第 5 章

聚　　类

我们来回顾一下联系人的数据集，假设要组织晚餐来介绍你的一些朋友，并决定在每次见面时，邀请的人都要具有相同受教育程度和年龄。为此，收集如表 5.1 所示的有关朋友的数据。受教育程度是一个数值属性，可以取 1(小学)、2(高中)、3(大学生)、4(大学毕业生)或 5(研究生)的值。十进制数字表示给定的级别正在进行，例如，4.5 表示正在上研究生课程的人。

表 5.1　简单社交网络数据集

姓　　名	年龄	受教育程度
Andrew (A)	55	1
Bernhard (B)	43	2
Carolina (C)	37	5
Dennis (D)	82	3
Eve (E)	23	3.2
Fred (F)	46	5
Gwyneth (G)	38	4.2
Hayden (H)	50	4
Irene (I)	29	4.5
James (J)	42	4.1

在前几章中，我们研究了一些可以用于联系人数据的简单数据分析器。虽然这些分析器为我们提供了有趣和有用的信息，但是我们可以通过使用其他更复杂的数据挖掘技术获得更有趣的分析结果。其中一个可以在联系人中找到年龄和受教育程度相似的朋友群体。

我们在前面的章节中已经看到，数据分析建模任务可以大致分为描述性任务和预测性任务。本章将介绍用于描述性任务的一系列重要技术，它们可以通过对数据集进行分区来描述数据集，从而使同一组中的对象彼此相似。这些“聚类”技术已经被开发出来了，并广泛用于将数据集划分为组。聚类技术只使用预测属性定义分区，因此它们是非监督的技术。图 5.1 给出了一个示例，应用聚类技术将 10 人的数据集生成两个聚类。

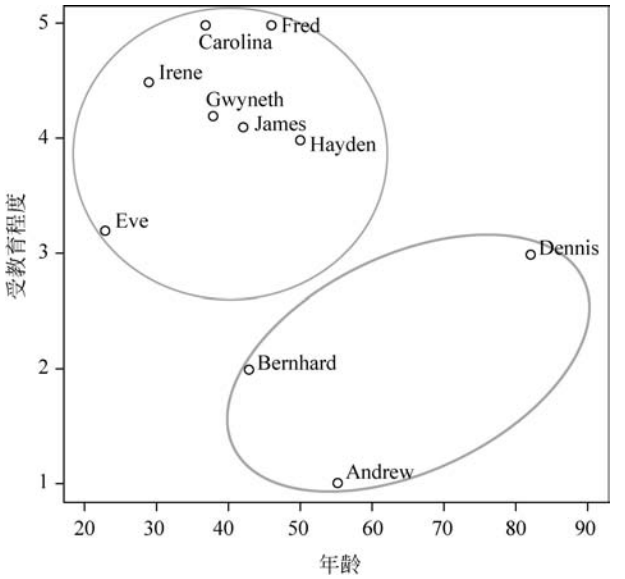


图 5.1 在数据集中使用聚类

聚类的数量通常不会预先定义，而是通过反复实验发现的，使用人工反馈或聚类验证措施找到合适数量的聚类，并将一个数据集划分到这些聚类中。图 5.2 所示为将之前的数据集分到 3 个聚类中。

聚类的数量越大，聚类内的对象可能就越相似。限制在于要将聚类的数量定义为与对象相等的数量，因此每个聚类只有一个对象。

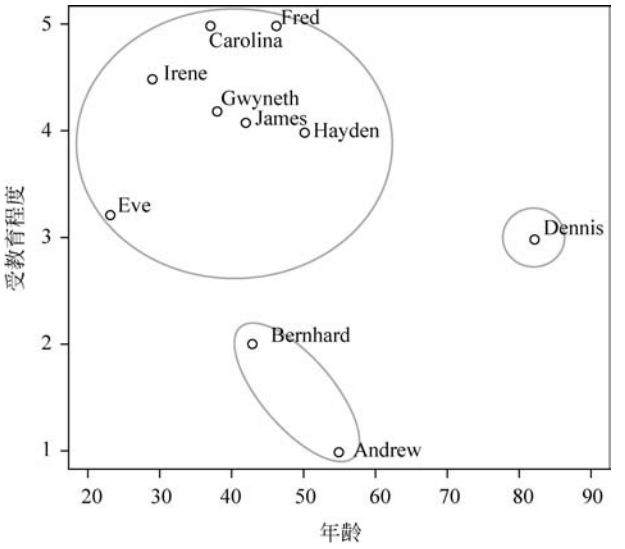


图 5.2 数据集的另一种聚类

5.1 距离度量

在定义相似数据组之前，我们必须就什么是相似的、什么是不相似的（不同的）达成一致。可以用一个数字表示两个事物之间的相似度，也可以说，对某个特定的对象，同一数据集中哪些其他对象更相似，哪些更不相似。将数字与两个对象之间的相似性（和差异性）联系起来的一种常见方法是使用距离度量，最相似的事物之间的距离最小，而最不相似的事物之间的距离则最大。计算事物间距离的方法取决于其属性的尺度类型——它们是定量的还是定性的。

5.1.1 常见属性类型值之间的差异

对于具有相同属性的两个值 a 和 b 之间的差值，可以将其记为 $d(a,b)$。对于定量属性，可以计算绝对差值为

$$d(a,b)=|a-b| \tag{5.1}$$

例 5.1 例如，Andrew（$a=55$）和 Carolina（$b=37$）的年龄差为 $|55-37|=18$。注意，即使改变值的顺序（$a=37, b=55$），结果也是一样的。

如果属性类型是定性的，可以使用适合给定类型的距离度量；如果定性属性有序数值，可以测量它们位置的差异为

$$d(a,b)=(|\mathrm{pos}_a-\mathrm{pos}_b|)/(n-1) \tag{5.2}$$

其中，n 为不同值的个数；pos_a 和 pos_b 分别为 a 和 b 在可能值排序中的位置。

例 5.2 在数据集中，受教育程度可以被认为是一个序数属性，值越大意味着受教育程度越高。因此，Andrew 和 Carolina 的受教育程度之间的差距为

$$|\mathrm{pos}_1-\mathrm{pos}_5|/4=|1-5|/4=1$$

注意，序数属性不是仅能用数字表示。例如，受教育程度可以有“小学”“高中”“大学生”“大学毕业生”和“研究生”等数值，不过这些数据可以很容易地转换为数字（见 2.1 节）。

如果一个定性属性有名义值，为了计算两个值之间的距离，只须确定它们是否相等（差值为 0）或不相等（差值为 1）。

$$d(a,b)=\begin{cases}1, & a\neq b\\ 0, & a=b\end{cases} \tag{5.3}$$

例 5.3 例如，因为 Andrew 和 Carolina 的名字不同，所以距离为 1。

一般来说，计算两个对象之间的距离包括聚合它们相应属性之间的距离（通常是差值）。对于我们的数据集，这意味着将两人的年龄和受教育程度的差聚合起来。

最常见的属性类型是具有数值的定量属性以及布尔、字或代码形式的定性属性（顺序和名义的）。数据集的定性属性可以转化为定量属性（见第 2 章），从而用一个数字向量表示数据集中的每个对象（数据表中的行）。

一个由 m 个数量属性向量表示的对象可以映射到 m 维空间。对于简化的数据集，有

“年龄”和“受教育程度”两个属性，如图 5.1 所示，每个人都可以映射到二维空间中的一个对象。两个对象越相似，它们在这个空间中的映射就越接近。现在让我们看看具有定量属性的对象的距离度量。

5.1.2 定量属性对象的距离度量

有些距离度量是闵可夫斯基距离(Minkowski Distance)的特殊情况。具有定量属性的二维对象 p 和 q 的闵可夫斯基距离为

$$d(p,q)=\sqrt[r]{\sum_{k=1}^{m}|p_k-q_k|^r} \tag{5.4}$$

其中，m 为属性个数；p_k 和 q_k 分别为对象 p 和 q 的第 k 个属性值。变量是通过不同的 r 值获得的。例如，对于曼哈顿距离，$r=1$；对于欧氏距离，$r=2$。

曼哈顿距离也称为街区或出租车距离，因为如果你在一个城市，想要从一个地方到另一个地方，它会测量沿着街道走过的距离。对于那些知道毕达哥拉斯定理的人，可能会对欧氏距离比较熟悉，该定理测量直角三角形最长边的长度。

图 5.3 展示了欧氏距离和曼哈顿距离的区别，欧氏距离为 7.07，用对角线表示，另一条线则是曼哈顿距离，长度为 10。

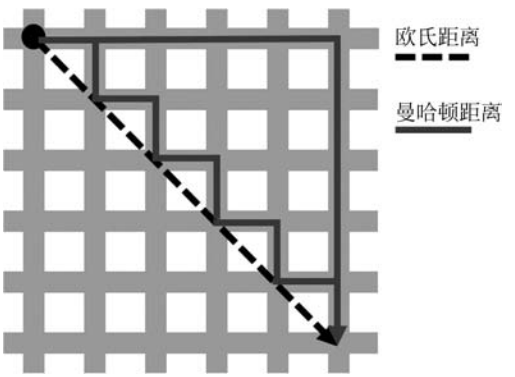


图 5.3 欧氏距离和曼哈顿距离

还有一些属性类型既不是定量的，也不是定性的，但是在数据挖掘中经常遇到。这些属性类型，称为非常规属性，其中包括：

- 生物序列
- 时间序列
- 图像
- 声音
- 视频

所有这些非常规的属性类型都可以转换为定量或定性属性。在这些非常规的属性类型中，最常见的是序列(文本、生物和时间序列)和图像。接下来讨论序列的距离度量问题。

5.1.3 非常规属性的距离度量

汉明距离可用于数值序列，这些值通常是字符或二进制值。二进制值(或二进制数)为1或0，表示真或假。汉明距离是两个字符串中对应字符或符号不同的位置数。

例如，James和Jimmy之间的汉明距离为3，Tom和Tim之间的汉明距离为1。请注意，汉明距离是曼哈顿距离的一个特殊情况，此时的属性只能假定为二进制值(0或1)。

对于可以具有不同大小的短序列，可以使用序列距离度量，如莱文斯坦距离，有时也称为编辑距离。编辑距离度量将一个序列转换为另一个序列所需的最小操作数量，可能的操作包括插入(一个字符)、删除(一个字符)和替换(另一个字符)。

例5.4 字符串Johnny和Jonston之间的编辑距离为5，因为需要将字符h、n、n、y替换为n、s、t、o(4种操作)，并在末尾添加一个字符n(第5种操作)。在生物信息学中也使用了类似的方法比较DNA、RNA和氨基酸序列。

对于长文本，如描述产品或故事的文本，可以使用称为“单词包”的方法将每个文本转换为整数向量。单词包方法首先提取一个单词列表，其中包含要挖掘的文本中出现的所有单词。每个文本被转换成一个定量向量，每个位置都与找到的一个单词相关，其值是该单词在文本中出现的次数。

例5.5 例如下面的两句话：

A=I will go to the party. But first, I will have to work.

B=They have to go to the work by bus.

这些单词生成的向量如表5.2所示。

表5.2 单词包向量示例

语句	I	will	go	to	the	party	but	first	have	work	they	by	bus
A	2	2	1	2	1	1	1	1	1	1	0	0	0
B	0	0	1	2	1	0	0	0	1	1	1	1	1

现在，欧氏距离可以计算出这些13维的定量向量，测量它们之间的距离。本书将在第13章中讨论单词包等其他技术，可以将文本(或文档)转换为定量向量。

其他常见的序列有时间序列(如医学领域的ECG或EEG数据)。为了计算两个时间序列之间的距离，一个常见的选择是动态时间规整(Dynamic Time Warping，DTW)距离，它类似于编辑距离。考虑到时间和速度的变化，它返回两个时间序列之间的最佳匹配值。图5.4举例说明了两个时间序列之间的最佳一致性。

要计算图像之间的距离，可以使用两种不同的方法。首先，可以从图像中提取与应用相关的特征。例如，对于人脸识别任务，可以提取眼睛之间的距离。每幅图像都由一个实数向量表示，其中每个元素对应一个特定的特征。在第2种方法中，首先将每幅图像转换为像素矩阵，其中矩阵的大小与图像所需的粒度相关联，然后可以将每个像素转换为一个整数值。

例如，对于黑色和白色的图像，像素越深，值越大。这个矩阵可以变换成向量。在这两种情况下，距离度量可以与定量属性值的向量相同。

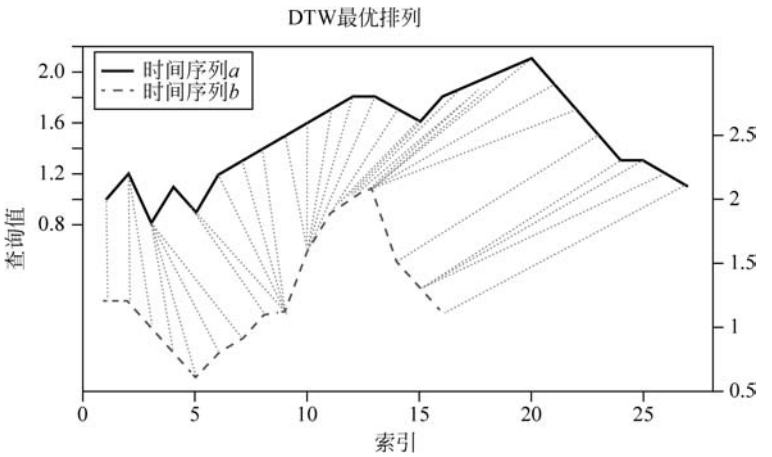
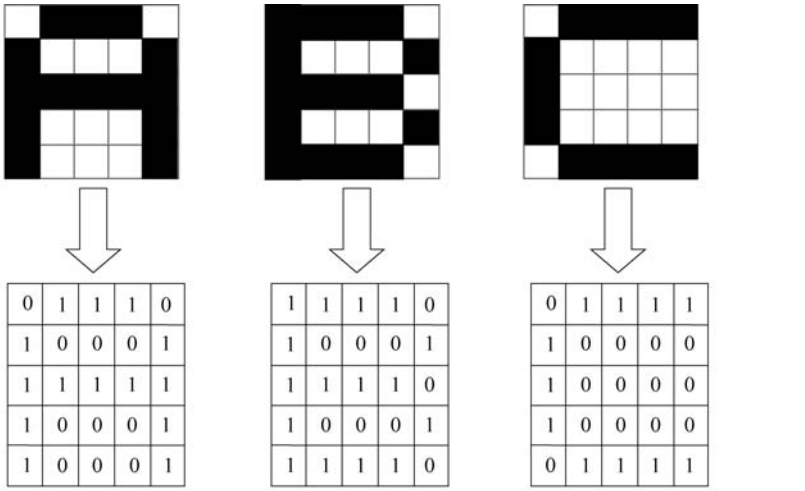


图 5.4 动态时间规整

例 5.6 图 5.5 中的图像代表了大小为 5×5 像素的画布上的字母 A、B 和 C，由于图像是黑白的(两种颜色)，所以每个图像都可以用一个 5 行 5 列的二进制矩阵或一个大小为 25 的



	第1行					第2行					第3行					第4行					第5行				
A	0	1	1	1	0	1	0	0	0	1	1	1	1	1	1	1	0	0	0	1	1	0	0	0	1
B	1	1	1	1	0	1	0	0	0	1	1	1	1	1	0	1	0	0	0	1	1	1	1	1	0
C	0	1	1	1	1	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	1	1	1	1

图 5.5 大小为 5×5 像素的图像转换为矩阵和向量

二进制向量表示，这样就可以一个接一个地复制矩阵的行，换句话说，就是把每幅图像映射到一个 25 维的空间。我们现在可以使用任意的距离度量计算这些图像之间的差异。图像 A 和 B 之间的曼哈顿距离为 6，图像 A 和 C 之间的曼哈顿距离是 11，图像 B 和 C 之间的曼哈顿距离为 9。需要注意的是，对于二进制值（只有 0 或 1）的情形，如果 0 和 1 表示字符或字母，则曼哈顿距离与编辑距离相等。

对于声音或视频等其他类型的数据，和图像类似，很容易通过提取特征的方法转换成定量向量序列。对于声音，它可能是一些低电平的信号参数，如带宽或音高，视频则可以看作是一系列的图像和声音。

5.2 聚类验证

要为数据集找到好的聚类划分，不管使用什么聚类算法，都必须评估划分的质量。与分类任务相比，对给定数据集的最佳聚类算法的识别缺乏明确的定义。

对于分类这种预测任务，预测模型的评估有一个明确的意义，也就是预测模型分类对象的效果如何。对于聚类划分，情况就不一样了，因为好的划分有很多定义，但在聚类任务中经常使用一些验证措施。

下面提出了几种聚类有效性准则，其中一些是自动的，另一些则需要专家输入。聚类划分评价的自动验证措施大致可分为 3 类。

（1）外部度量：使用外部信息（如类标签）定义给定划分中的聚类质量。两个最常见的外部度量是 RAND 和 Jaccard。

（2）内部度量：寻找每个聚类内部和/或不同聚类之间的离散性。两个最常见的内部度量是轮廓指数（同时度量紧密度和分离度）和组内平方和（只度量紧密度）。

（3）相对度量：比较由两个或多个聚类技术或同一技术的多次运行找到的划分。

接下来讨论轮廓指数、组内平方和这两个内部指数，以及 Jaccard 外部指数。

1. 轮廓指数

评估每个聚类的紧密度，并度量：

（1）和聚类内的其他对象之间的距离；

（2）不同聚类的分散度，每个聚类中的对象到另一个聚类中最近的对象的距离。

为此，它对每个对象 x 应用以下方程。

$$s(x_i)=\begin{cases}1-a(x_i)/b(x_i), & a(x_i)<b(x_i)\\ 0, & a(x_i)=b(x_i)\\ b(x_i)/a(x_i)-1, & a(x_i)>b(x_i)\end{cases} \tag{5.5}$$

其中，$a(x_i)$为 x_i 与聚类中所有其他对象之间的平均距离；$b(x_i)$为 x_i 与每个其他聚类中所有其他对象之间的最小平均距离。

所有 $s(x_i)$的平均值给出了划分轮廓的测量值。

2. 组内平方和

这也是一个内部度量，但只测量紧密度。它对每个实例和其聚类的质心之间的欧氏距离的平方求和。由式(5.4)可知，具有 m 个属性的两个实例 p 和 q 之间的欧氏距离的平方为

$$\mathrm{sed}(p,q)=\sum_{k=1}^{m}|p_k-q_k|^2 \tag{5.6}$$

组内平方和由式(5.7)得出。

$$s=\sum_{i=1}^{K}\sum_{j=1}^{J_i}\mathrm{sed}(p_j,C_i) \tag{5.7}$$

其中，K 为聚类的数量；J_i 为聚类 i 的实例数量；C_i 为聚类 i 的质心。

3. Jaccard 指数

这是一种在分类任务中使用的类似度量的变体，它评估每个聚类中对象的分布相对于类标签的均匀程度，计算式如下。

$$J=M_{11}/(M_{01}+M_{10}+M_{11}) \tag{5.8}$$

其中，M_{01} 为其他聚类中对象的数量，但标签相同；M_{10} 为同一聚类对象的数量，但标签不同；M_{00} 为其他聚类中对象的数量，且标签不同；M_{11} 为同一聚类中对象的数量，且标签相同。

5.3 聚类技术

由于我们已经知道如何度量记录、图像和单词之间的相似性，因此可以继续研究如何使用聚类技术得到组/聚类。聚类技术有数百种，可以用不同的方式进行分类。其中之一是如何创建划分，它定义了如何将数据分为组。大多数技术在一个步骤中定义划分(划分聚类)，而其他技术则逐步定义，增加或减少聚类的数量(层次聚类)。图 5.6 列出了划分聚类和层次聚类的区别。此外，在许多技术中，每个对象必须属于一个聚类(CRISP 聚类)，但是对于其他对象，每个对象可以不同程度地属于几个聚类(范围从 0%(不属于)到 100%(完全属于))。

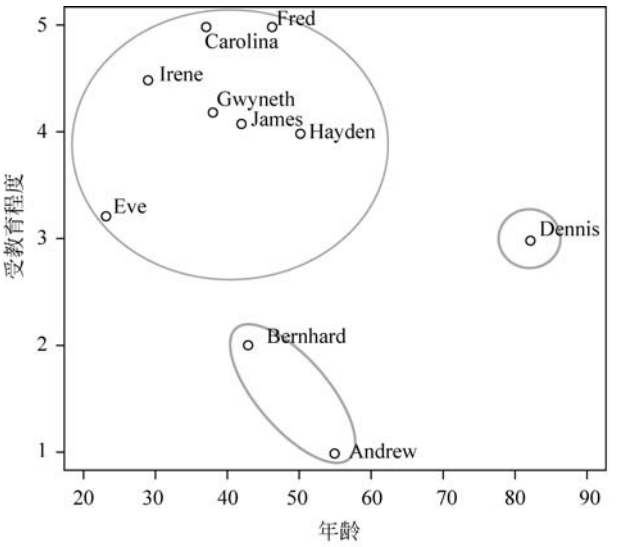
另一个标准是用于聚类定义的方法，确定要包含在同一聚类中的元素。根据这个标准，聚类的主要类型如下。

(1) 基于分离：聚类中的每个对象都比聚类外的任何对象更接近聚类中的每个其他对象。

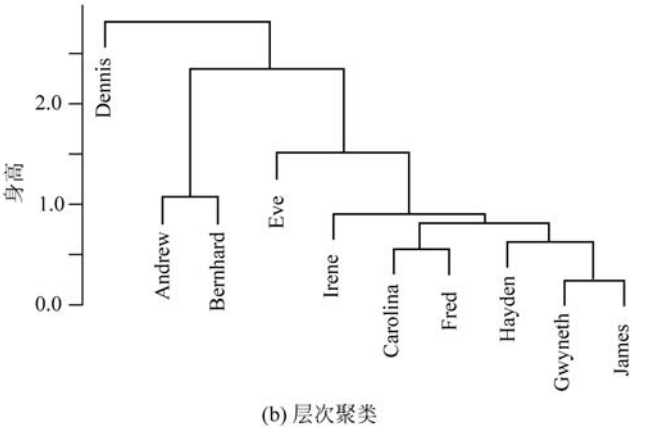
(2) 基于原型：聚类中的每个对象都更接近于表示聚类的原型，而不是表示任何其他聚类的原型。

(3) 基于图表：通过一个图结构呈现数据集，该图结构将每个节点与一个对象关联起来，并将属于同一聚类的对象与一条边连接起来。

(4) 基于密度；该聚类是一个区域，其中的对象有大量的近邻(即一个密集区域)，且被一个低密度的区域包围。



(a) 划分聚类



(b) 层次聚类

图 5.6 划分聚类和层次聚类

(5) 共享属性：该聚类共用某种属性的对象组。

接下来我们将介绍 3 种不同的聚类方法，它们代表了不同的聚类方式。没有哪个比其他的更好，我们将会看到，每种方法都有其优点和缺点。这 3 种方法如下。

(1) K 均值：最常见的聚类算法，代表了传统以及基于原型的聚类方法。

(2) DBSCAN：另一种划分聚类方法，但在本例中是基于密度的。

(3) 凝聚层次聚类：层次聚类和基于图的聚类方法的代表。

5.3.1 K 均值

质心是理解 K 均值的一个重要概念，其代表了一组实例的某种重心。我们首先描述其概念，然后解释 K 均值如何工作、如何读取结果以及如何设置超参数。

1. 中心点和距离测量

质心也可以看作是聚类中所有对象的原型或概况，如聚类中所有对象的平均值。因此，如果我们有几张关于猫和狗的照片，把所有的狗放在一个聚类中，所有的猫放在另一个聚类中，狗聚类中的质心就是一张代表所有狗照片平均特征的照片。因此，可以观察到，聚类的质心一般不是聚类中的一个对象。

例 5.7 Bernhard、Gwyneth 和 James 这 3 位联系人的质心是他们的平均年龄和受教育程度：41 岁(43 岁、38 岁和 42 岁的平均值)和 3.4(2.0、4.2 和 4.1 的平均值)。可以看到，这 3 位联系人都没有这种年龄和受教育程度。

为了将聚类的某个对象作为原型，使用中心点而不是质心。聚类的中心点是与聚类中其他实例间距离最短的实例。

例 5.8 使用相同的例子，3 位联系人的中心点是 Andrew，因为他是与其他两位联系人距离平方和最短的联系人，这个度量被称为组内平方和(参见式(5.7))。由表 4.12 可知，James(J)的组内平方和为 2.33+4=6.33。而另外两位朋友，Bernhard(B)和 Gwyneth(G)的组内平方和更高，分别为 7.79 和 9.46。

我们已经看到，根据所拥有的数据，可以使用几种距离度量。K 均值也是一样，默认情况下，使用欧氏距离，假设所有属性都是定量的。对于其他特性的问题，应选择不同的距离度量。距离测量是 K 均值的主要超参数之一。

2. K 均值如何工作

图 5.7 形象地展示了 K 均值的工作方式，其遵循了 K 均值算法，伪代码如下所示。

K 均值算法

```
1) 输入数据集 D;
2) 输入距离度量 d;
3) 输入聚类数量 K;
4) 定义初始 K 个中心体(它们通常是随机定义的,但可以在某些软件包中明确定义);
5) 重复
6)     根据所选择距离度量 d 将 D 中的每个实例与最近的质心关联;
7)     使用与之相关的所有实例重新计算每个质心;
8) 直到没有实例从 D 中改变相关联的质心.
```

通过 K 均值找到的聚类总是凸形状的，如果连接聚类中任意两个实例的连线位于聚类内，则聚类实例具有凸形状(见图 5.8)，实例包括超立方体、超球体或超椭球体。特定的形状取决于闵可夫斯基距离中的 r 值和输入向量中的属性个数。例如，假定 $r=1$(曼哈顿距离)，如果属性数量为 2，则聚类为正方形；如果属性数量为 3，则为立方体；如果属性数量大于 3，则为超立方体。如果 $r=2$(欧氏距离)，属性数量为 2 的聚类为圆，属性数量为 3 的聚类为球，属性数量大于 3 的聚类则为超球。

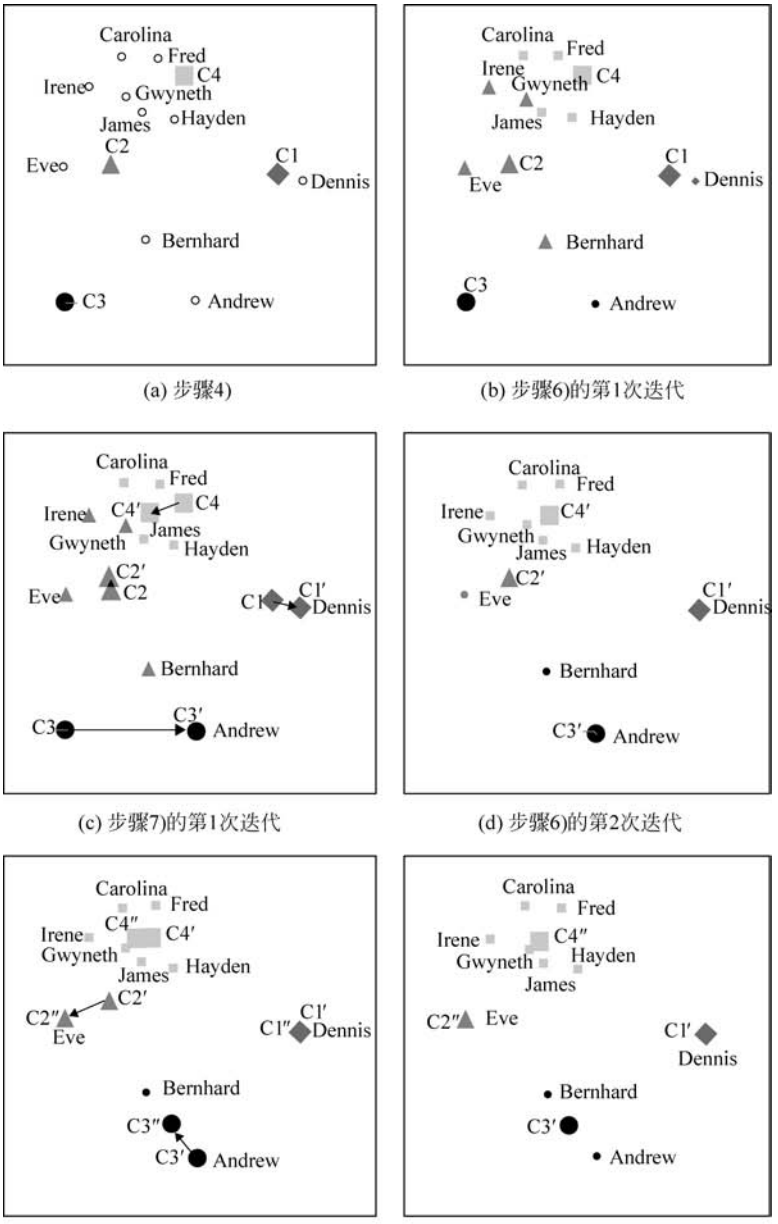


图 5.7　K 均值，$K=4$（大的符号代表中心体，实例按照 K 均值算法逐步进行）

其他距离度量将给出其他凸形状。例如，如果使用马氏距离，则属性数量为 2 的聚类为椭圆，属性数量为 3 的聚类为椭圆体，属性数量大于 3 的聚类则为超椭圆体。

如果一个划分给出的聚类具有某种非凸形状，则应该使用另一种技术查找划分。在使用距离度量时，还应该注意数据要标准化，如 4.3 节所示。

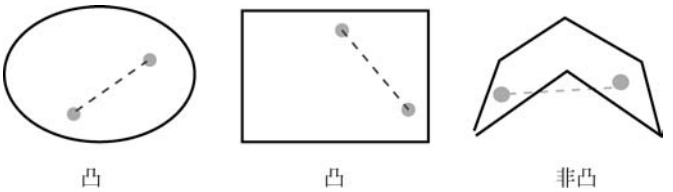


图 5.8 凸和非凸形状示例

随机产生的质心数目是第 2 个超参数，通常用字母 K 表示。现在你知道为什么这个方法以字母 K 开头了。那为什么还有“均值”这两个字？

在有了聚类结果后，如何对它们进行评估？下面我们就要进行介绍。无论使用哪种工具，都可以查看每个聚类的实例（如表 5.3 所示）。通常，聚类按递增顺序编号，从 0 或 1 开始。

表 5.3 根据图 5.7，每位联系人所属的聚类

A	B	C	D	E	F	G	H	I	J
3	3	4	1	2	4	4	4	4	4

如图 5.6 所示，我们也可以绘制聚类，但是，当属性的数量大于 2 时，分析就比较复杂了。有很多方法可以做到这一点，如使用主成分分析（参见 4.5.1 节）。

另一种显示聚类结果的方法则是质心分析，但需要小心。例如，在使用 K 均值等聚类方法之前，应该对定量数据进行规范化。因此，质心一般是规范化的（见表 5.4）。

我们已经从 4.3 节中了解了什么是规范化以及如何对规范化数据进行解释。要分析聚类中心，若在使用 K 均值之前对数据进行了规范化，那么需要理解质心是规范化的。如何读取规范化数据？规范化数据通常在－3 和 3 之间变化。负值越大，其平均值越低；一个值越正，它就越高于平均值。对于每个属性，该值表示偏离质心属性平均值的上下多少。但是，更重要的是，它显示了哪些属性最能区分两个聚类。

表 5.4 示例数据集的规范化质心

质心	年龄	教育
1	－0.55	－0.13
2	1.94	－0.38
3	－1.31	－1.97
4	0.70	1.01

例 5.9 经过分析上述 4 个质心，可以看到聚类 3 的成员平均年龄较小，受教育程度较低；聚类 4 的成员平均受教育程度最高；聚类 2 的成员平均年龄较大；而聚类 1 成员的年龄和受教育程度都略低于平均水平。我们也可以将质心去标准化，以得到原始测量值（参见 4.3 节）。以聚类 2 的质心为例，年龄为 $1.94\times16.19+44.5=75.90$ 岁，受教育程度则为

$-0.38\times1.31+3.6=3.10$。

我们仍然不知道如何设置超参数。距离度量很简单，它取决于数据的特性，这个话题之前已经讨论过了。K 的值是多少？怎么设置呢？

我们从一个问题开始介绍。如果聚类的数量等于实例的数量，那么组内平方和是多少(参见式(5.7))？是0！为什么？因为每个实例都是不同聚类的质心，如图 5.7(a)～图 5.7(f)中 Dennis 和 C1 的位置。所以，一般来说，质心越多，组内平方和就越小。但是，没有人希望拥有大小只有 1 的聚类，因为它在数据描述方面毫无意义，当然也不希望只有一个聚类。如果计算不同 K 值的组内平方和(有些软件包可以做到)并绘制线状图，就会得到图 5.9 所示的曲线。

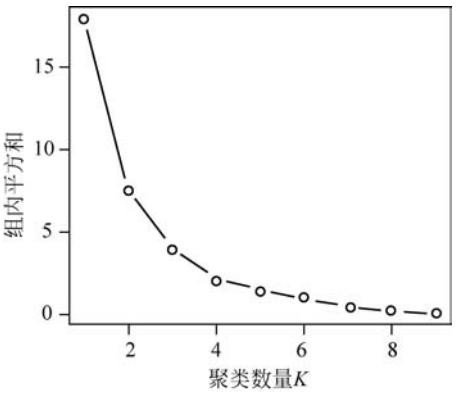


图 5.9　不同 K 值的组内平方和

它被称作肘部曲线，原因很明显(尽管这个弯有点圆)，K 的取值是曲线近似于直线和水平的那个值，也就是肘部的位置。在本例中，$K=4$(或 3)，这是定义 K 值的一种简单而有效的方法。

表 5.5 总结了 K 均值的主要优点和缺点。

表 5.5　K 均值的优缺点

优　　点	缺　　点
• 计算效率 • 经常获得良好的结果：全局最优	• 通常，由于质心文本的随机初始化，每次运行 K 均值的结果都不同 • 需要提前定义聚类的数量 • 不处理有噪声的数据和异常值 • K 均值只能找到聚类具有凸形状的划分

5.3.2　DBSCAN

与 K 均值类似，DBSCAN(带噪声应用的基于密度的空间聚类)也用于划分聚类；与 K 均值不同，DBSCAN 自动确定聚类的数量。DBSCAN 是一种基于密度的技术，它将形成密

集区域的对象定义为属于同一聚类，而不属于密集区域的对象则被认为是噪声。

高密度区域的识别是通过首先识别核心实例来实现的。核心实例 p 直接达到的其他实例最少，这个最小值是一个超参数。要被认为是“直接可达的”，实例 q 到 p 的距离必须小于预定义的阈值（用 ε 表示）。ε 是另一个超参数，常称为半径。如果 p 是一个核心实例，那么它与所有可以从它直接或间接访问的实例一起组成一个聚类。每个聚类至少包含一个核心实例，但非核心实例可以是聚类边缘的一部分，因为不能用它们访问更多实例。DBSCAN 还具有一些随机性，因为当有多个核心实例可以直接访问某个给定实例时，需要决定将该实例附加到哪个核心实例。尽管如此，随机化通常不会对聚类的定义产生有意义的影响。

图 5.10 给出了一个使用 DBSCAN 进行分析的实例，其中使用图中所示的半径作为超参数，且实例数量最少为两个。我们可以看到一个聚类（Carolina，Fred，Gwyneth，Hayden，Irene 和 James）和 4 个离群值（Andrew，Bernhard，Dennis 和 Eve）的存在。

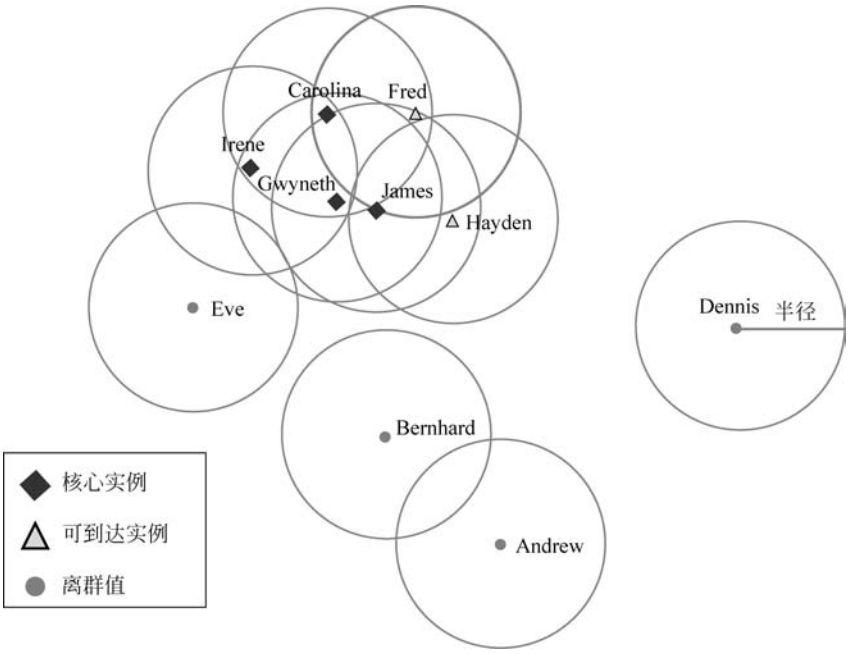
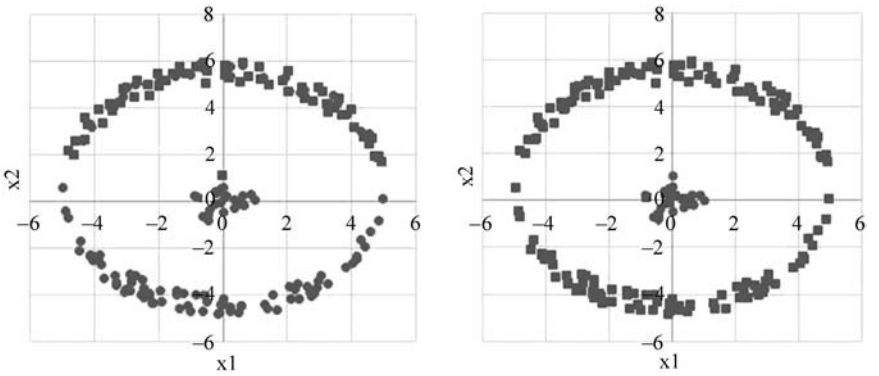


图 5.10 使用最少数量为 2 的示例数据集的 DBSCAN

为了更好地表示 K 均值和 DBSCAN 之间的差异，我们可以在图 5.11 中看到一个不同的数据集，它更好地显示了差异。对于 K 均值，我们将聚类的数量定义为 2，而对于 DBSCAN，使用 3 作为实例的最小数量，并将 ε 设置为 2。

DBSCAN 的评估结果没有任何特殊的特点，在 DBSCAN 中也没有质心，可以评估每个聚类的实例。



(a) $K=2$时的K均值　　(b) 实例数最少为3的DBSCAN($\varepsilon=2$)

图 5.11　K 均值与 DBSCAN 的比较

DBSCAN 的主要超参数是将给定实例划分为核心所需的最小实例数，某个实例的最大距离应该是直接可达的。尽管存在一些超参数设置的研究，但通常的做法是测试不同的值，并基于观察到的聚类和异常值的数量，分析得到的结果是否可以接受。距离度量也是一个超参数，应该根据属性的尺度类型进行选择。

表 5.6 总结了 DBSCAN 的主要优点和缺点。

表 5.6　DBSCAN 的优缺点

优　　点	缺　　点
• 可以检测到任意形状的聚类 • 对离群值的鲁棒性	• 由于一些随机性，每次运行 DBSCAN 的结果可能会有些不同，但结果通常不会有太大的差异 • 计算上比 K 均值更复杂 • 设置超参数值比较困难

5.3.3　聚合层次聚类技术

层次算法逐步构建聚类，需要从单个聚类中的所有实例开始并逐步划分，也可以通过从与实例数量相同的聚类开始并逐步将它们连接起来实现。第 1 种方法是自顶向下，第 2 种方法是自底向上。

聚合层次聚类是一种自底向上的聚类方法，我们将用自己的例子展示这个方法。首先需要计算所有实例对之间的距离。为此，我们使用规范化数据和欧氏距离。由于实例和它本身之间的距离总是 0，因此对角线也总是 0。因为这个矩阵是对称的，所以只给出了一半。例如，Andrew 和 Bernhard 之间的距离等于 Bernhard 和 Andrew 之间的距离，所以说矩阵的另一半是没有必要的(见表 5.7)。

表 5.7 聚合层次聚类的一次迭代

	A	B	C	D	E	F	G	H	I	J
A	0									
B	1.07	0								
C	3.26	2.33	0							
D	2.26	2.53	3.17	0						
E	2.60	1.54	1.63	3.65	0					
F	3.11	2.31	0.56	2.70	1.98	0				
G	2.67	1.71	0.62	2.87	1.20	0.79	0			
H	2.32	1.59	1.11	2.12	1.78	0.80	0.76	0		
I	3.13	2.10	0.63	3.47	1.06	1.12	0.60	1.35	0	
J	2.51	1.61	0.76	2.61	1.36	0.73	0.26	0.50	0.86	0

接下来找到距离最短的一对，在本例中是 Gwyneth-James，下一步是创建另一个少一行和一列的矩阵。在这个地方，Gwyneth 和 James 将被视为一个单独的对象 GJ（见表 5.8）。目前仍然存在一个问题，如何计算任何实例到 GJ 的距离？在这个例子中，我们使用平均连接标准。例如，Eve 和 GJ 之间的距离是 Eve 和 Gwyneth(1.20)以及 Eve 和 James(1.36)之间距离的平均值，也就是 1.28。

表 5.8 聚合层次聚类的二次迭代

	A	B	C	D	E	F	GJ	H	I
A	0								
B	1.07	0							
C	3.26	2.33	0						
D	2.26	2.53	3.17	0					
E	2.60	1.54	1.63	3.65	0				
F	3.11	2.31	0.56	2.70	1.98	0			
GJ	2.59	1.66	0.69	2.74	1.28	0.76	0		
H	2.32	1.59	1.11	2.12	1.78	0.80	0.63	0	
I	3.13	2.10	0.63	3.47	1.06	1.12	0.73	1.35	0

在每个步骤中，由于两个单元格的聚合，将从矩阵中删除一行和一列。在矩阵的大小为 2 时结束（两行两列）。而对于一个有 n 个实例的问题，会有 $n-1$ 步。

在上面的例子中，Eve 和 GJ 之间的距离是 Eve 和 Gwyneth 以及 Eve 和 James 之间距离之和的平均值。下面我们还会看到其他方法。

1. 连接标准

假设有两组实例，如何计算这两组实例之间的距离？下面提出了 4 个最常用的连接标准。

(1) 单一连接：度量最接近的实例之间的距离，每个集合一个，适用于优势聚类，如图 5.12(a)所示。

(2) 完整连接：度量距离最远实例间的距离，每个集合一个实例，适用于类似聚类，如图 5.12(b)所示。

(3) 平均连接：度量每对实例间的平均距离，每个集合一对实例。它介于前两种方法之间，如图 5.12(c)所示。

(4) Ward 连接：用于测量合并后聚类内方差的增加，支持紧凑的聚类。

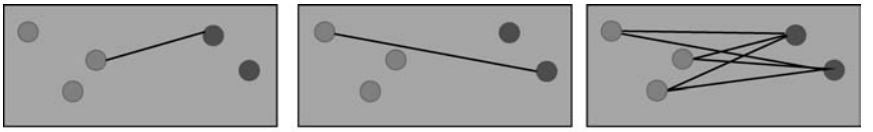


图 5.12　单一、完整和平均连接标准

我们来看这 4 个连接标准如何应用于数据集，如图 5.13 所示。

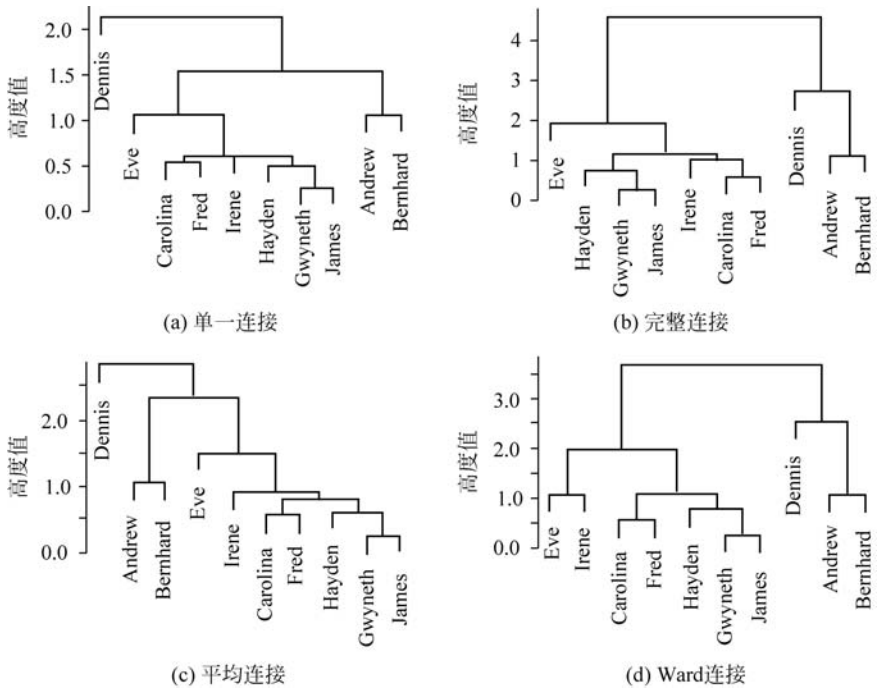


图 5.13　不同连接标准对样本数据集的影响

值得注意的是，单一连接和平均连接标准并没有太大的不同，主要的区别在于 Irene。完整连接和 Ward 连接也相对类似，主要不同之处也和 Irene 有关。不过，与完整连接和 Ward 连接相比，单一连接和平均连接表现出更有意义的差异。事实上，后两个连接标准有两个定义良好的聚类，而前两个聚类则有一个异常值(Dennis)。

但是，为了更好地理解图 5.13，我们来看看如何阅读树状图，这是层次聚类中的一个重要课题。

2. 树状图

利用与每个矩阵间的最小距离(第一矩阵和第二矩阵的值分别为 0.26 和 0.56),可以绘制所谓的树状图,如图 5.14 所示。连接 Gwyneth 和 James 的水平分支的高度值为 0.26,即平均连接的高度值,而连接 Carolina 和 Fred 的水平分支的高度值则为 0.56。

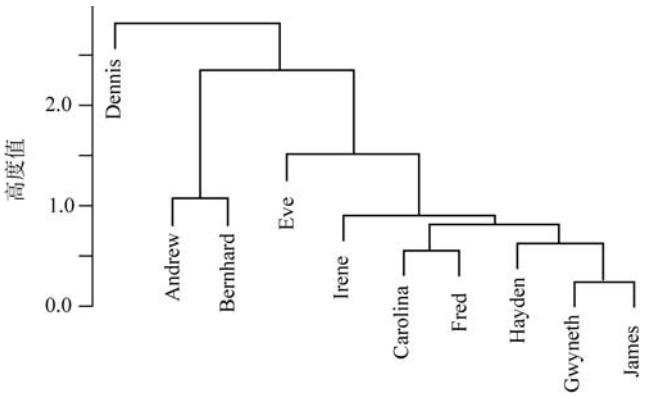


图 5.14 聚合层次聚类定义的树状图,平均连接标准

从树状图中很容易得到所需的聚类数。如果我们想要 4 个聚类,应该看看最高水平分支 4−1=3。这些分支定义了 4 个聚类:Dennis 在第 1 个聚类中,Andrew 和 Bernhard 在第 2 个聚类中,Eve 在第 3 个聚类中,其他的则都在第 4 个聚类中。如你所见,这个图很容易理解。但是应该始终注意到,聚类定义的相似性很大程度上取决于距离度量如何"捕捉"不同概念的方式。

树状图的优点之一在于可以很容易地识别离群值。在这种情况下,最高的分支(最大的距离)只有一个元素在一边,而所有其他的元素在另一边。Dennis 就是这种情况,主要是因为他比其他人年龄大得多。

树状图是分析层次聚类结果的主要工具,也可以从树状图生成一定数量的聚类。利用这种可能性,可以评估每个聚类的实例,对于不太大的数据集,树状图已经是一种评估的有效方式了。

聚合层次聚类的主要超参数是距离测度和连接标准。上述两种情况都已在前面介绍过。

表 5.9 总结了聚合层次聚类的主要优点和缺点。

表 5.9 聚合层次聚类的优缺点

优 点	缺 点
• 易于解释,但对于大型数据集更混乱 • 设置超参数很容易	• 在计算上比 K 均值更复杂 • 对于几个领域,对树状图的解释相当主观 • 经常给出局部最优:4.5 节有一个典型的问题查找方法

5.4 本章小结

本章介绍了数据分析中常用的一种技术，即聚类分析。聚类技术的研究涉及范围很广，这里介绍该领域当前的一些趋势。

当前研究的一个领域是通过对象和属性同时分组创建聚类。因此，如果数据集以表的形式（见表 5.1）出现，那么聚类技术的应用将对行（实例/对象）进行分组，而对该数据集应用双聚类技术将同时对行和列（属性）进行分组。生成的每个双聚类是对列的子集具有类似值的行的子集，或对行的子集具有类似值的列的子集。

在一些实际应用中，数据是连续生成的，进而形成数据流。将聚类技术应用于不同时刻的数据流可以产生不同的划分。聚类的大小和形状可能会发生变化，出现现有聚类的合并或划分，以及包含新的聚类。在数据流挖掘中使用的聚类技术，必须能在处理和内存使用方面有效地应对不断增长的数据量。

BIRCH（使用层次结构的平衡迭代减少和聚类）这种特殊的技术，已经被设计出来以满足这些需求，并已经成功地用于数据流挖掘。BIRCH 有两个阶段，在第 1 阶段，它扫描数据集并逐步构建一个分层数据结构，称为聚类特征树（CF-Tree），其中每个节点存储从相关聚类中提取的统计数据，而不是聚类中的所有数据，这样就节省了计算机内存；在第 2 阶段，对 CF-Tree 的叶节点中的子聚类应用另一种聚类算法。

K 均值等一些聚类技术，可以在每次运行中生成不同的划分，这是因为它们随机生成初始质心。利用不同的聚类技术获得的划分通常是不同的，即使这些划分对验证索引有不同的值，它们也可以提取数据的相关方面。理想的情况是在划分中找到一个模式或“共识”。集成技术的使用，使不同的划分可以组合成一个一致的划分。例如，尝试将出现在同一聚类中大多数划分中的实例保持在一起。

大量的属性是真实数据集中常见的问题，有些属性是相关的，有些是不相关的，还有些是冗余的。它们的存在会增加计算成本并降低所找到的解决方案的质量。特征选择技术可以确定最相关的特征，从而降低成本并提供更好的解决方案。虽然看起来很简单，但是特征选择是数据分析的一个关键挑战。在监督任务中，类标签可以指导特征选择过程。而在聚类分析等非监督任务中，由于缺乏类标签的信息，就加大了特征选择过程中相关特征的识别难度。所选择的特性对所找到的划分有很大的影响。

如果我们使用外部信息，如哪些对象应该（或不应该）在同一个聚类中，那么通过聚类技术可以提高划分的质量。例如，如果知道数据集中某些对象的类标签，那么相同聚类中已标记的对象应该来自相同的类。利用外部信息帮助聚类被称为半监督聚类。一些真正的任务可以受益于半监督学习的使用，异常检测就是其中之一，它查找数据集划分中与同一聚类大多数其他对象不同的对象。

5.5 练习

(1) 使用社交网络数据集,对不同的 K 值($K=2$、$K=3$ 以及 $K=5$)运行 K 均值算法。

(2) 使用社交网络数据集,运行 DBSCAN 算法,测试两个主要超参数的不同值。

(3) 使用社交网络数据集,运行聚合层次算法,绘制生成的树状图。

(4) 如果改变向 K 均值算法输入数据的顺序,会发生什么? 使用社交网络数据集找出答案。

(5) 计算 composition 与 conception 之间的编辑距离。

(6) 分析图 5.15 中的树状图,其中为美国的一个政党在几次选举过程中得到的数据。

① 该党派在哪个州的选举结果更像下面两个州?

A. 密歇根州

B. 威斯康星州

② 只考虑从阿拉斯加到德克萨斯的几个州,对于下面两种情况,每个聚类的成员是什么?

A. 3 个聚类

B. 8 个聚类

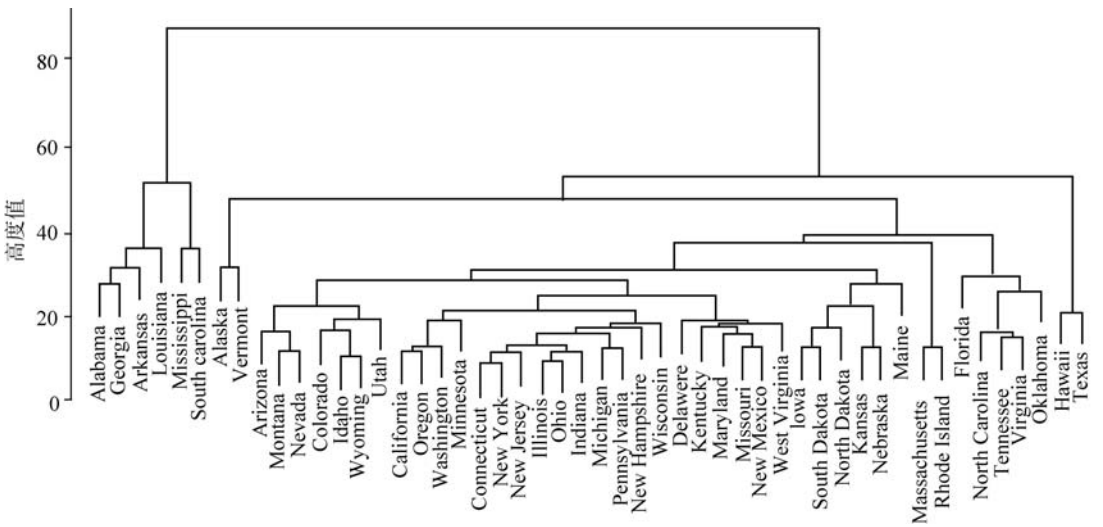


图 5.15 树状图

(7) 通过定量向量表示图 5.16 中“太空入侵者”的图像,并使用欧氏距离、曼哈顿距离和汉明(或编辑)距离度量计算它们之间的距离(即它们的相似度)。

(8) 一些聚类算法生成的划分具有任意形状,而另一些算法则限制了所形成的聚类形状。描述每组算法与其他组相比的一个优点和一个缺点。

（9）写 3 篇至少包含 3 句话的短文。以单词包的形式表示这些文本，并根据这种格式计算它们的距离。

图 5.16　太空入侵者