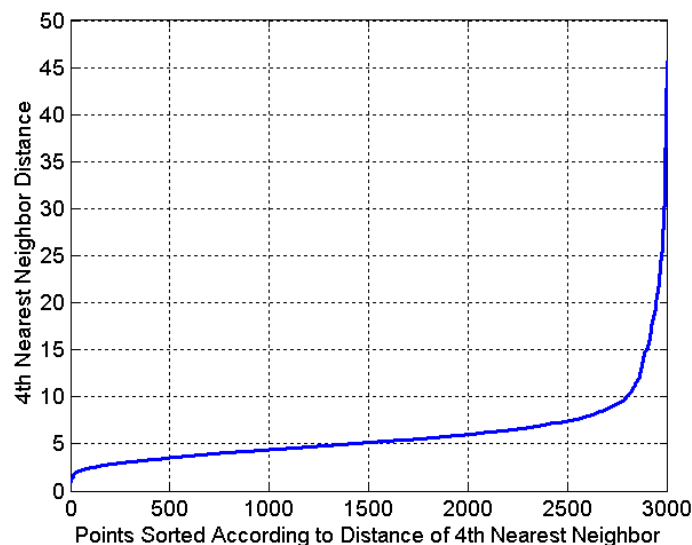


(MinPts=4, Eps=9.92)

聚类分析：基于密度的

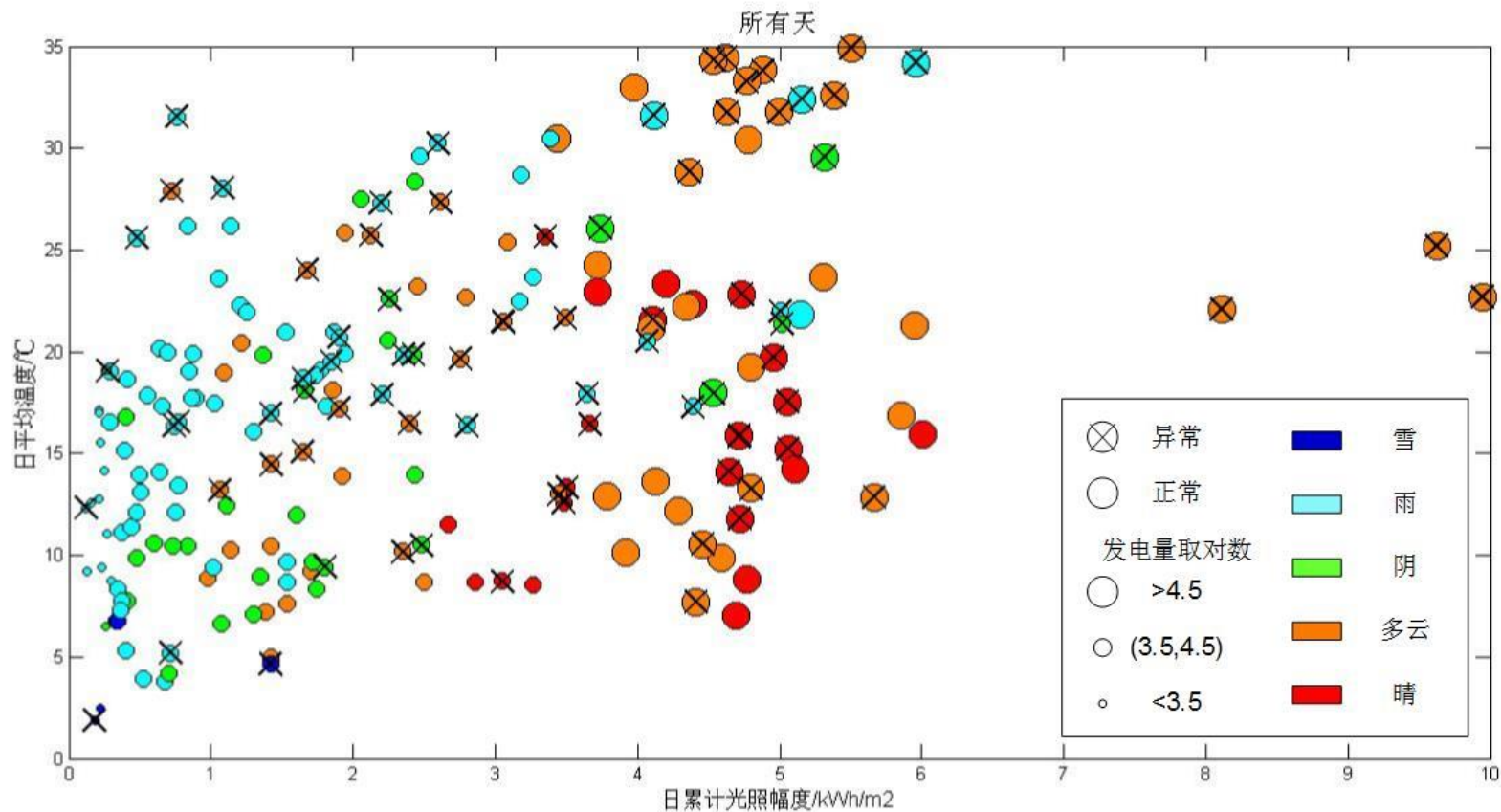
➤ 如何确定EPS和MinPts

- 簇内的每点的第kth近邻距离相近
- 噪声点的第kth近邻的距离较远
- 如，绘制所有点的第4近邻的距离

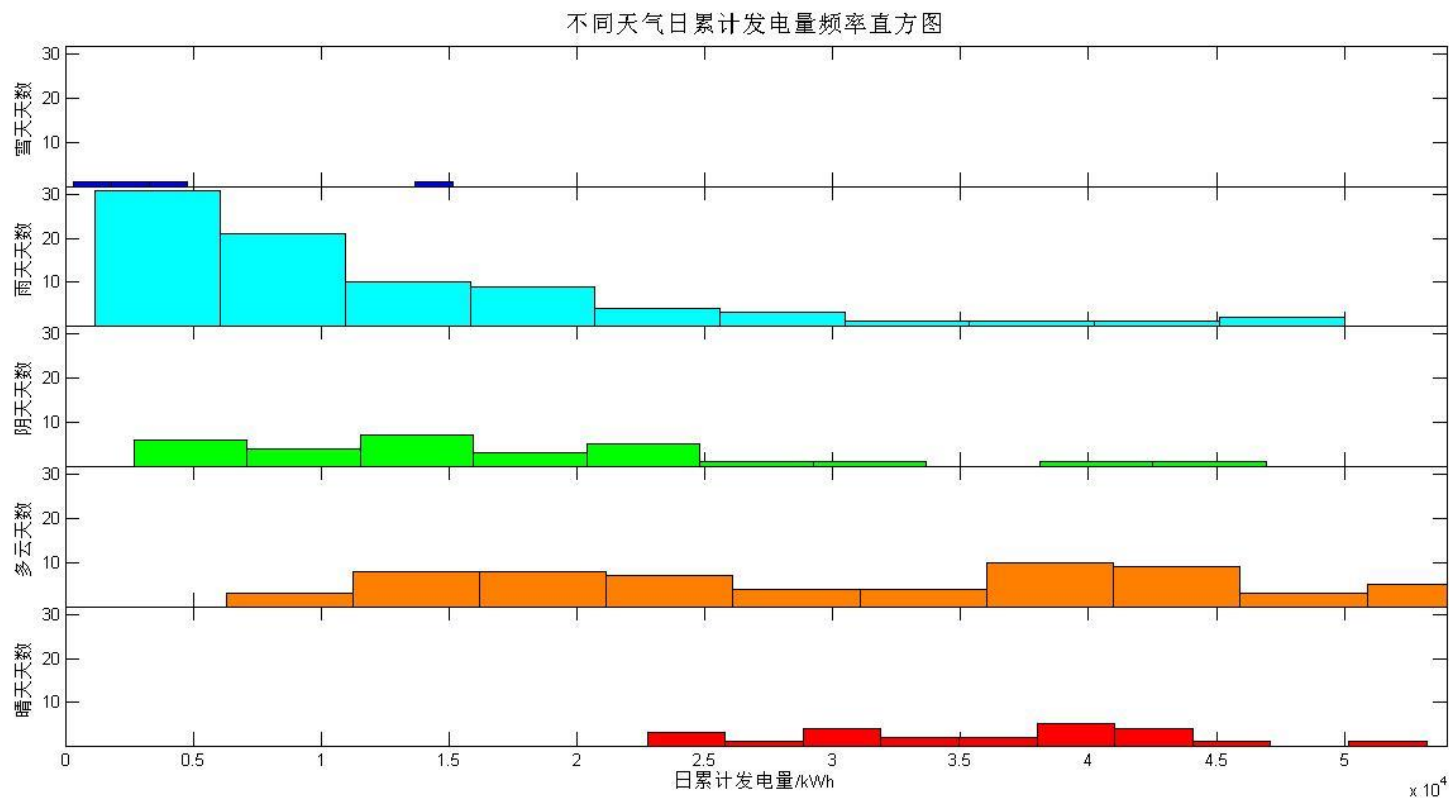


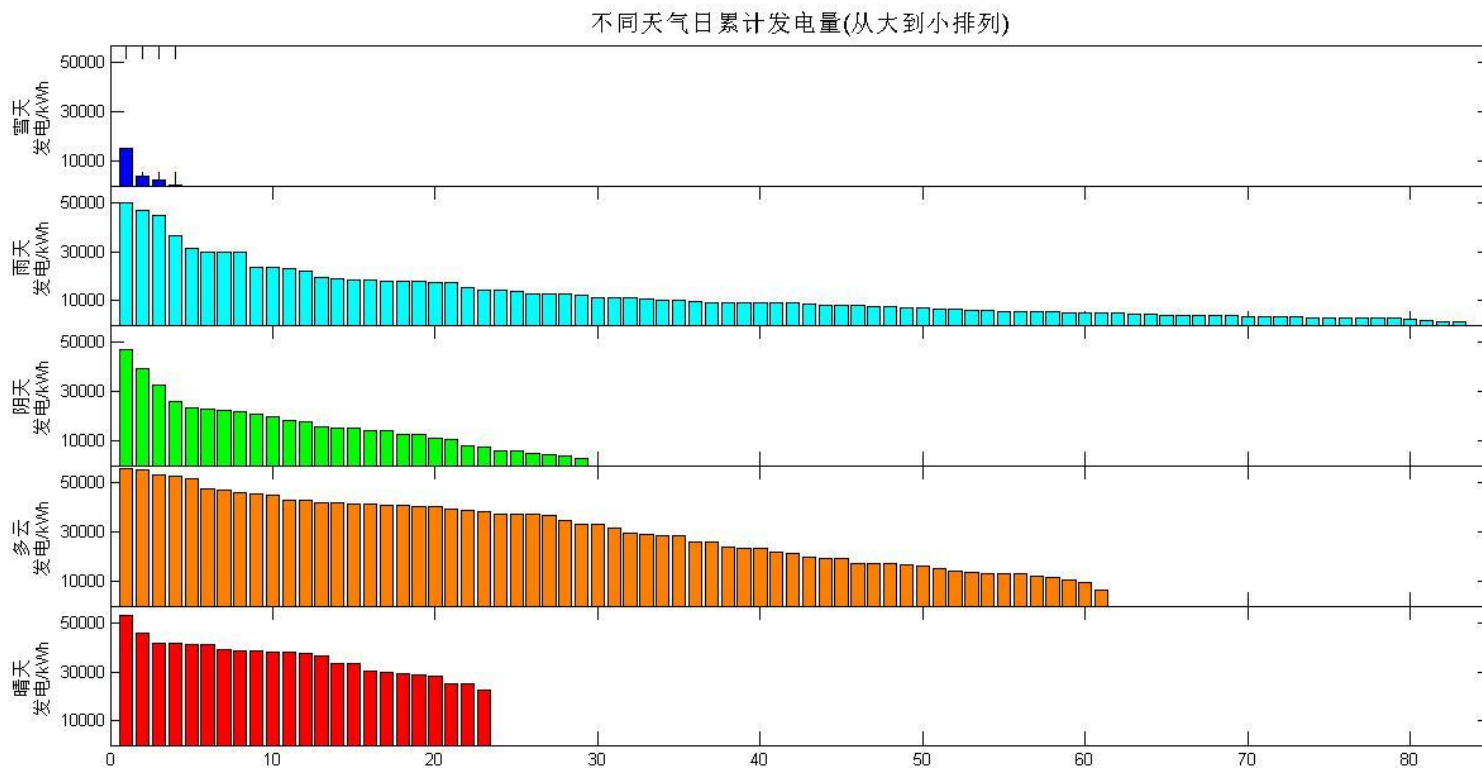
聚类分析：

➤ 扩展话题



聚类分析：





聚类分析：簇的确认

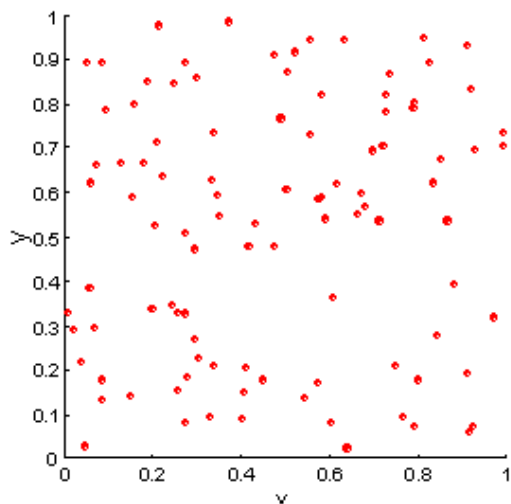
- 如何验证和评价聚类分析的结果？
 - “goodness” of the resulting clusters?
- 目的：
 - 避免发现噪声产生的模式
 - 比较不同的聚类算法
 - 比较不同的簇集合
 - 簇之间的比较

聚类分析：簇的确认

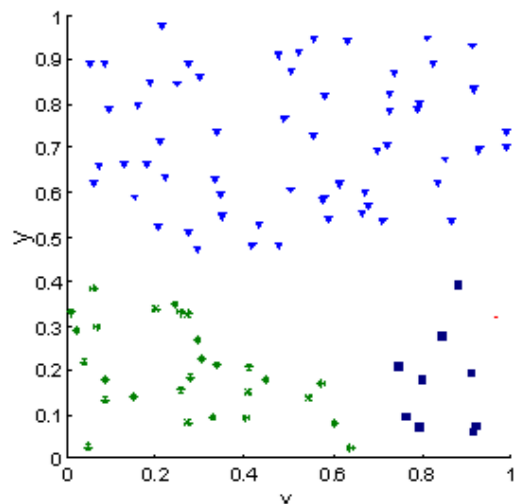


小象学院
ChinaHadoop.cn

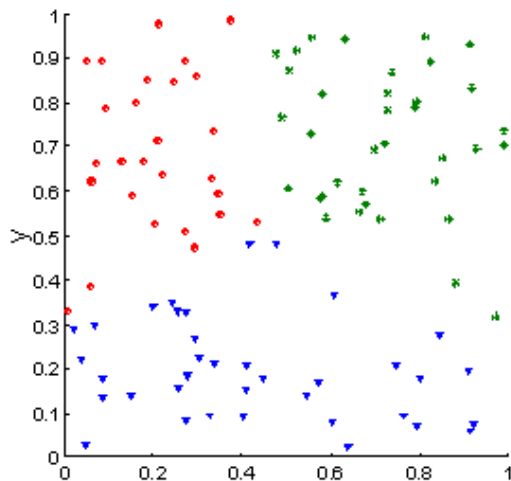
Random
Points



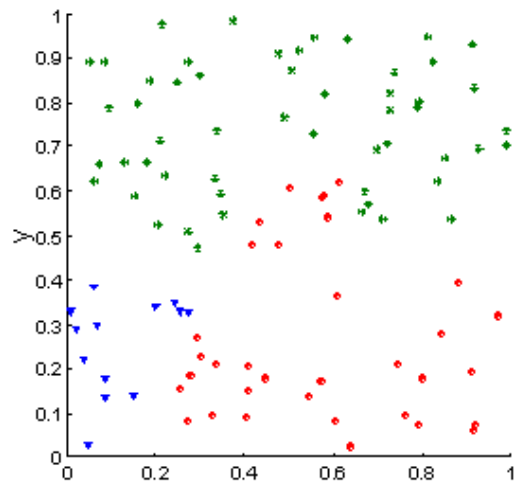
DBSCAN



K-means



Complete
Link



聚类分析：簇的确认

1. 确定数据集的聚类趋势 **clustering tendency** , 即是否存在非随机结构
2. 确定正确的簇的个数.
3. 评估聚类分析结果对数据的拟合情况 - Use only the data
4. 将聚类分析的结果跟已知的客观结果 (如 , 外部提供的类标号) 比较
5. 比较不同的聚类方法的优劣.

1,2,3 非监督

3,4,5 进一步区分是评估整个聚类还是单个簇

三种度量方式：**外部指标**（**监督的**，如：熵），**内部指标**（**非监督的**，如：**SSE**），**相对指标**

➤ 比较两个矩阵

- 邻近性矩阵 Proximity Matrix
- 理想的邻近性矩阵 “Incidence” Matrix
 - 每个数据点对应一行一列
 - 矩阵中每项对应的两点如果是同簇，为1
 - 否则为 0
- 计算两个矩阵的相关性
- 高相关-->簇中的点相近
 - 但并不太适合基于密度和连接的簇

聚类分析：簇的确认

➤ 只需要计算 $n(n-1) / 2$ 个

	p1	p2	p3	p4	p5	...
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

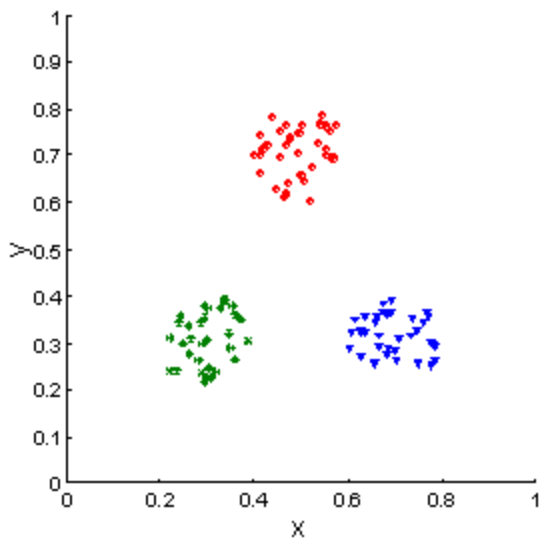
Proximity Matrix

	p1	p2	p3	p4	p5	...
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

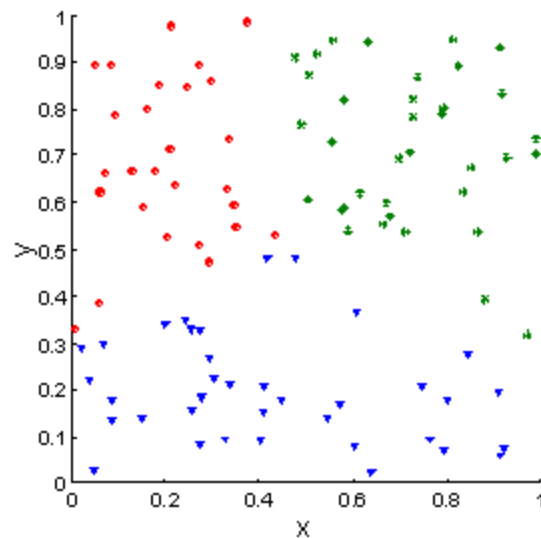
incidence Matrix

聚类分析：簇的确认

➤ 相关系数



Corr = -0.9235

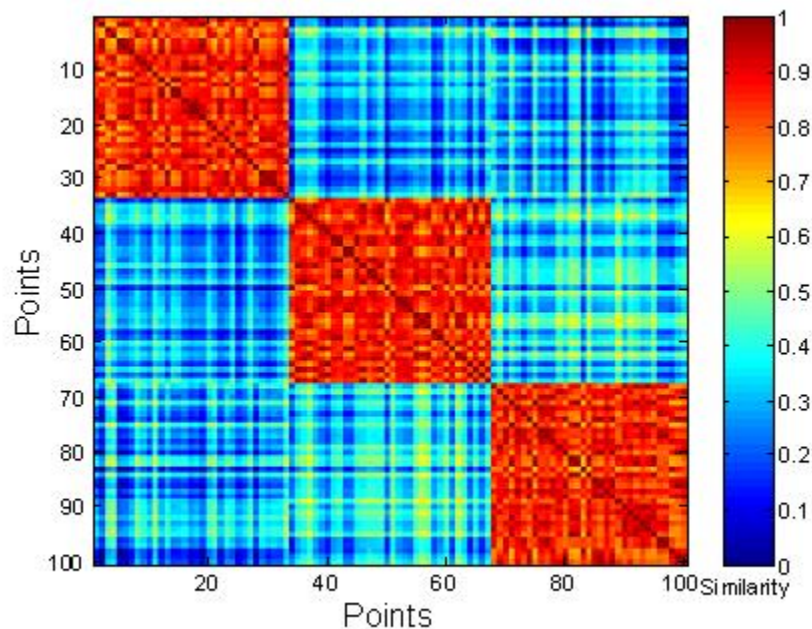
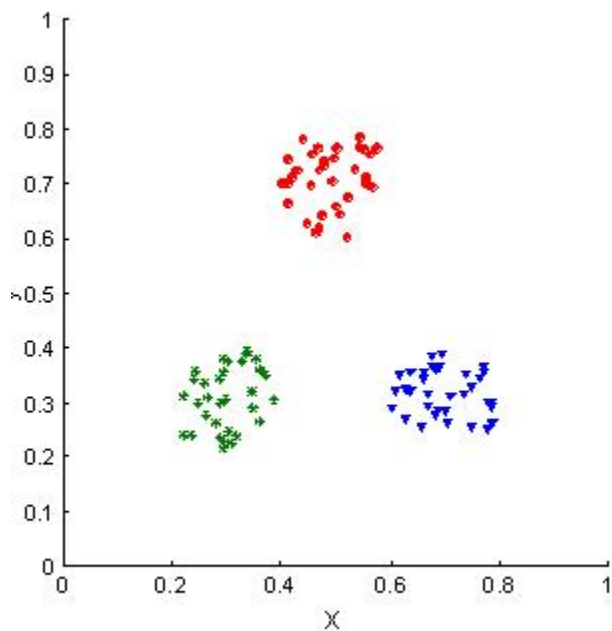


Corr = -0.5810

聚类分析：簇的确认

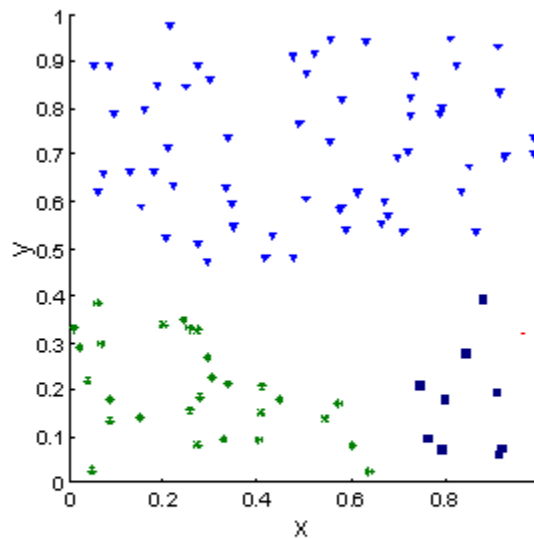
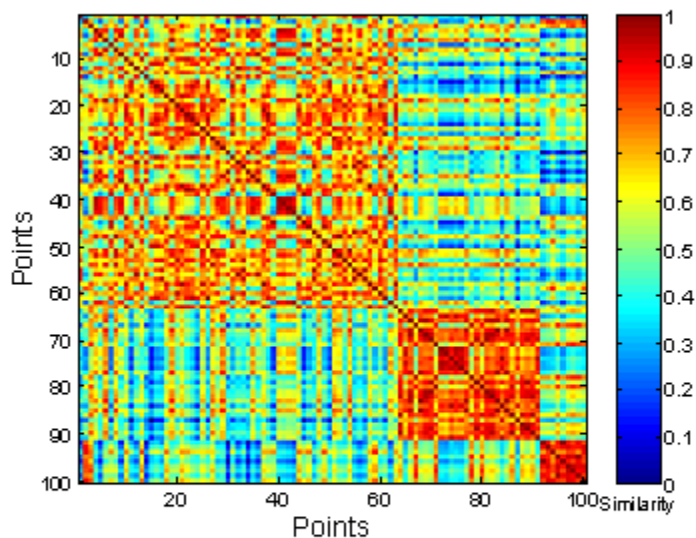
➤ 通过邻近性矩阵

- 根据簇对数据排序后的邻近性矩阵
- 可视化



聚类分析：簇的确认

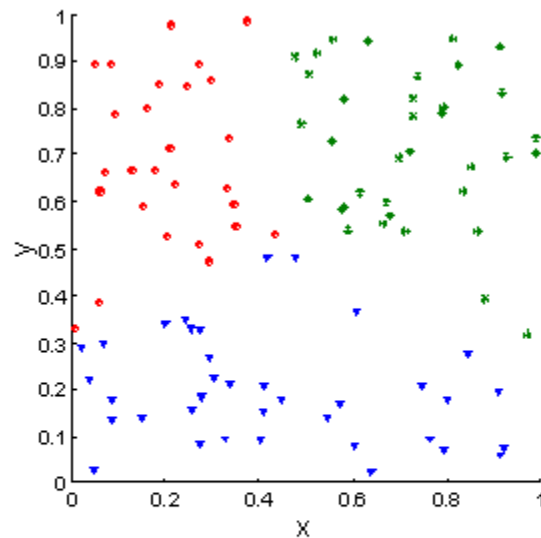
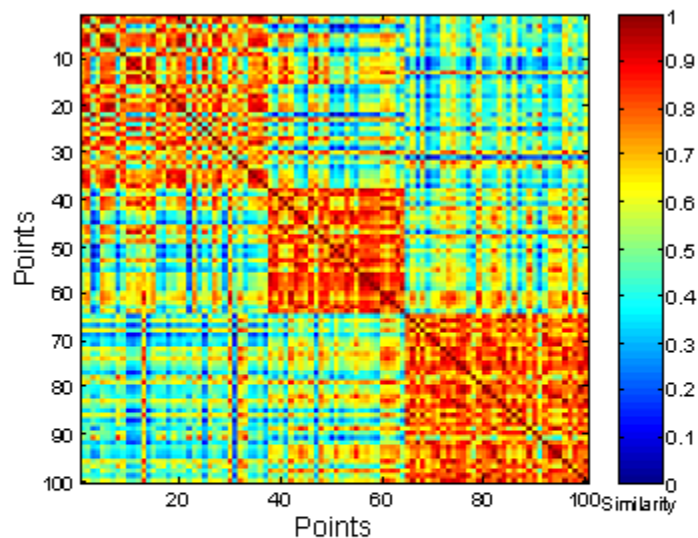
➤ 随机点产生的簇



DBSCAN

聚类分析：簇的确认

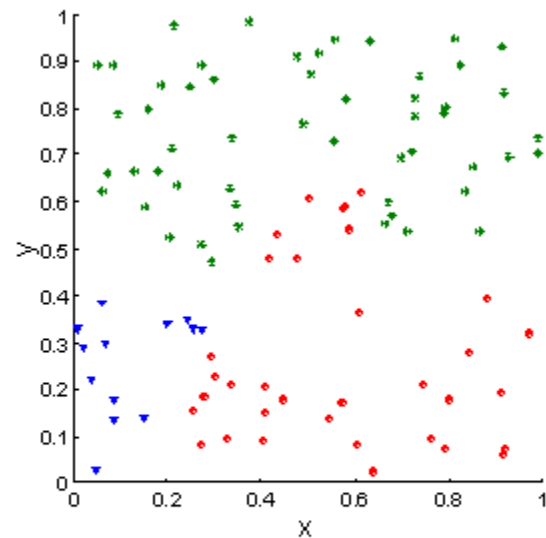
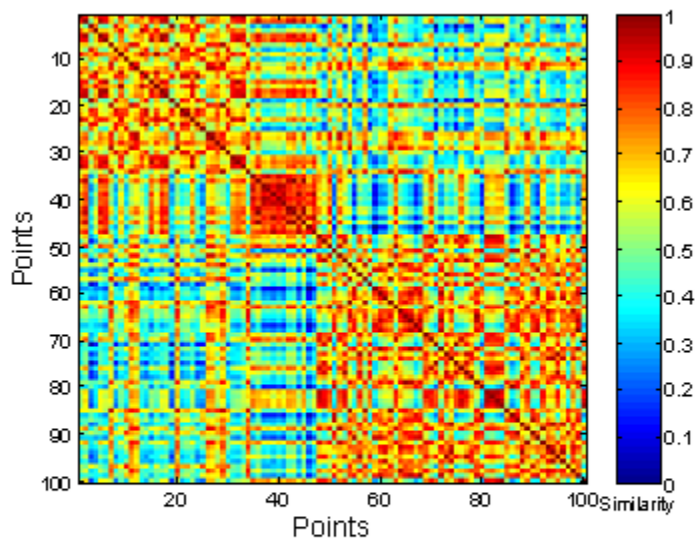
➤ 随机点产生的簇



K-means

聚类分析：簇的确认

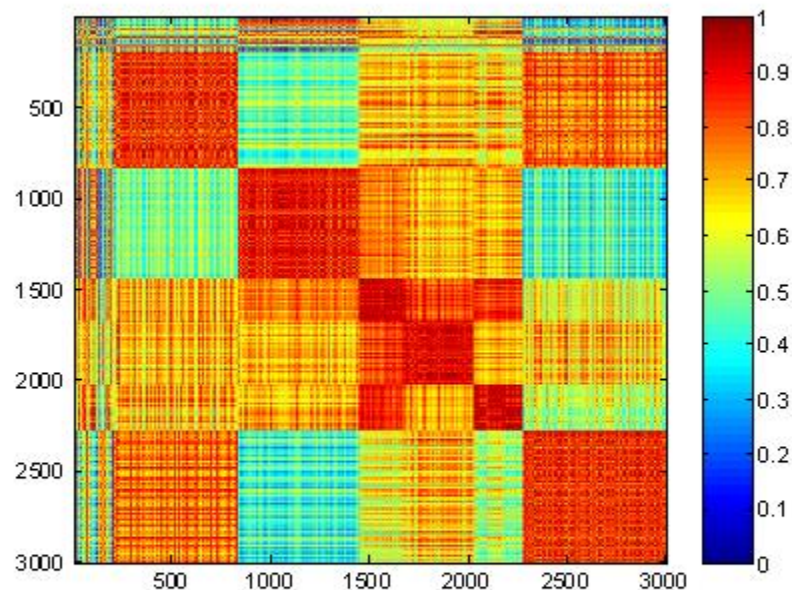
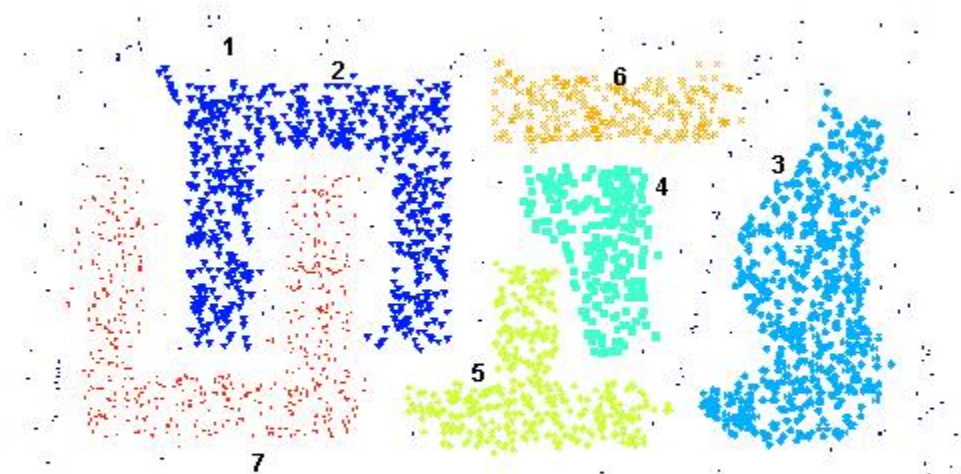
➤ 随机点产生的簇



Complete Link

聚类分析：簇的确认

➤ 邻近性矩阵

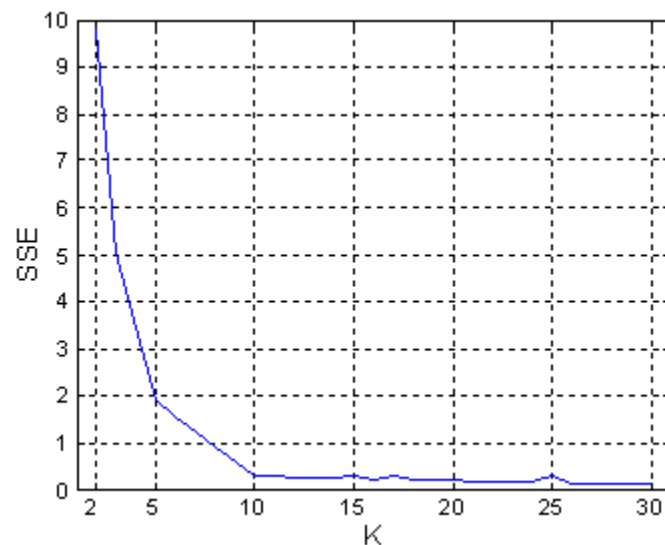
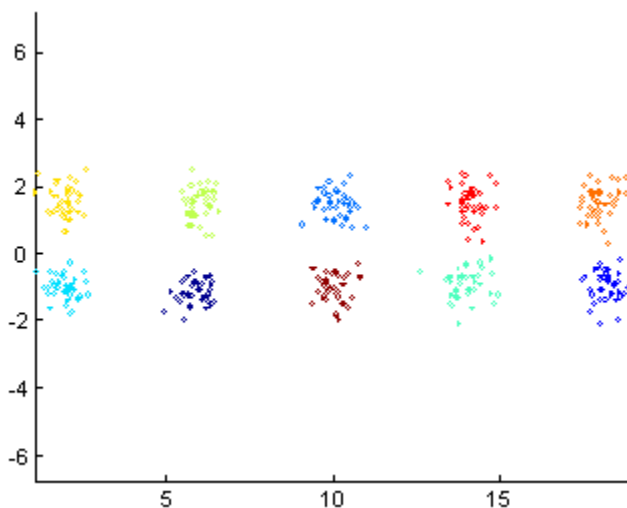


DBSCAN

聚类分析：簇的确认

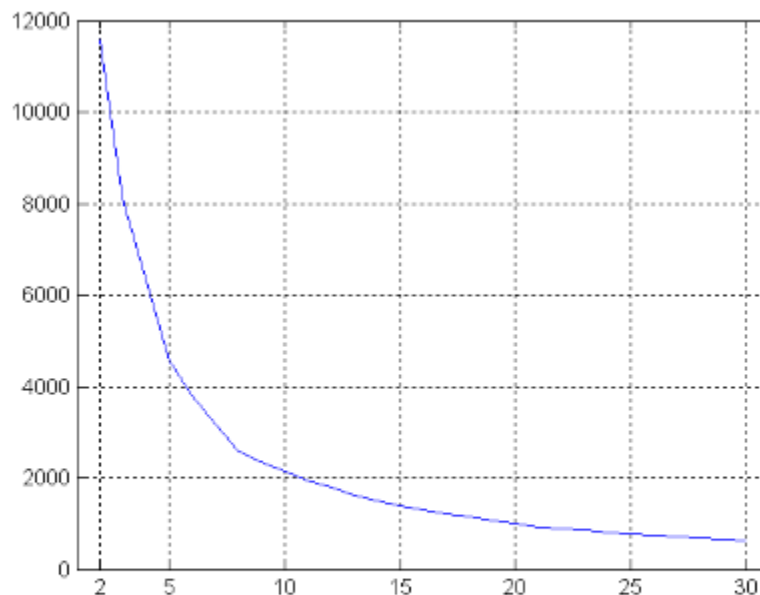
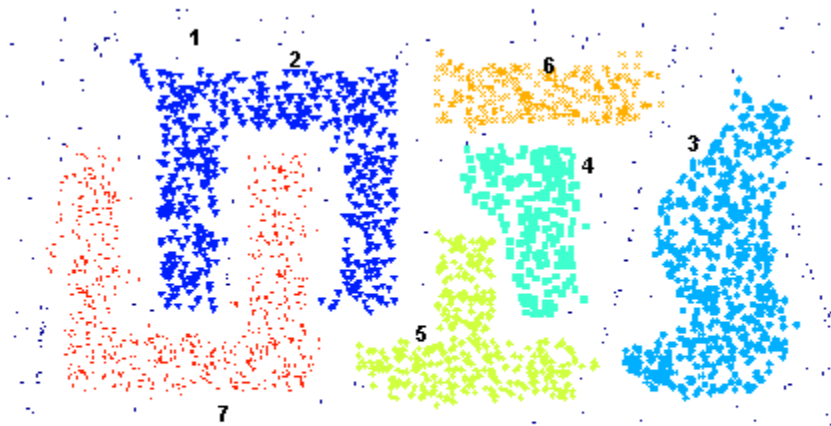
➤ 内部指标

- SSE 适合评估多个簇集或者多个簇 (average SSE).
- 也可以用来估计簇的个数



聚类分析：簇的确认

➤ SSE曲线



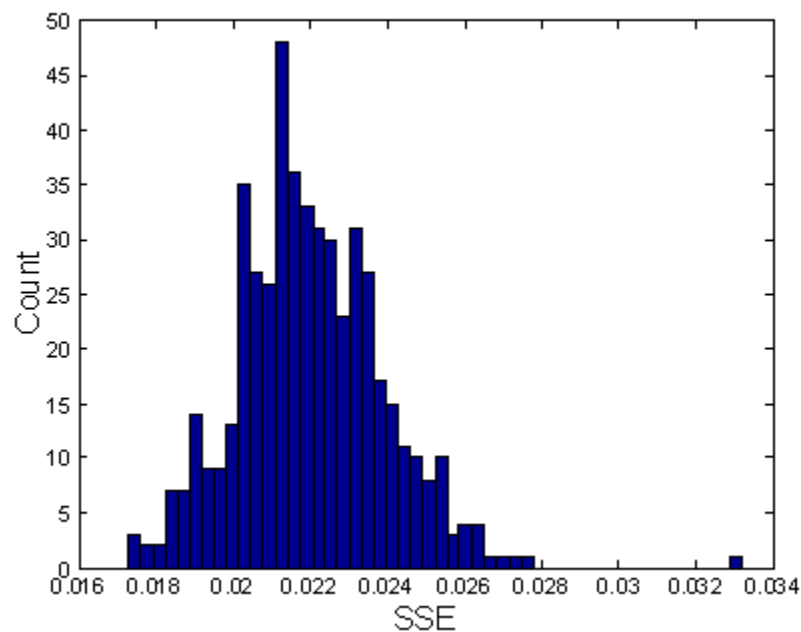
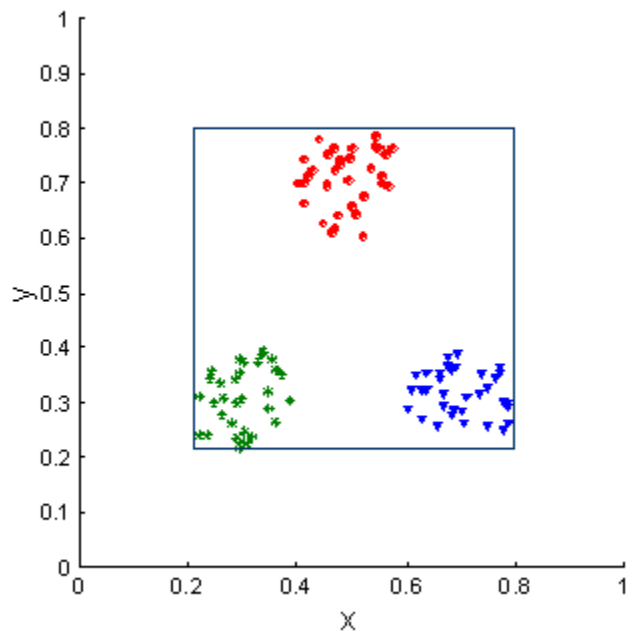
SSE of clusters found using K-means

➤ 统计框架

- 需要框架来解释度量
- 统计学角度：
 - 聚类结果得分的典型性意味着结果的正确性
 - 比较随机数据和聚类后的数据的某项指标
- 如果是相互比较，则框架的必要性降低

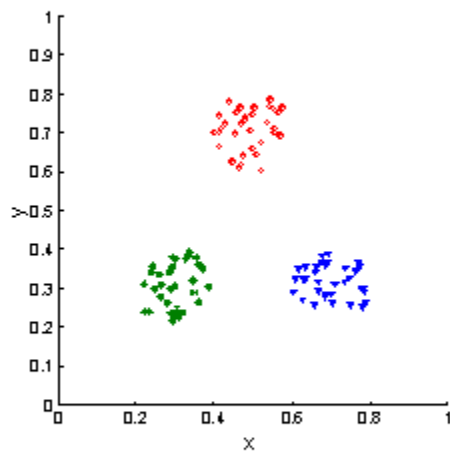
聚类分析：

- 三个簇的SSE ~ 0.005
- Histogram shows SSE of three clusters in 500 sets of random data points of size 100 distributed over the range 0.2 – 0.8 for x and y values

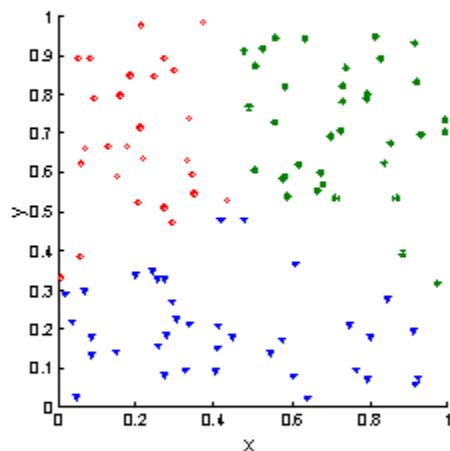


聚类分析：簇的确认

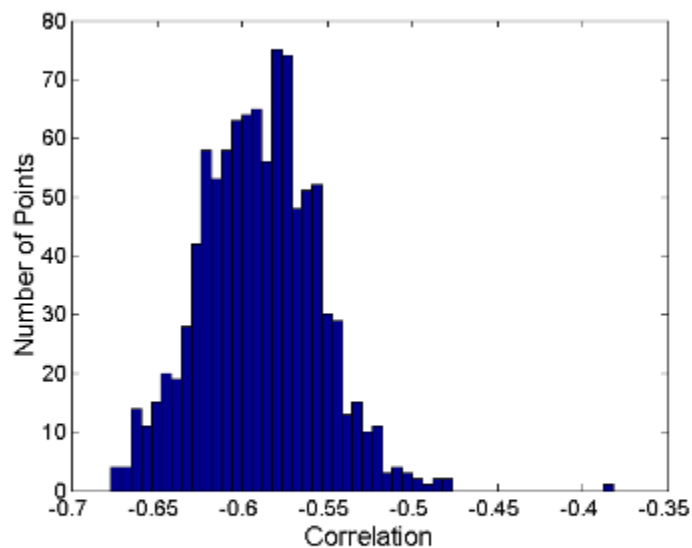
➤ 邻近性矩阵和理想邻近性矩阵的比较



Corr = -0.9235



Corr = -0.5810



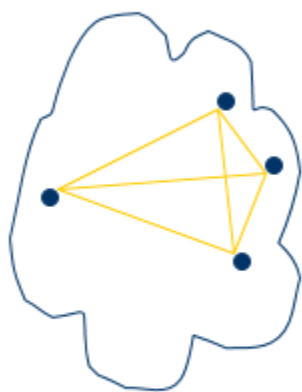
聚类分析：簇的确认

➤ 簇凝聚度 Cluster Cohesion v.s. 簇分离度 Cluster Separation

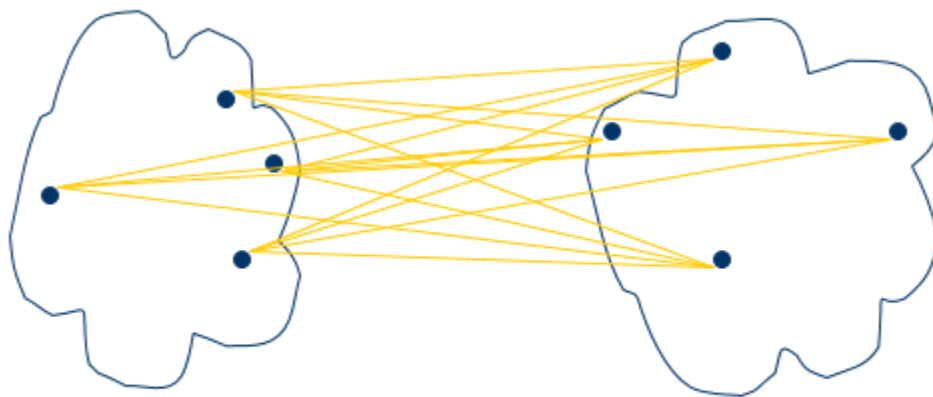
➤ 例：

$$WSS = \sum_i \sum_{x \in C_i} (x - m_i)^2$$

$$BSS = \sum_i |C_i| (m - m_i)^2$$



cohesion



separation

➤ 外部指标

Table 5.9. K-means Clustering Results for LA Document Data Set

Cluster	Entertainment	Financial	Foreign	Metro	National	Sports	Entropy	Purity
1	3	5	40	506	96	27	1.2270	0.7474
2	4	7	280	29	39	2	1.1472	0.7756
3	1	1	1	7	4	671	0.1813	0.9796
4	10	162	3	119	73	2	1.7487	0.4390
5	331	22	5	70	13	23	1.3976	0.7134
6	5	358	12	212	48	13	1.5523	0.5525
Total	354	555	341	943	273	738	1.1450	0.7203

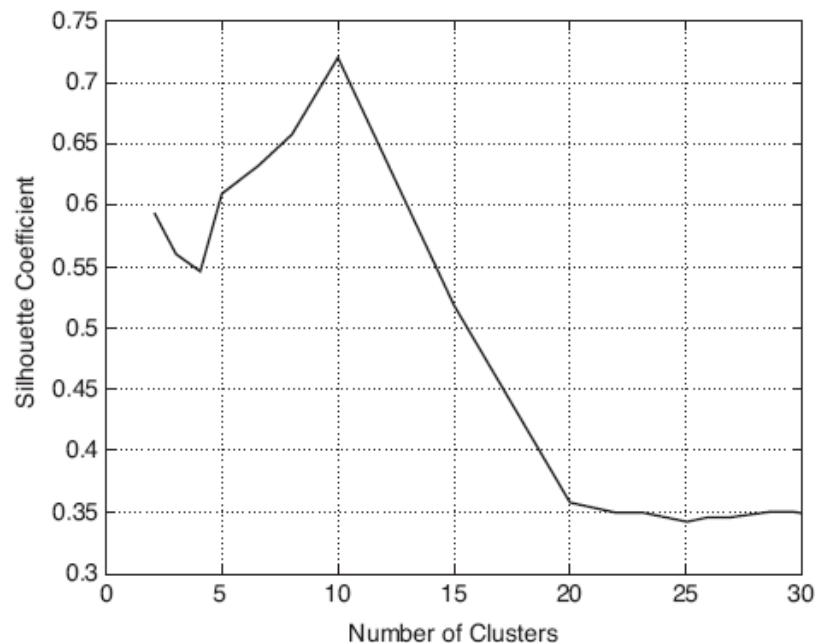
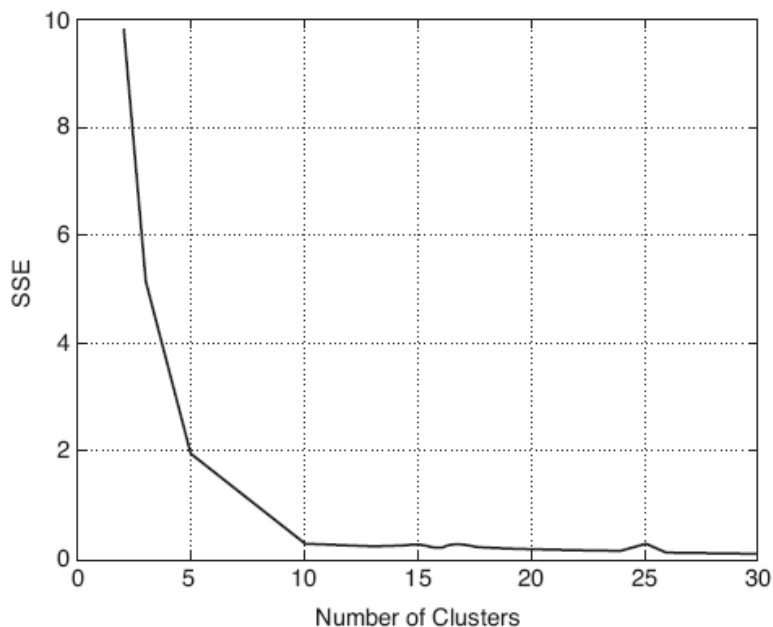
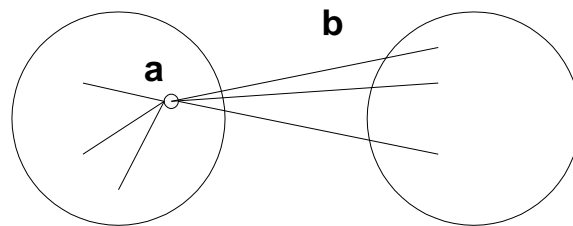
entropy For each cluster, the class distribution of the data is calculated first, i.e., for cluster j we compute p_{ij} , the ‘probability’ that a member of cluster j belongs to class i as follows: $p_{ij} = m_{ij}/m_j$, where m_j is the number of values in cluster j and m_{ij} is the number of values of class i in cluster j . Then using this class distribution, the entropy of each cluster j is calculated using the standard formula $e_j = \sum_{i=1}^L p_{ij} \log_2 p_{ij}$, where the L is the number of classes. The total entropy for a set of clusters is calculated as the sum of the entropies of each cluster weighted by the size of each cluster, i.e., $e = \sum_{j=1}^K \frac{m_j}{m} e_j$, where m_j is the size of cluster j , K is the number of clusters, and m is the total number of data points.

purity Using the terminology derived for entropy, the purity of cluster j , is given by $purity_j = \max_i p_{ij}$ and the overall purity of a clustering by $purity = \sum_{j=1}^K \frac{m_j}{m} purity_j$.

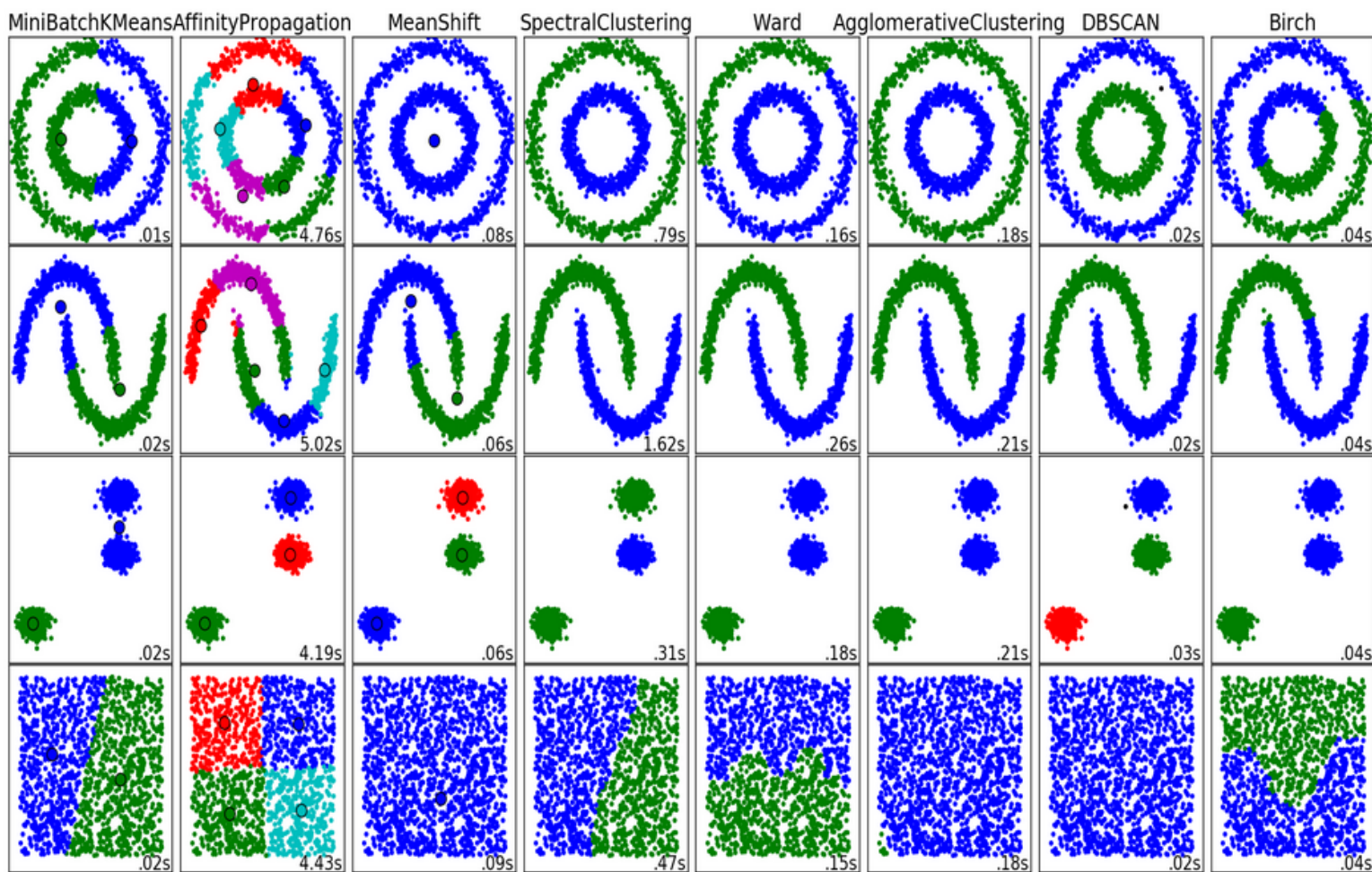
聚类分析：簇的确认

➤ 轮廓系数 Silhouette Coefficient

- a = 簇内点之间的平均距离
- b = \min (到簇外点的平均距离)
- $s = 1 - a/b$ if $a < b$, (or $s = b/a - 1$ if $a > b$)



聚类分析：scikit-learn

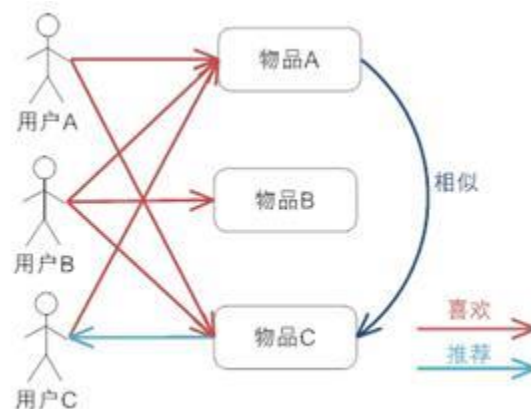
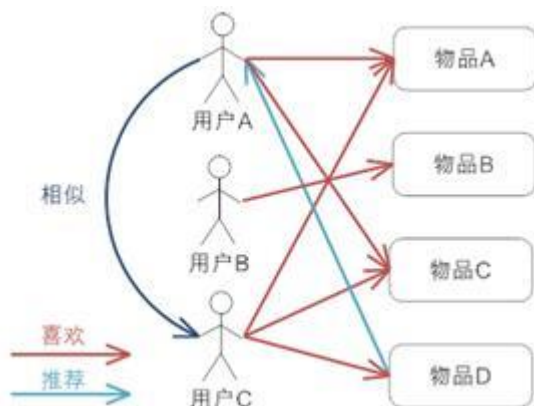
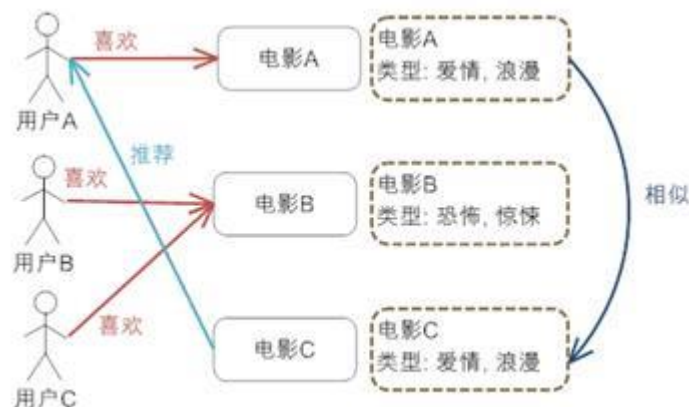
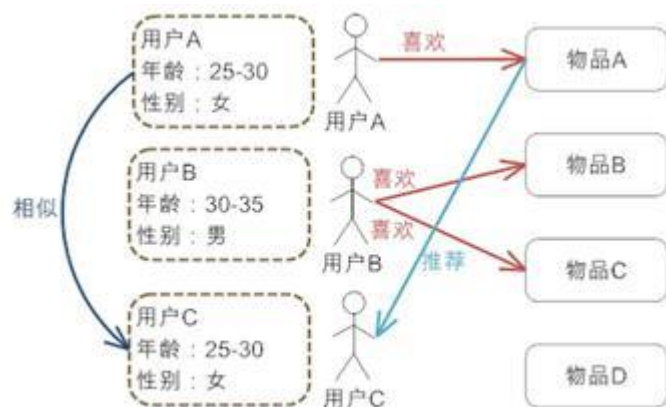


A comparison of the clustering algorithms in scikit-learn

聚类分析：scikit-learn

Method name	Parameters	Scalability	Usecase	Geometry (metric used)
K-Means	number of clusters	Very large <code>n_samples</code> , medium <code>n_clusters</code> with MiniBatch code	General-purpose, even cluster size, flat geometry, not too many clusters	Distances between points
Affinity propagation	damping, sample preference	Not scalable with <code>n_samples</code>	Many clusters, uneven cluster size, non-flat geometry	Graph distance (e.g. nearest-neighbor graph)
Mean-shift	bandwidth	Not scalable with <code>n_samples</code>	Many clusters, uneven cluster size, non-flat geometry	Distances between points
Spectral clustering	number of clusters	Medium <code>n_samples</code> , small <code>n_clusters</code>	Few clusters, even cluster size, non-flat geometry	Graph distance (e.g. nearest-neighbor graph)
Ward hierarchical clustering	number of clusters	Large <code>n_samples</code> and <code>n_clusters</code>	Many clusters, possibly connectivity constraints	Distances between points
Agglomerative clustering	number of clusters, linkage type, distance	Large <code>n_samples</code> and <code>n_clusters</code>	Many clusters, possibly connectivity constraints, non Euclidean distances	Any pairwise distance
DBSCAN	neighborhood size	Very large <code>n_samples</code> , medium <code>n_clusters</code>	Non-flat geometry, uneven cluster sizes	Distances between nearest points
Gaussian mixtures	many	Not scalable	Flat geometry, good for density estimation	Mahalanobis distances to centers
Birch	branching factor, threshold, optional global clusterer.	Large <code>n_clusters</code> and <code>n_samples</code>	Large dataset, outlier removal, data reduction.	Euclidean distance between points

推荐引擎（个性化）



练习：豆瓣书评数据

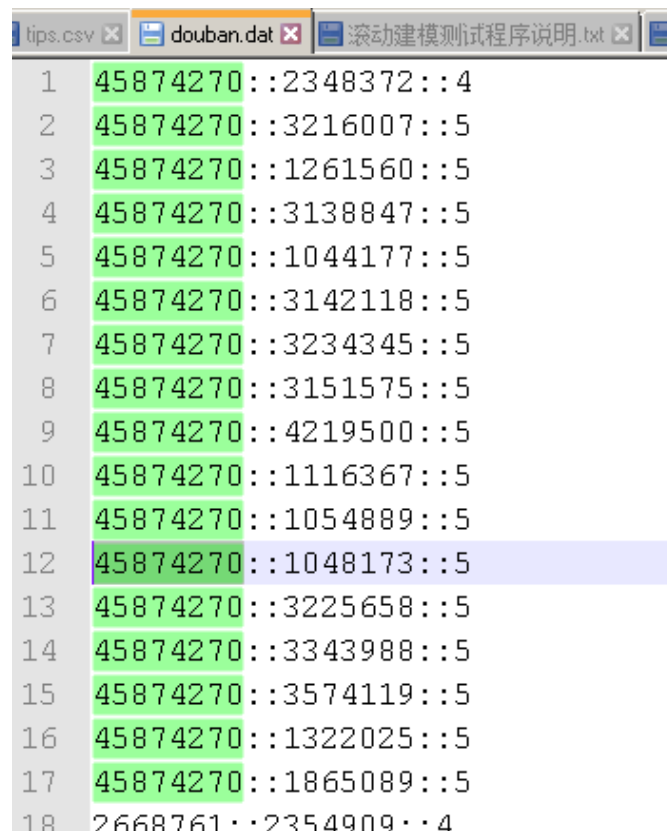
➤ 数据：

– 链接: <http://pan.baidu.com/s/1bpKAd8V> 密码: dw8g

➤ 书的信息样例：<https://book.douban.com/subject/1048173/>

➤ 目标：

- 1. 找出跟某一用户相似的若干读者
- 2. 找出跟某一本书有类似读者群的书
- 3. 为读者划分类型
- 4. 为书划分类型
- 5. 为某一读者推荐他没有读过的书



tips.csv	douban.dat	滚动建模测试程序说明.txt
1	45874270::2348372::4	
2	45874270::3216007::5	
3	45874270::1261560::5	
4	45874270::3138847::5	
5	45874270::1044177::5	
6	45874270::3142118::5	
7	45874270::3234345::5	
8	45874270::3151575::5	
9	45874270::4219500::5	
10	45874270::1116367::5	
11	45874270::1054889::5	
12	45874270::1048173::5	
13	45874270::3225658::5	
14	45874270::3343988::5	
15	45874270::3574119::5	
16	45874270::1322025::5	
17	45874270::1865089::5	
18	2668761::2354909::4	

联系我们:

- 新浪微博: ChinaHadoop
- 微信公号: ChinaHadoop
- 网站: <http://chinahadoop.cn>
- 问答社区: <http://wenda.ChinaHadoop.cn>

