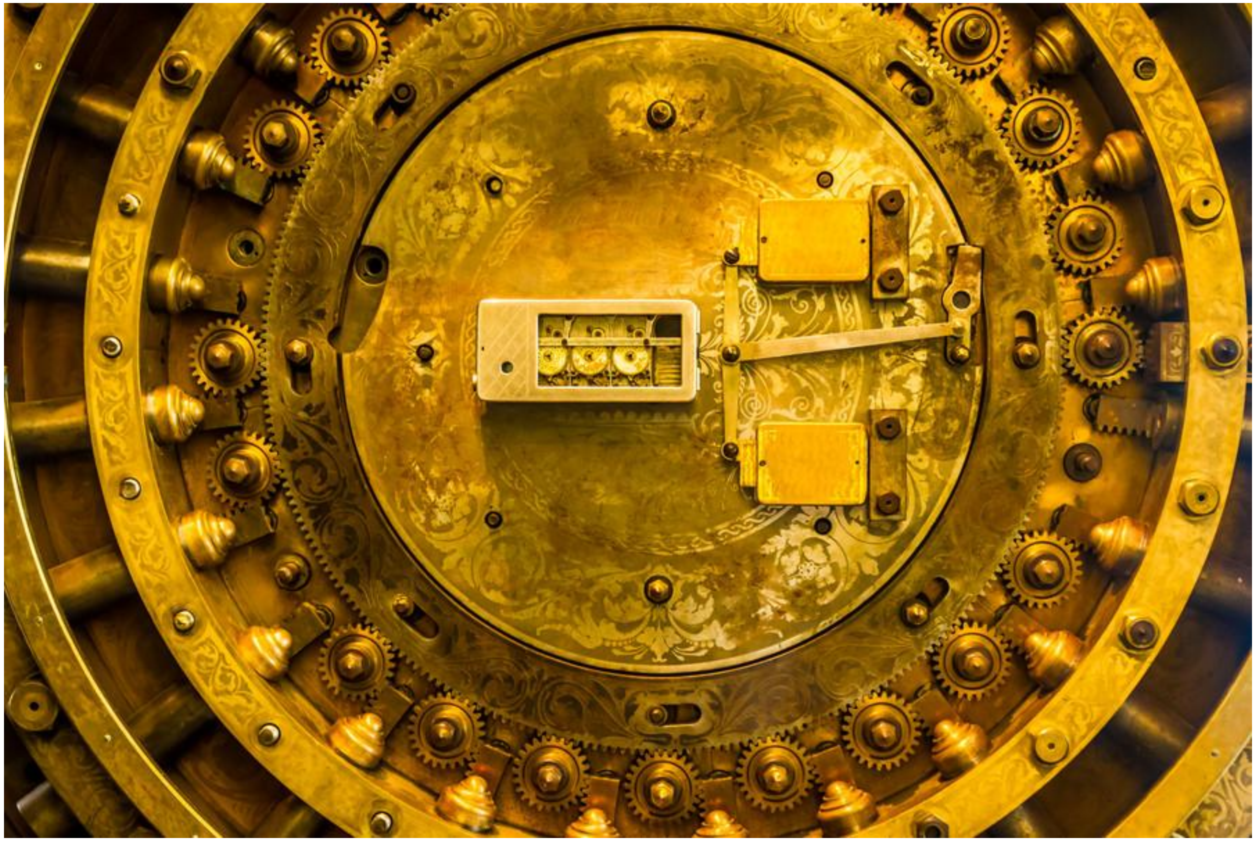


机器学习 第11周

法律声明

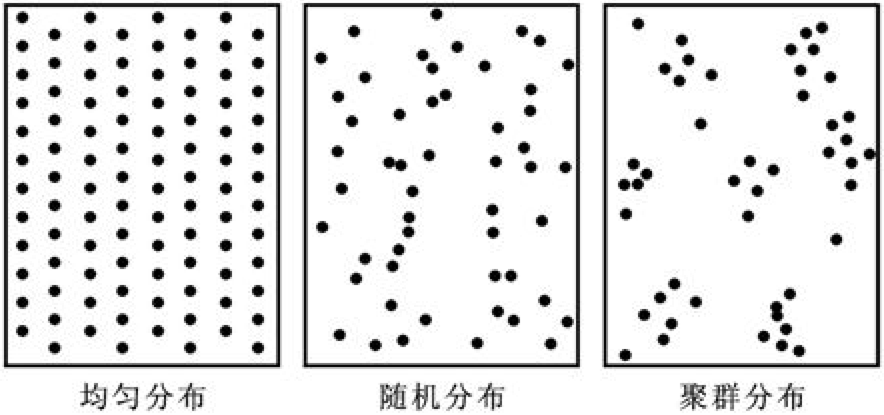
【声明】本视频和幻灯片为炼数成金网络课程的教学资料，所有资料只能在课程内使用，不得在课程以外范围散播，违者将可能被追究法律和经济责任。

课程详情访问炼数成金培训网站

http://edu.dataguru.cn

估计聚类趋势

- 聚类要求数据不能均匀分布
- 霍普金斯统计量：空间统计量，检验空间随机性



霍普金斯统计量

■ 计算步骤：韩家炜书第316页

（1）均匀地从 D 的空间中抽取 n 个点 $\boldsymbol{p}_1$，…，$\boldsymbol{p}_n$。也就是说，D 的空间中的每个点都以相同的概率包含在这个样本中。对于每个点 $\boldsymbol{p}_i(1 \leqslant i \leqslant n)$，我们找出 $\boldsymbol{p}_i$ 在 D 中的最近邻，并令 x_i 为 $\boldsymbol{p}_i$ 与它在 D 中的最近邻之间的距离，即

$$x_i = \min_{\boldsymbol{v} \in D} \{dist(\boldsymbol{p}_i, \boldsymbol{v})\} \tag{10.25}$$

（2）均匀地从 D 中抽取 n 个点 $\boldsymbol{q}_1$，…，$\boldsymbol{q}_n$。对于每个点 $\boldsymbol{q}_i(1 \leqslant i \leqslant n)$，我们找出 $\boldsymbol{q}_i$ 在 $D - \{\boldsymbol{q}_i\}$ 中的最近邻，并令 y_i 为 $\boldsymbol{q}_i$ 与它在 $D - \{\boldsymbol{q}_i\}$ 中的最近邻之间的距离，即

$$y_i = \min_{\boldsymbol{v} \in D, \boldsymbol{v} \neq \boldsymbol{q}_i} \{dist(\boldsymbol{q}_i, \boldsymbol{v})\} \tag{10.26}$$

（3）计算霍普金斯统计量 H

$$H = \frac{\sum_{i=1}^{n} y_i}{\sum_{i=1}^{n} x_i + \sum_{i=1}^{n} y_i} \tag{10.27}$$

相应解读

"霍普金斯统计量告诉我们数据集 D 有多大可能遵守数据空间的均匀分布吗?"如果 D 是均匀分布的，则 $\sum_{i=1}^{n} y_i$ 和 $\sum_{i=1}^{n} x_i$ 将会很接近，因而 H 大约为0.5。然而，如果 D 是高度倾斜的，则 $\sum_{i=1}^{n} y_i$ 将显著地小于 $\sum_{i=1}^{n} x_i$ ，因而 H 将接近于0。

我们的原假设是同质假设——D 是均匀分布的，因而不包含有意义的簇。非均匀假设（即 D 不是均匀分布的，因而包含簇）是备择假设。我们可以迭代地进行霍普金斯统计量检验，使用0.5作为拒绝备择假设阈值，即如果 $H > 0.5$，则 D 不大可能具有统计显著的簇。

簇数制定

- 经验判断，例如样本点数目为n，则取k=sqrt（n/2）
- 肘方法
- PSF或PST2这类统计量
- 信息论方法与信息准侧
- 交叉验证

肘方法

- 韩家炜书第317页

肘方法（elbow method）基于如下观察：增加簇数有助于降低每个簇的簇内方差之和。这是因为有更多的簇可以捕获更细的数据对象簇，簇中对象之间更为相似。然而，如果形成太多的簇，则降低簇内方差和的边缘效应可能下降，因为把一个凝聚的簇分裂成两个只引起簇内方差和的稍微降低。因此，一种选择正确的簇数的启发式方法是，使用簇内方差和关于簇数的曲线的拐点。

严格地说，给定 $k>0$，我们可以使用一种像 k-均值这样的算法对数据集聚类，并计算簇内方差和 $var(k)$。然后，我们绘制 var 关于 k 的曲线。曲线的第一个（或最显著的）拐点暗示“正确的”簇数。

PSF和PST2

- 在SAS中CLUSTER过程里被使用
- 可以先通过观察层次聚类时PSF和PST2的取值决定聚类簇数，再用来作kmeans

例 12-2-1　根据美国十城市之间的距离进行聚类。

1. **程序**

```
*ex12-2-1;
DATA mileages(TYPE=DISTANCE);
    INPUT(atlanta chicago denver houston losangel
    miami newyork sanfran seattle washdc)(5.)@51 city $15.;
CARDS;
 0                                                  ATLANTA
 587  0                                             CHICAGE
 1212 920  0                                        DENVER
 701  940  879  0                                   HOUSTON
 1936 1745 831  1374 0                              LOS ANGELES
 604  1188 1726 968  2339 0                         MIAMI
 748  713  1631 1420 2451 1092 0                    NEW YORK
 2139 1858 949  1645 347  2594 2571 0               SANFRANCISCO
 2182 1737 1021 1891 959  2734 2408 678  0          SEATTLE
 543  597  1494 1220 2300 923  205  2442 2329 0     WASHINGTON D.C
;
PROC CLUSTER DATA=mileages METHOD=AVERAGE PSEUDO;
    ID city;
PROC TREE;
RUN;
```

PSF和PST2

程序运行结果见图 12-1 和图 12-2。

①
The CLUSTER Procedure
Average Linkage Cluster Analysis

② Root-Mean-Square Distance Between Observations = 1580.242

Cluster History

③ NCL	④ Clusters Joined		⑤ FREQ	⑥ PSF	⑦ PST2	⑧ Norm RMS Dist	Tie
9	NEW YORK	WASHINGTON D.C	2	66.7	.	0.1297	
8	LOS ANGELES	SANFRANCISCO	2	39.2	.	0.2196	
7	ATLANTA	CHICAGE	2	21.7	.	0.3715	
6	CL7	CL9	4	14.5	3.4	0.4149	
5	CL8	SEATTLE	3	12.4	7.3	0.5255	
4	DENVER	HOUSTON	2	13.9	.	0.5562	
3	CL6	MIAMI	5	15.5	3.8	0.6185	
2	CL3	CL4	7	16.0	5.3	0.8005	
1	CL2	CL5	10	.	16.0	1.2967	

⑥ PSF 伪 F 值　在 $G=2$ 处 PSF 较大，分 2 类较好。

⑦ PST2 伪 t^2 值。在 $G=1$ 和 $G=5$ 处有峰值，由于最佳分类为它上面一种，故本例表明它支持 2 分类和 6 分类。

统计量的计算方法

- http://stat.smmu.edu.cn/field/sas07.htm
- PSF：伪F统计量
- PST2：伪T平方统计量

评估聚类质量

- 可以选择不同的方法，不同的簇数进行聚类，不同的选择可能导致聚类结果不尽相同。因此有必要对聚类效果、质量进行评估
- 外在方法：有基准可以使用
- 内在方法：没有基准
- 韩家炜书第318页

Bcubed

- 属于外在方法，oj是样本点，L(oj)代表基准

$$\text{Correctness}(o_i, o_j) = \begin{cases} 1 & \text{如果 } L(o_i) = L(o_j) \Leftrightarrow C(o_i) = C(o_j) \\ 0 & \text{其他} \end{cases}$$

BCubed 精度定义为

$$\text{Precision BCubed} = \frac{1}{n} \sum_{i=1}^{n} \frac{\sum_{o_j: i \neq j, C(o_i) = C(o_j)} \text{Correctness}(o_i, o_j)}{\| \{ o_j \mid i \neq j, C(o_i) = C(o_j) \} \|}$$

BCubed 召回率定义为

$$\text{Recall BCubed} = \frac{1}{n} \sum_{i=1}^{n} \frac{\sum_{o_j: i \neq j, L(o_i) = L(o_j)} \text{Correctness}(o_i, o_j)}{\| \{ o_j \mid i \neq j, L(o_i) = L(o_j) \} \|}$$

轮廓系数

- 内在方法

轮廓系数（silhouette coefficient）就是这种度量。对于 n 个对象的数据集 D，假设 D 被划分成 k 个簇 C_1，…，C_k。对于每个对象 $\boldsymbol{o} \in D$，我们计算 $\boldsymbol{o}$ 与 $\boldsymbol{o}$ 所属的簇的其他对象之间的平均距离 $a(\boldsymbol{o})$。类似地，$b(\boldsymbol{o})$ 是 $\boldsymbol{o}$ 到不属于 $\boldsymbol{o}$ 的所有簇的最小平均距离。假设 $\boldsymbol{o} \in C_i$ $(1 \leqslant i \leqslant k)$，则

$$a(\boldsymbol{o}) = \frac{\sum_{\boldsymbol{o}' \in C_i, \boldsymbol{o} \neq \boldsymbol{o}'} dist(\boldsymbol{o}, \boldsymbol{o}')}{|C_i| - 1} \tag{10.31}$$

而

$$b(\boldsymbol{o}) = \min_{C_j: 1 \leqslant j \leqslant k, j \neq i} \left\{ \frac{\sum_{\boldsymbol{o}' \in C_j} dist(\boldsymbol{o}, \boldsymbol{o}')}{|C_j|} \right\} \tag{10.32}$$

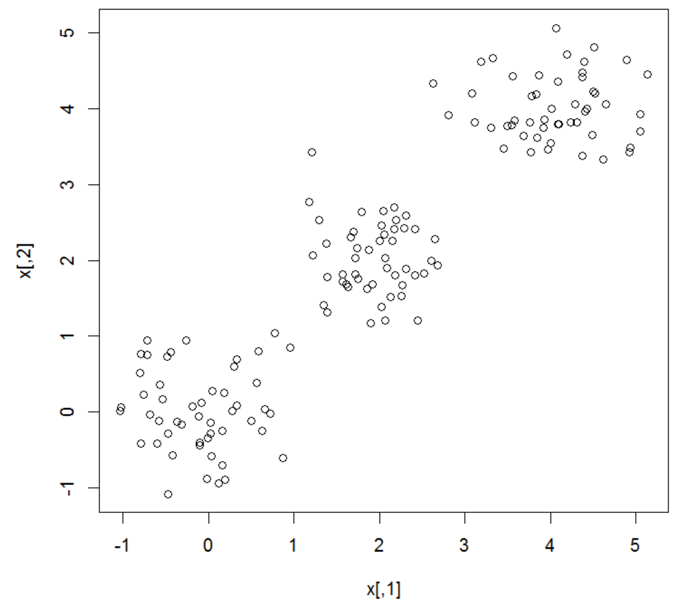
对象 $\boldsymbol{o}$ 的轮廓系数定义为

$$s(\boldsymbol{o}) = \frac{b(\boldsymbol{o}) - a(\boldsymbol{o})}{\max\{a(\boldsymbol{o}), b(\boldsymbol{o})\}} \tag{10.33}$$

R语言聚类评估

■ 安装clusterCrit包

```
# Create some spheric data around three distinct centers
x <- rbind(matrix(rnorm(100, mean = 0, sd = 0.5), ncol = 2),
matrix(rnorm(100, mean = 2, sd = 0.5), ncol = 2),
matrix(rnorm(100, mean = 4, sd = 0.5), ncol = 2))
vals <- vector()
for (k in 2:6) {
# Perform the kmeans algorithmclusterCrit 3
cl <- kmeans(x, k)
# Compute the Calinski_Harabasz index
vals <- c(vals,as.numeric(intCriteria(x,cl$cluster,"Calinski_Harabasz")))
}
idx <- bestCriterion(vals,"Calinski_Harabasz")
cat("Best index value is",vals[idx],"\n")
```



R语言聚类评估

```
> vals <- vector()
> for (k in 2:6) {
+ cl <- kmeans(x, k)
+ vals <- c(vals,as.numeric(intCriteria(x,cl$cluster,"Calinski_Harabasz")))
+ }
> vals
[1] 331.6777 841.5490 669.2320 579.4699 520.2916
> idx <- bestCriterion(vals,"Calinski_Harabasz")
> cat("Best index value is",vals[idx],"\n")
Best index value is 841.549
> 
```

实现内在评估方法

```
# Create some data
x <- rbind(matrix(rnorm(100, mean = 0, sd = 0.5), ncol = 2),
matrix(rnorm(100, mean = 1, sd = 0.5), ncol = 2),
matrix(rnorm(100, mean = 2, sd = 0.5), ncol = 2))
# Perform the kmeans algorithm
cl <- kmeans(x, 3)
# Compute all the internal indices
intCriteria(x,cl$cluster,"all")
# Compute some of them
intCriteria(x,cl$cluster,c("C_index","Calinski_Harabasz","Dunn"))
# The names are case insensitive and can be abbreviated
intCriteria(x,cl$cluster,c("det","cal","dav"))
```

基于概率模型的聚类

- 模糊簇与划分矩阵（韩家炜书，第325页）

表 11.2　评论和所用关键词的集合

评论 ID	关键词	评论 ID	关键词
R_1	数码相机、镜头	R_4	数码相机、镜头、计算机
R_2	数码相机	R_5	计算机、CPU
R_3	镜头	R_6	计算机、计算机游戏

我们可以把这些评论分成两个模糊簇 C_1 和 C_2。C_1 关于数码相机和镜头，而 C_2 关于计算机。划分矩阵是

$$M = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ \frac{2}{3} & \frac{1}{3} \\ 0 & 1 \\ 0 & 1 \end{bmatrix}$$

SSE

- 用于评估模糊簇聚类，度量模糊聚类对数据的拟合程度

对于对象 o_i，**误差的平方和**（SSE）由下式给出

$$SSE(o_i) = \sum_{j=1}^{k} w_{ij}^{p} dist(o_i, c_j)^2 \tag{11.2}$$

其中，参数 $p(p \geq 1)$ 控制隶属度的影响。p 的值越大，隶属度的影响越大。簇 C_j 的 SSE 是

$$SSE(C_j) = \sum_{i=1}^{n} w_{ij}^{p} dist(o_i, c_j)^2 \tag{11.3}$$

最后，聚类$\mathcal{C}$的 SSE 定义为

$$SSE(\mathcal{C}) = \sum_{i=1}^{n} \sum_{j=1}^{k} w_{ij}^{p} dist(o_i, c_j)^2 \tag{11.4}$$

概率簇的例子

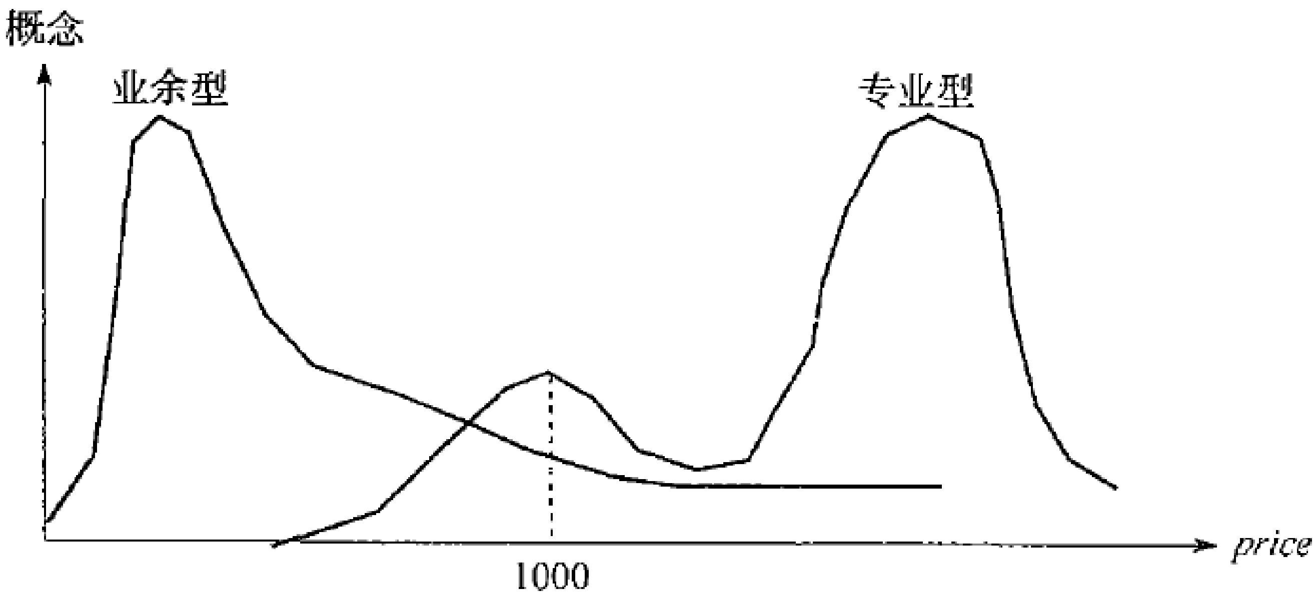


图 11.1 两个概率簇的概率密度函数

把聚类问题抽象出来

- 混合模型，数据集D的产生

（1）按照概率 ω_1，…，ω_k，选择一个簇 C_j。

（2）按照 C_j 的概率密度函数 f_j，选择一个 C_j 的实例。

该数据产生过程是混合模型的基本假定。**混合模型**假定观测对象集是来自多个概率簇的实例的混合。从概念上讲，每个观测对象都独立地由两步产生：首先，根据簇的概率选择一个概率簇；然后，根据选定簇的概率密度函数选择一个样本。

把聚类问题抽象出来

- 对于前一页幻灯片中的取样过程逆向思维：我们已经知道数据集D是按照上述方法取出，簇数k亦是已知，但概率簇（分布密度函数）未知，通过数据集D倒推出每个簇的分布密度（回忆贝叶斯信念网络的训练方法）

考虑 k 个概率簇 C_1，…，C_k 的集合 $\boldsymbol{C}$，k 个簇的概率密度函数分别为 f_1，…，f_k，而它们的概率分别为 ω_1，…，ω_k。对于对象 o，o 被簇 $C_j(1\leqslant j\leqslant k)$ 产生的概率为 $P(o\mid C_j)=\omega_j f_j(o)$。因此，$o$ 被簇的集合 $\boldsymbol{C}$ 产生的概率为

$$P(o\mid \boldsymbol{C})=\sum_{j=1}^{k}\omega_j f_j(o) \tag{11.5}$$

由于我们假定对象是独立地产生的，因此对于 n 个对象的数据集 $D=\{o_1,\cdots,o_n\}$，我们有

$$P(D\mid \boldsymbol{C})=\prod_{i=1}^{n}P(o_i\mid \boldsymbol{C})=\prod_{i=1}^{n}\sum_{j=1}^{k}\omega_j f_j(o_i) \tag{11.6}$$

现在，数据集 D 上的基于概率模型的聚类分析的任务是，找出 k 个概率簇的集合 $\boldsymbol{C}$，使得 $P(D\mid \boldsymbol{C})$ 最大化。最大化 $P(D\mid \boldsymbol{C})$ 通常是难处理的，因为通常来说，簇的概率密度函数可以取任意复杂的形式。为了使得基于概率模型的聚类是计算可行，我们通常折中，假定概率密度函数是一个参数分布。

基于概率模型的聚类分析

设 o_1，…，o_n 是 n 个观测对象，Θ_1，…，Θ_k 是 k 个分布的参数，分别令 $\boldsymbol{O}=\{o_1,\cdots,o_n\}$，$\boldsymbol{\Theta}=\{\Theta_1,\cdots,\Theta_k\}$。于是，对于任意对象 $o_i\in\boldsymbol{O}(1\leqslant i\leqslant n)$，（11.5）式可以改写为

$$P(o_i\mid\boldsymbol{\Theta})=\sum_{j=1}^{k}\omega_jP_j(o_i\mid\Theta_j)\tag{11.7}$$

其中，$P_j(o_i\mid\Theta_j)$ 是 o_i 使用参数 Θ_j，由第 j 个分布产生的概率。因此，（11.6）式可以改写为

$$P(\boldsymbol{O}\mid\boldsymbol{\Theta})=\prod_{i=1}^{n}\sum_{j=1}^{k}\omega_jP_j(o_i\mid\Theta_j)\tag{11.8}$$

使用参数概率分布模型，基于概率模型的聚类分析任务是推导出最大化（11.8）式的参数集 $\boldsymbol{\Theta}$。

最大似然估计

- http://zh.wikipedia.org/wiki/%E6%9C%80%E5%A4%A7%E4%BC%BC%E7%84%B6%E4%BC%B0%E8%AE%A1

给定一个概率分布D，假定其概率密度函数（连续分布）或概率质量函数（离散分布）为f_D，以及一个分布参数θ，我们可以从这个分布中抽出一个具有n个值的采样$X_1, X_2, \ldots, X_n$，通过利用f_D，我们就能计算出其概率：

$$\mathbb{P}(x_1, x_2, \ldots, x_n) = f_D(x_1, \ldots, x_n \mid \theta)$$

但是，我们可能不知道θ的值，尽管我们知道这些采样数据来自于分布D。那么我们如何才能估计出θ呢？一个自然的想法是从这个分布中抽出一个具有n个值的采样$X_1, X_2, ..., X_n$，然后用这些采样数据来估计θ.

一旦我们获得$X_1, X_2, \ldots, X_n$，我们就能从中找到一个关于θ的估计。最大似然估计会寻找关于θ的最可能的值（即，在所有**可能的θ取值中，寻找一个值使这个采样的“可能性”最大化）**。这种方法正好同一些其他的估计方法不同，如θ的非偏估计，非偏估计未必会输出一个最可能的值，而是会输出一个既不高估也不低估的θ值。

要在数学上实现最大似然估计法，我们首先要定义似然函数：

$$\text{lik}(\theta) = f_D(x_1, \ldots, x_n \mid \theta)$$

并且在θ的所有取值上，使这个函数最大化(一阶导数)。这个使可能性最大的$\widehat{\theta}$值即被称为θ的**最大似然估计**。

一般情况下的EM算法

- 最大期望算法（Expectation Maximization Algorithm，又译期望最大化算法），是一种迭代算法，用于含有隐变量（hidden variable）的概率参数模型的最大似然估计或极大后验概率估计。
- 最大期望算法经过两个步骤交替进行计算：

E步骤：估计未知参数的期望值，给出当前的参数估计。

M步骤：重新估计分布参数，以使得数据的似然性最大，给出未知变量的期望估计。

M 步上找到的参数估计值被用于下一个 E 步计算中，这个过程不断交替进行。重复直到收敛.

http://www.cnblogs.com/jerrylead/archive/2011/04/06/2006936.html

使用EM算法的模糊聚类

- 算例：韩家炜书第328页
- 在分布密度中，簇中心未知
- 首先任意定出簇中心，然后可以根据隶属度定义算出划分矩阵
- 根据划分矩阵又重新算出新的簇中心，使到在新的簇中心下SSE极小化
- 不断迭代直至收敛
- 可以看成是kmeans算法的推广

在 **E** - 步中，对于每个点，我们计算它属于每个簇的隶属度。对于任意点 o，我们分别以隶属权重

$$\frac{\dfrac{1}{dist(o,c_1)^2}}{\dfrac{1}{dist(o,c_1)^2}+\dfrac{1}{dist(o,c_2)^2}}=\frac{dist(o,c_2)^2}{dist(o,c_1)^2+dist(o,c_2)^2} \text{和} \frac{dist(o,c_1)^2}{dist(o,c_1)^2+dist(o,c_2)^2}$$

使用EM算法解决混合模型

例 11.8 对混合模型使用 EM 算法。给定数据对象集 $\boldsymbol{O}=\{o_1, \cdots, o_n\}$，我们希望挖掘参数集 $\Theta=\{\Theta_1, \cdots, \Theta_k\}$，使得（11.11）式的 $P(\boldsymbol{O} \mid \Theta)$ 最大化，其中 $\Theta_j=(\mu_j, \sigma_j)$ 分别是第 $j(1 \leqslant j \leqslant k)$ 个单变量高斯分布的均值和标准差。

我们可以使用 EM 算法。把随机值作为初值赋予参数 Θ，然后迭代地执行 E－步和 M－步，直到参数收敛或改变充分小。

在 E－步中，对于每个对象 $o_i \in \boldsymbol{O}(1 \leqslant i \leqslant n)$，我们计算 o_i 属于每个分布的概率，即

$$P(\Theta_j \mid o_i, \Theta)=\frac{P(o_i \mid \Theta_j)}{\sum_{l=1}^{k} P(o_i \mid \Theta_l)} \tag{11.13}$$

在 M－步中，我们调整参数 Θ，使得（11.11）式的 $P(\boldsymbol{O} \mid \Theta)$ 期望似然最大化。这可以通过设置

$$\mu_j=\frac{1}{k} \sum_{i=1}^{n} o_i \frac{P(\Theta_j \mid o_i, \Theta)}{\sum_{l=1}^{n} P(\Theta_j \mid o_l, \Theta)}=\frac{1}{k} \frac{\sum_{i=1}^{n} o_i P(\Theta_j \mid o_i, \Theta)}{\sum_{i=1}^{n} P(\Theta_j \mid o_i, \Theta)} \tag{11.14}$$

和

$$\sigma_j=\sqrt{\frac{\sum_{i=1}^{n} P(\Theta_j \mid o_i, \Theta)(o_i-u_j)^2}{\sum_{i=1}^{n} P(\Theta_j \mid o_i, \Theta)}} \tag{11.15}$$

来实现。 ■

离群值检测

- 又称为异常检测，孤立点检测等
- 什么是离群值？离群值**是一个观测值，它与其它观测值的差别如此之大，以至于怀疑它是由不同的机制产生的**
- 离群值的一些场景

1 网站日志中的离群值，试图入侵者

2 一群学生中的离群值，天才 or 白痴？

3 天气数据，灾害，极端天气

4 信用卡行为，试图欺诈者

5 低概率事件，接种疫苗后却发病的

6 实验误差或仪器和操作问题造成的错误数据

等等

离群点分析场景：信用卡诈骗

离群点分析场景：黑客攻击

```
xmenu=1&inajax=1" "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; .NET CLR 1.1.4322; .NET CLR 2.0.50727; .NET CLR 3.0.045
06.30)"
183.3.51.76 - - [29/Nov/2013:01:27:25 +0800] "GET /member.php?mod=logging&action=login HTTP/1.1" 200 17707 "http://r.dataguru
.cn/member.php?mod=logging&action=login" "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; .NET CLR 1.1.4322; .NET CLR 2.0.
50727; .NET CLR 3.0.04506.30)"
183.3.51.76 - - [29/Nov/2013:01:27:26 +0800] "GET /member.php?mod=logging&action=login HTTP/1.1" 200 17707 "http://r.dataguru
.cn/member.php?mod=logging&action=login" "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; .NET CLR 1.1.4322; .NET CLR 2.0.
50727; .NET CLR 3.0.04506.30)"
183.3.51.76 - - [29/Nov/2013:01:27:26 +0800] "POST /member.php?mod=logging&action=login&loginsubmit=yes&inajax=1&ajaxmenu=1 H
TTP/1.1" 200 297 "http://r.dataguru.cn/member.php?mod=logging&action=login&loginsubmit=yes&inajax=1&ajaxmenu=1" "Mozilla/4.0
(compatible; MSIE 7.0; Windows NT 5.1; .NET CLR 1.1.4322; .NET CLR 2.0.50727; .NET CLR 3.0.04506.30)"
66.249.64.1 - - [29/Nov/2013:01:30:19 +0800] "GET /home.php?mod=space&uid=50144&do=home&view=me&from=space HTTP/1.1" 200 5769
 "-" "Mozilla/5.0 (iPhone; CPU iPhone OS 6_0 like Mac OS X) AppleWebKit/536.26 (KHTML, like Gecko) Version/6.0 Mobile/10A5376
e Safari/8536.25 (compatible; Googlebot-Mobile/2.1; +http://www.google.com/bot.html)"
66.249.64.8 - - [29/Nov/2013:01:30:44 +0800] "GET /space-uid-73446.html HTTP/1.1" 200 4782 "-" "Mozilla/5.0 (compatible; Goog
lebot/2.1; +http://www.google.com/bot.html)"
210.51.177.136 - - [29/Nov/2013:01:35:28 +0800] "GET / HTTP/1.0" 200 46531 "-" "User-Agent: Mozilla/5.0 (compatible; MSIE 6.0
;Windows XP)"
66.249.64.1 - - [29/Nov/2013:01:36:52 +0800] "GET /space-uid-73384.html HTTP/1.1" 200 4776 "-" "Mozilla/5.0 (compatible; Goog
lebot/2.1; +http://www.google.com/bot.html)"
66.249.64.1 - - [29/Nov/2013:01:38:25 +0800] "GET /space-uid-73345.html HTTP/1.1" 200 4434 "-" "Mozilla/5.0 (compatible; Goog
lebot/2.1; +http://www.google.com/bot.html)"
183.3.20.129 - - [29/Nov/2013:01:38:45 +0800] "GET /member.php?mod=logging&action=login HTTP/1.1" 200 17707 "http://r.datagur
u.cn/member.php?mod=logging&action=login" "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; .NET CLR 1.1.4322; .NET CLR 2.0
.50727; .NET CLR 3.0.04506.30)"
183.3.20.129 - - [29/Nov/2013:01:38:49 +0800] "GET /member.php?mod=logging&action=login HTTP/1.1" 200 17707 "http://r.datagur
u.cn/member.php?mod=logging&action=login" "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; .NET CLR 1.1.4322; .NET CLR 2.0
.50727; .NET CLR 3.0.04506.30)"
183.3.20.129 - - [29/Nov/2013:01:38:49 +0800] "POST /member.php?mod=logging&action=login&loginsubmit=yes&inajax=1&ajaxmenu=1
HTTP/1.1" 200 297 "http://r.dataguru.cn/member.php?mod=logging&action=login&loginsubmit=yes&inajax=1&ajaxmenu=1" "Mozilla/4.0
 (compatible; MSIE 7.0; Windows NT 5.1; .NET CLR 1.1.4322; .NET CLR 2.0.50727; .NET CLR 3.0.04506.30)"
[root@class2room web_logs]#
```

检测离群值的方法

- 基于统计学的方法
- 基于邻近性的方法
- 基于聚类的方法

统计学方法：正态分布下检出一元离群值

- 韩家炜书第357页例子

$$\hat{\mu} = \frac{24.0 + 28.9 + 28.9 + 29.0 + 29.1 + 29.1 + 29.2 + 29.2 + 29.3 + 29.4}{10} = 28.61$$

$$\begin{aligned}\hat{\sigma}^2 = {} & ((24.1 - 28.61)^2 + (28.9 - 28.61)^2 + (28.9 - 28.61)^2 + (29.0 - 28.61)^2 \\ & + (29.1 - 28.61)^2 + (29.1 - 28.61)^2 + (29.2 - 28.61)^2 + (29.2 - 28.61)^2 \\ & + (29.3 - 28.61)^2 + (29.4 - 28.61)^2)/10 \approx 2.29\end{aligned}$$

由此，有 $\hat{\sigma} = \sqrt{2.29} = 1.51$。

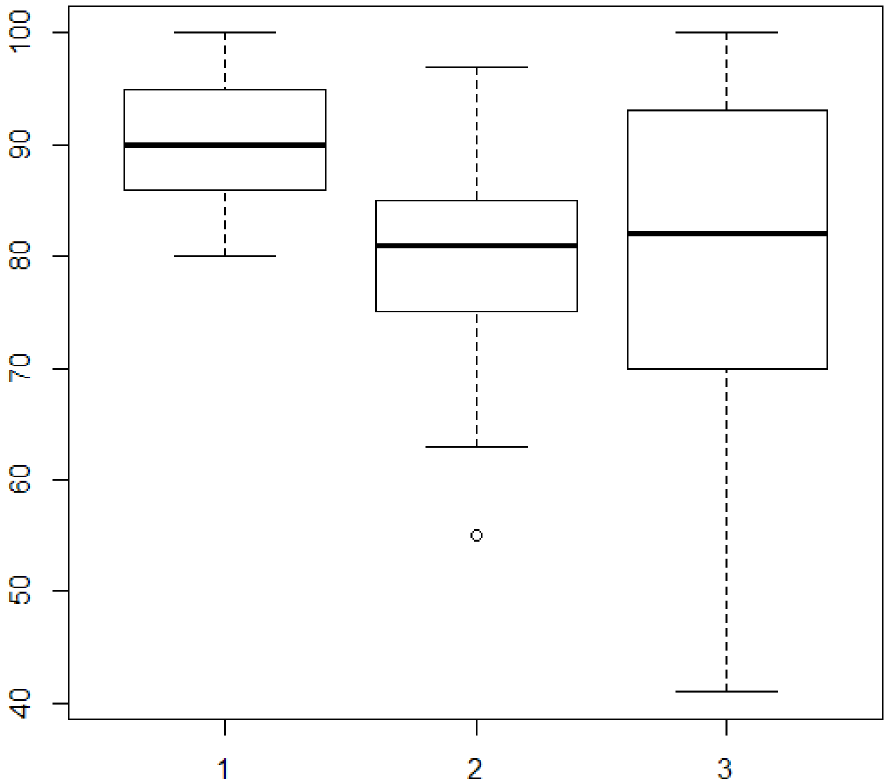
最大偏离值为24.0℃，偏离估计的均值4.61℃。在正态分布的假定下，区域 $\mu \pm 3\sigma$ 包含99.7%的数据。由于 $\frac{4.61}{1.51} = 3.04 > 3$，24.0℃被该正态分布产生的概率小于0.15%，因此它被识别为离群点。

■ ● 离群点

箱型图识别离群值

- 箱子的上下横线为样本的25%和75%分位数
- 箱子中间的横线为样本的中位数
- 上下延伸的直线称为尾线，尾线的尽头为最高值和最低值
- **离群值标示**

```
> boxplot(x$x1,x$x2,x$x3)
> 
```



一元离群值的Grubb检验

另一种使用正态分布的一元离群点检测的统计学方法是 *Grubb* 检验（又称为最大标准残差检验）。对于数据集中的每个对象 x，定义 z 分数（z-score）为

$$z = \frac{|x - \bar{x}|}{s} \tag{12.4}$$

其中，$\bar{x}$ 是输入数据的均值，s 是标准差。对象 x 是离群点，如果

$$z \geqslant \frac{N-1}{\sqrt{n}} \sqrt{\frac{t^2_{\alpha/(2N),N-2}}{N-2+t^2_{\alpha/(2N),N-2}}} \tag{12.5}$$

其中，$t^2_{\alpha/(2N),N-2}$是显著水平 $\alpha/(2N)$ 下的 t－分布的值，N 是数据集中的对象数。

多元离群值检测

- 思路：利用马氏距离将多元转化为一元情形处理

例 12.9 使用马哈拉诺比斯距离检测多元离群点。对于一个多元数据集，设 $\bar{\boldsymbol{o}}$ 为均值向量。对于数据集中的对象 $\boldsymbol{o}$，从 $\boldsymbol{o}$ 到 $\bar{\boldsymbol{o}}$ 的马哈拉诺比斯（Mahalanobis）距离为

$$MDist(\boldsymbol{o},\bar{\boldsymbol{o}}) = (\boldsymbol{o}-\bar{\boldsymbol{o}})^T S^{-1}(\boldsymbol{o}-\bar{\boldsymbol{o}}) \tag{12.6}$$

其中 S 是协方差矩阵。

$MDist(\boldsymbol{o},\bar{\boldsymbol{o}})$ 是一元变量，于是可以对它进行 Grubb 检验。因此，可以按如下方法对多元离群点检测任务进行变换：

（1）计算多元数据集的均值向量。

（2）对于每个对象 $\boldsymbol{o}$，计算从 $\boldsymbol{o}$ 到 $\bar{\boldsymbol{o}}$ 的马哈拉诺比斯距离 $MDist(\boldsymbol{o},\bar{\boldsymbol{o}})$。

（3）在变换后的一元数据集 $\{MDist(\boldsymbol{o},\bar{\boldsymbol{o}}) \mid \boldsymbol{o}\in D\}$ 中检测离群点。

（4）如果 $MDist(\boldsymbol{o},\bar{\boldsymbol{o}})$ 被确定为离群点，则 $\boldsymbol{o}$ 也被视为离群点。 ■

可能存在多个簇的复杂情形

- 首先用EM算法计算出簇的具体表示
- 不属于任何簇的样本点判为离群点

例 12.11　使用混合参数分布检测多元离群点。考虑图 12.4 中的数据，其中有两个大簇 C_1 和 C_2。这里，假定数据由一个正态分布产生效果不好。估计的均值落在这两个簇之间，而不是任何一个簇的内部。这两个簇之间的对象不可能被检测为离群点，因为它们离均值很近。■

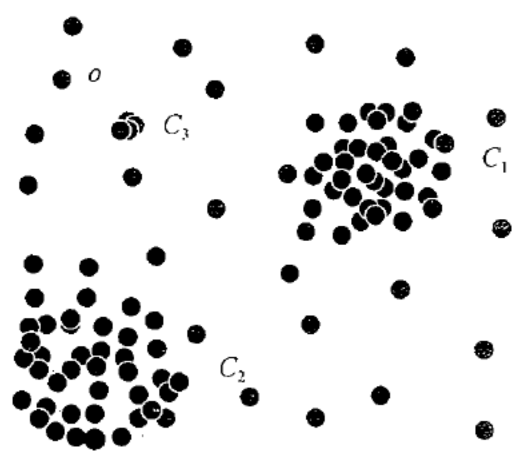


图 12.4　一个复杂的数据集

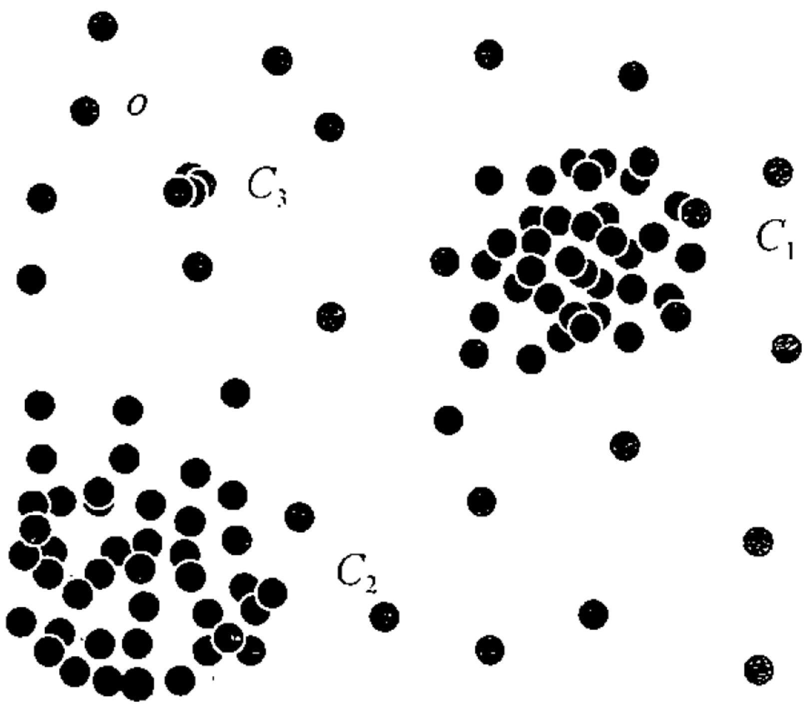
为了克服这一困难，假定正常的数据对象被多个正态分布产生（这里是两个）。也就是说，假定两个正态分布 $\Theta_1(\mu_1, \sigma_1)$ 和 $\Theta_2(\mu_2, \sigma_2)$。对于数据集中的任意对象 o，o 被这两个分布产生的概率为

$$Pr(o \mid \Theta_1, \Theta_2) = f_{\Theta_1}(o) + f_{\Theta_2}(o)$$

其中，f_{Θ_1} 和 f_{Θ_2} 分别是 Θ_1 和 Θ_2 的概率密度函数。可以使用期望最大化（EM）算法（第 11 章），由该数据学习参数 μ_1，σ_1，μ_2，σ_2，就像用混合模型聚类所做的那样。每个簇都用学习得到的正态分布表示。一个对象 o 被检测为离群点，如果它不属于任何簇，即它被这两个分布的组合产生的概率很低。

离群值页可能组成簇

- 注意下图中的C3簇



直方图方法

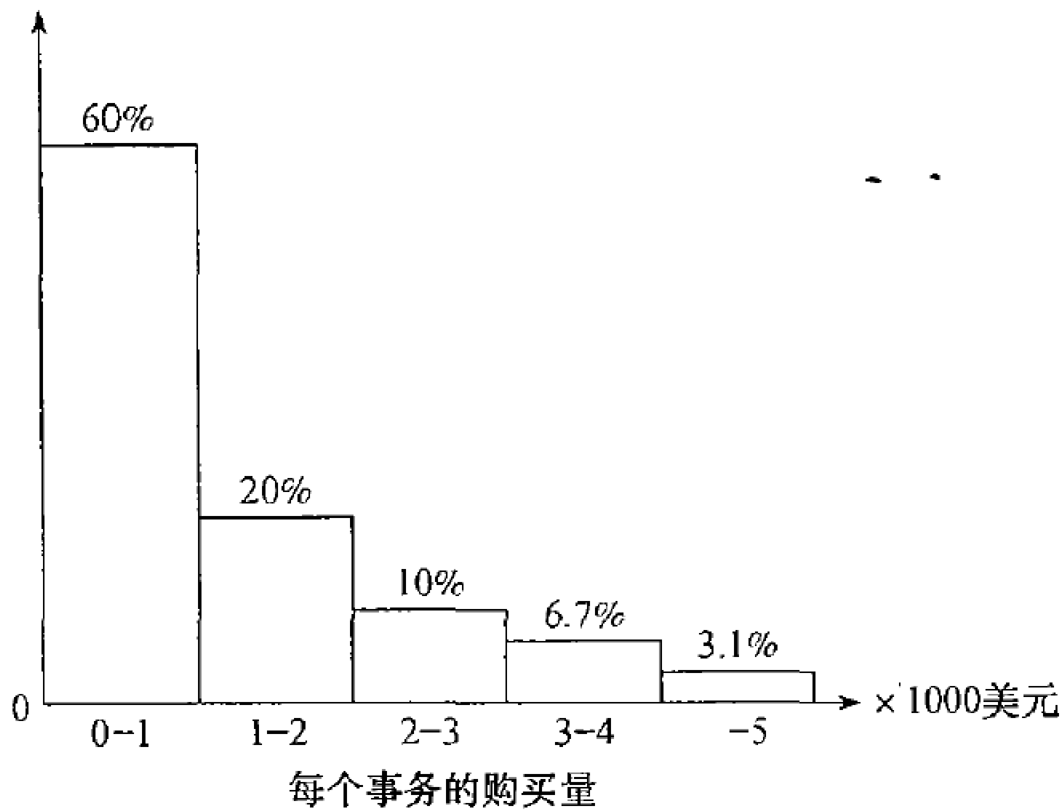


图 12.5　每个事务的购买量的直方图

基于邻域判断离群点

DATAGURU炼数成金

$$\frac{\| \{o' \mid dist(o,o') \leqslant r\} \|}{\| D \|} \leqslant \pi$$

算法：基于距离的离群点检测。

输入：

- 对象集$D=\{o_1,\cdots,o_n\}$，阈值r（$r>0$）和π（$0<\pi\leqslant 1$）。

输出：D中的DB（r, π）-离群点。

方法：

```
for i=1 to n do
    count←0
    for j=1 to n do
        if i≠j and dist(o_i, o_j) ≤r then
            count←count+1
            if count≥π·n then
                exit{o_i不可能是DB(r, π)-离群点}
            end if
        end if
    end for
    print o_i{根据（12.10）式，o_i是DB(r, π)-离群点}
end for;
```

基于网格的方法

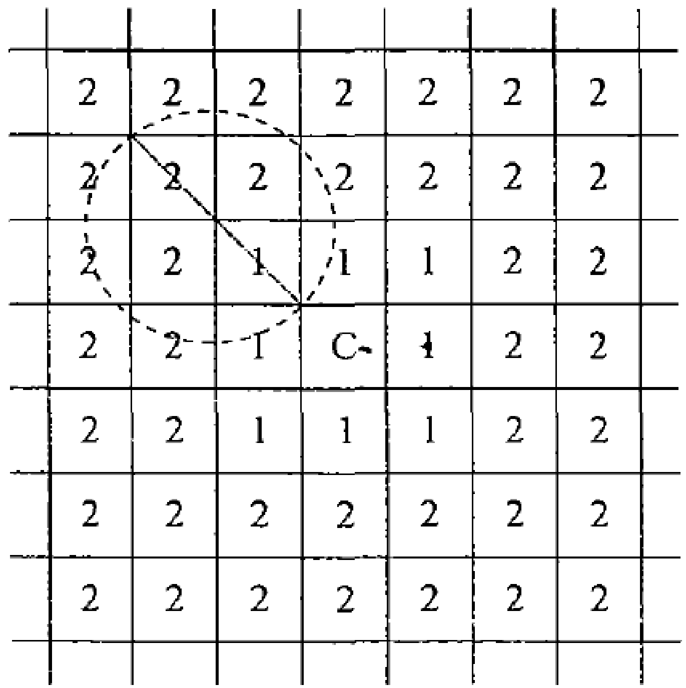


图 12.7 CELL 方法的网格

基于聚类的方法

- 韩家炜书第366页
  - 该对象属于某个簇吗？如果不，则它被识别为离群点。
  - 该对象与最近的簇之间的距离很远吗？如果是，则它是离群点。
  - 该对象是小簇或稀疏簇的一部分吗？如果是，则该簇中的所有对象都是离群点。

炼数成金逆向收费式网络课程

- Dataguru（炼数成金）是专业数据分析网站，提供教育，媒体，内容，社区，出版，数据分析业务等服务。我们的课程采用新兴的互联网教育形式，独创地发展了逆向收费式网络培训课程模式。既继承传统教育重学习氛围，重竞争压力的特点，同时又发挥互联网的威力打破时空限制，把天南地北志同道合的朋友组织在一起交流学习，使到原先孤立的学习个体组合成有组织的探索力量。并且把原先动辄成千上万的学习成本，直线下降至百元范围，造福大众。我们的目标是：低成本传播高价值知识，构架中国第一的网上知识流转阵地。
- 关于逆向收费式网络的详情，请看我们的培训网站 http://edu.dataguru.cn

Thanks

FAQ时间