第5章 聚类分析[①]

本章学习目标

- 掌握聚类分析中距离测量以及结果评估方法
- 掌握 K-means、密度聚类、层次聚类等常用聚类方法的基本原理
- 掌握利用 Python 进行聚类分析
- 了解聚类分析方法在计算社会科学的应用场景

5.1 计算社会科学也需要聚类分析

群体是社会科学研究的基本分析单位之一。Sociology 有“群学”的译法，量化研究对异质性群体进行分解，在理论和类型学中都极为重要。本章介绍的聚类分析（Cluster Analysis）就是一类从量化数据中自动提取异质性群体的方法，它通过测量或感知到的内在特征或相似性对对象进行分组或聚类，进而探索和发现数据中的潜在结构。聚类分析是量化社会科学的基础统计方法，然而它在计算社会科学中也有许多应用场景，是实现非结构化数据快速分组和降维的基础策略。

让我们试想一个场景：如果你从互联网公司获得了一个用户行为数据库，该数据库捕捉到了许多有趣的社会现象，却具有诸多复杂而特殊的变量（或说特征），这些信息远远超出了传统调查的预想，请问你该怎么办呢？传统的量化社会科学分析通常遵循“理论－假设－测量”的研究思路，基于低维特征对用户进行分类，而这种方法在面对特征维度极高的用户行为大数据时往往“捉襟见肘”，且生成成本极高。机器学习的兴起则大大拓展了原有的研究工具，帮助我们得以在数据驱动下充分挖掘数据信息。具体来说，通过将相似的观察结果

① 感谢范晓光对本章文字和内容的校对.

或特征分组到子群体中，无监督的降维和特征工程可以为大型数据集的内在结构提供基于数据本身的科学化建议。①

毋庸置疑，数字时代的到来为我们提供了大量面对以互联网用户行为数据（包括文本、图像和视频等类型）为代表的高容量、高维数据集的机遇，也为发现数据中新的群体提出了更高的要求，计算社会科学家需要新的工具来挖掘这些来源、类型和结构都非常复杂的大数据。我们希望在对新数据没有过多知识积累的前提下，可以找到计算方法能帮助探索数据集中承载了哪些重要的社会群体，数据具有怎样的自然分组等。在我们看来，引入聚类分析就是一种恰当的选择。

5.2 聚类分析基础

5.2.1 距离：如何测量两个人的相似程度？

中国素来有句古语，“物以类聚，人以群分”，它很好地道出了聚类分析的两个原则：“类聚”与“群分”。一般来说，我们认为相同群体的成员“臭味相投”，而不同群体的成员间则“道不同，不相为谋”。志趣相投的人经常处在一起，而风格迥异的人则通常不相往来，这正是聚类分析的基本假定。然而，这样的思想如何在数据中体现呢？

在一个常见的数据集中，每个成员大多具有相同维度的描述性特征，这意味着我们对于这些成员具有一个共同的描述体系（虽然这个评估体系是有侧重的）。如果让我们运用简单的数学知识，为每个独立特征设定一个相互正交的坐标轴来反映其数值度量，这实际就将所有成员的描述性特征设定为了某种空间坐标（当维度很高时是难以想象的）。借助这种转换，我们会发现所有在描述性特征上完全相同的成员，在空间上就具有相同的位置；相反，如果两个成员在空间上远离，那么他们的特征就可能具有较大的差异。我们不难发现，此时空间上的距离关系可以反映两个成员在特征上的相似程度。由此，借助于空间中成员间的聚集和分散，我们就可以较好地观察群体内的一致性与群体间的异质性，也就是对数据本身的自然分组进行展示。

为了更好地评估这种相似性与异质性，我们引入距离度量（Distance Measure）来统一数据集内相似度的测算。在规定了距离度量后，数据集内两两成员间的相似程度（也就是距离）可以确定。聚类分析正是通过引入距离度量，将数据集的成员放入了相同的向量空间。由于空间上聚集与远离会反映成员间的相似或相异，我们可以较好地观察成员间的自然分组，该分组潜在地具有社会群体层面的解释意义。距离度量是人为规定的，我们可以根据数据的特性来选择。在机器学习中，我们通常使用 L_p 距离或称“闵可夫斯基距离”（Minkowski Distance）作为距离度量。对具有 m 个样本以及 n 个特征的数据集 $\chi_{m\times n}$ 而言，$x_i, x_j \in \chi_{m\times n}$，$x_i = (x_i^{(1)}, x_i^{(2)}, \cdots, x_i^{(n)})^{\mathrm{T}}$，$x_j = (x_j^{(1)}, x_j^{(2)}, \cdots, x_j^{(n)})^{\mathrm{T}}$，我们定义 L_p 距离为：

$$L_p(x_i, x_j) = \left(\sum_{l=1}^{n} | x_i^{(l)} - x_j^{(l)} |^p \right)^{\frac{1}{p}} \tag{5.1}$$

① Grimmer, Justin, Roberts, et al. Machine Learning for Social Science: An Agnostic Approach. Annual Review of Political Science, 2021, 24: 395-419.

上式中 $p \geqslant 1$，欧氏距离(Euclidean distance)其实是 $p=2$ 时，L_p 距离的特例：

$$L_2(x_i, x_j) = \sqrt{\sum_{l=1}^{n} | x_i^{(l)} - x_j^{(l)} |^2} \tag{5.2}$$

当 $p=1$ 时，则有曼哈顿距离(Manhattan distance)也称街区距离，为 x_i, x_j 各坐标距离之和：

$$L_1(x_i, x_j) = \sum_{l=1}^{n} | x_i^{(l)} - x_j^{(l)} | \tag{5.3}$$

最后，当 $p=\infty$ 时，则有切比雪夫距离(Chebyshev distance)，也称棋盘距离，它的计算实质上是 x_i, x_j 各个坐标距离的最大值：

$$L_\infty(x_i, x_j) = \max_l | x_i^{(l)} - x_j^{(l)} | \tag{5.4}$$

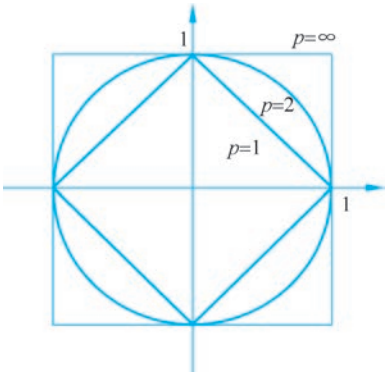

图 5.1 不同 L_p 距离的等距

图 5.1 展示了二维空间中，p 取不同值时，与原点 L_p 距离为 1 的点的集合，它更直观地呈现了这些距离度量的区别。

我们不难发现，在不同的距离度量下，同一数据集中两个成员的距离也会随之改变。实际上，距离度量会影响聚类分析所发现群体的数据特征。由于通常使用的欧氏距离自身的特性(圆形的等距线)，大部分使用欧氏距离的算法都偏好发现在空间中呈现超球面形(Hyperspherical-shaped)的群体，此时采用更具弹性的距离度量则有助于适应更广泛的数据特征。①

通过距离度量可以把数据集中不同成员间的相似度转换到一个统一的空间。一般而言，相同群体的成员会因为特质的相似而在空间上相互接近，不同群体的成员则在空间上相互远离，由此，我们就有可能识别数据集中的异质性群体。然而，如何自动地获取这些群体？如何衡量这些机器生成的群体的质量？前一问题涉及具体的聚类算法，后文将集中介绍；而后一问题则涉及我们对机器生成的异质性群体的预期。总的来说，聚类结果的性能评估是对群体内一致性以及群体间异质性的综合衡量。

5.2.2 聚类结果的性能评估

再次回到对群体的基本假设上，在定义了距离(也就是基于成员间的相似性)后，我们就已经构建了一个用来透视群体的空间。现在我们假定已经获得了聚类结果，也就是为数据集中的成员唯一划定了所属群体。我们该如何衡量这种划分的优劣呢？先来回顾一下我们对群体的一些基本期望。在生活中，我们首先认为群体成员间“臭味相投”，这指出了社会群体的内部同质性；另外，我们也会认为社会群体具有区域性，“南橘北枳”这种表述暗示了社会群体由区位而产生的群体层次异质性。以上两种常识则同时构成了我们评估聚类结果的基本思想。在理想状况下，我们希望获得的群体在内部尽可能一致，而与外部则尽可能不同，这就产生了“类型内相似度”(Intra-cluster Similarity)与“类型间相似度”(Inter-cluster

① Mao J C, Jain A. A Self-Organizing Network for Hyperellipsoidal Clustering (HEC). Proceedings of 1994 IEEE International Conference on Neural Networks (ICNN'94), 1994.

Similarity)两个概念。在构建数学指标时,我们通常希望聚类结果前者更高,而后者更低。

对聚类结果的评估通常基于数据集本身已有类标签。此时,我们可以将聚类结果与既有标签进行比较,直接评估聚类结果的精确性与有效性。但这样的情况在社会科学领域并不多见,本章不做过多讨论,有兴趣的读者可自行阅读。事实上,当拿到数字痕迹或者传感器数据时,除了数据提供方的简单介绍(有时甚至没有),我们往往对数据集本身的类型知之甚少。在这种情况下,根据聚类结果本身的数据特征来进行评估就尤为关键,我们接下来简要介绍其中的数学过程。

在评估聚类结果时,最广泛使用的指标是"轮廓系数"(Silhouette Coefficient)。该指标通过描述群体的轮廓,可以较好地测量群体间的重叠与分离趋势;并且在需要研究者指定群体数量 k 的模型中,通过轮廓系数能够可视化地进行参数选择。[①] 由于轮廓系数很好地体现了聚类结果性能评估的思路,我们将在本节着重介绍。首先要对一些重要的数学指标进行说明和界定。假定我们手头的数据集仍然是 $\chi_{m\times n}$,它具有 m 个样本以及 n 个特征。

与之前不同,我们目前已经定义了距离度量,用来确定成员间的相似度(也就是距离)。对于 $x_i, x_j \in \chi_{m\times n}$,我们用 $\mathrm{dist}(x_i, x_j)$ 来表示两个成员间的距离;假定已经得到了聚类结果 C,在 C 中我们区分了 $\chi_{m\times n}$ 中的 k 个群体,于是有 $C=\{C_1, C_2, \cdots, C_k\}$。对于个体 x_i 而言,他/她在聚类结果中被归属于群体 C_a,则有 x_i 的"平均类内距离"(Mean Intra-cluster Distance),该指标反映了群体中特定成员与其他群内成员的平均相似程度,其计算公式为

$$a(x_i)=\frac{1}{|C_a|}\sum_{\substack{j=1\\ j\neq i}}^{|C_a|}\mathrm{dist}(x_i, x_j) \tag{5.5}$$

在式(5.5)中,$x_j \in C_a$,$|C_a|$ 代表集合 C_a 中的元素个数。我们接下来计算 x_i 与 C_a 之外其他某个群体 C_n 的"平均类间距离"(Mean Inter-cluster Distance),该指标反映了群体中特定成员与其他群体成员的平均距离。当 $x_j \in C_n$ 时,我们将这一指标的计算公式表达为:

$$d(x_i, C_n)=\frac{1}{|C_n|}\sum_{\substack{j=1\\ j\neq i}}^{|C_n|}\mathrm{dist}(x_i, x_j) \tag{5.6}$$

对 x_i 这个成员来说,我们只关心与他所属的群体 C_a 平均距离最近的那个群体 C_b,因为这反映了两个最贴近的群体间边界,即轮廓。我们称 x_i 与 C_b 的平均类间距离为"平均最近类间距离"(Mean Nearest-cluster Distance):

$$b(x_i)=\min_{1\leqslant n\leqslant k} d(x_i, C_n) \tag{5.7}$$

在式(5.7)中,C_n 是除了 x_i 所属群体 C_a 之外其他的某个群体。在获得了 $a(x_i)$ 与 $b(x_i)$ 后,我们就能确定 x_i 的轮廓系数,其大小反映了 x_i 远离群体轮廓的程度:

$$s(x_i)=\frac{b(i)-a(i)}{\max\{a(i), b(i)\}} \tag{5.8}$$

当求取数据集 $\chi_{m\times n}$ 中所有成员的轮廓系数并计算均值后,我们就获得了聚类结果 C 的轮廓系数,它反映了聚类结果所划分的群体间分离重叠的整体状况。

我们从轮廓系数的指标构成可知,它的取值范围在[-1,1]。轮廓系数通常都是大于 0

① Rousseeuw P. Silhouettes: A Graphical Aid to the Interpretation and Validation of Cluster Analysis. Journal of Computational and Applied Mathematics, 1987, 20: 53-65.

的，系数越大说明聚类结果中不同群体间的轮廓越明确。在模型选择上，我们可以比较不同模型的轮廓系数，并结合理论需求，选择系数更为优越的模型。当该值为 0 时，则代表不同群体完全重叠；当该值小于 0 时，代表聚类结果更多将成员划入了不恰当的群体，此时需考虑更改模型的参数或选用其他聚类算法。

由于轮廓系数反映了每个成员与聚类结果的关系，我们可以进行可视化，以辅助我们选择更恰当的模型。图 5.2 中的例子展示了如何借助每个样本的轮廓系数对聚类结果进行整体的评估，进而帮助我们进行模型选择。[①] 在这个案例中，研究者使用了具有 75 个样本的 Ruspini 数据集，[②]它是常用于聚类分析的测试数据集，在理想状态下应被分成 4 类。在借助轮廓系数进行参数选择时，我们可以先按照 k 从小到大依次生成一定数量的模型，并对聚类结果中各个群体内成员的轮廓系数按照从大到小的顺序全部绘制，这样就能直观地看到聚类结果中每个群体的大致状况。我们发现，较好的模型平均轮廓系数较高且划分的群体在体量与轮廓上都更为匀称。

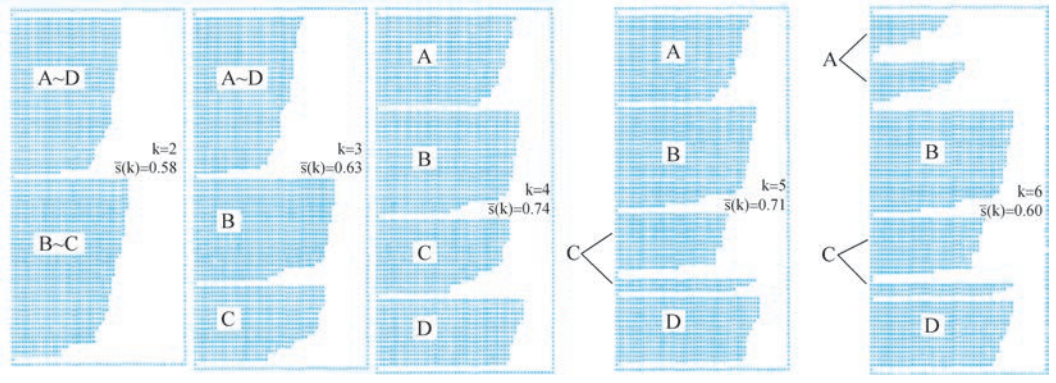

图 5.2 借助轮廓系数对不同 k 值下 K-median 聚类结果进行可视化

以上内容主要介绍了轮廓系数的数学过程，以及如何使用轮廓系数来评估聚类结果。作为一种指标体系，轮廓系数在使用上较为广泛，且本身对聚类结果的描述也相对直观，另外，当我们希望对聚类的效果进行可视化时具有比较直观的表达，这些特性都满足了社会科学研究对可理解性的偏好。当然，读者完全可以选用其他指标来佐证自己对聚类模型的选择。但在理解这些指标时，我们要认识到它们都会综合考虑类内距离与类间距离的问题，也就是“类聚”与“群分”。只要牢牢把握这两点，就能更好地理解数学过程背后非常直白的意图。我们下面再介绍两个常用的聚类评估指标，后续的案例也会加以展示。

CH 指标(Caliliski-Harabasz Index)对类间与类内的离差矩阵进行了统计调整，以综合考虑聚类结果的分离与聚合状况。[③] CH 值总是大于 0 且没有极值，聚类结果中较好的群间分离与较好的群内凝聚都会带来指标的提升。该指标适用于研究者对聚类结果类间、类内状况的关注比较平衡的场景。

① Rousseeuw P. Silhouettes: A Graphical Aid to the Interpretation and Validation of Cluster Analysis. Journal of Computational and Applied Mathematics, 1987, 20: 53-65.

② Ruspini E. Numerical Methods for Fuzzy Clustering. Information Sciences, 1970, 2(3): 319-350.

③ Caliński T, Harabasz J. A Dendrite Method for Cluster Analysis. Communications in Statistics, 1974, 3(1): 1-27.

DB 指标(Davies-Bouldin Index)与前两个指标的关注点略有不同,它仅考虑聚类结果中效果最差的区域,并将最差区域的重叠程度(在聚类结果较好的时候是分离程度)除以 k 值来平衡模型复杂程度的影响。[①] 读者一定已经意识到,由于这一指标评估的是聚类结果最差区域的情况,当该值越大,说明模型在局部有越大的混淆和重叠,这意味着模型质量更差,由此我们期待 DB 指标越小越好。

5.2.3 聚类结果的选择与解读

在本节中,我们将对聚类结果评估做一个简要的归纳。如果我们已经选择了恰当的聚类算法,并得到了一系列参数下的聚类结果。在这种情况下:

(1) 可以先通过单个或者多个评估指标的使用来挑选数据在自然特征上更为优越的模型。通过把不同模型的客观指标在折线图中连续绘制,我们可以更好地识别不同聚类结果的变化趋势。

(2) 为了使结果直观化,我们可以把聚类结果通过散点图进行展示,来直观地透视聚类结果的分布与交叠。对于高维数据可以通过 t-SNE、UWAP 算法进行降维后再可视化,我们会在后续的案例中做介绍。

(3) 我们通常不会武断地挑选唯一的模型,这在特征复杂以及体量过大的数据上更是如此。我们应该结合客观指标、可视化结果以及具体划分出的群体的特征进行综合性评估。

(4) 但正如我们已经提及的,评估结果更多只是作为参照,以服务于我们的研究目的。由于聚类分析具有很强的探索性特质,最终的模型选择通常要基于解释性“一锤定音”。如何在可解释性这个“弹性指标”中,将经验、理论与数据完美结合,是每一位研究者都必须认真思索的议题。

然而,不同聚类中心数量的模型很多时候具有相似的性能。在这种情况下,决定性的判准来源于研究者的洞见与视野。虽然我们可以通过观察划分群体的特征、对聚类整体进行可视化等方式进行合适模型的判断,但最关键所在是所划分群体是否具有社会科学意义上的“可解释性”,[②]以及能否助力我们进行新的知识发现。[③]

5.3 原型聚类

5.3.1 经典聚类算法: K-means

上一节侧重从理论上介绍聚类分析的距离度量以及性能评估。在此基础上,我们将在最后展示具体的算法。考虑到不同聚类方法的思路差异较大,难以归纳出通用的聚类流程,我们在介绍聚类方法的共同背景后再来依次介绍聚类算法。常见的聚类算法主要包括原型聚类(Prototype-based Clustering)、密度聚类(Density-based Clustering)和层次聚类(Hierarchical Clustering)。下面,我们将优先介绍其中发展最早也是最成熟的原型聚类,该方法最大的特点是假设数据自身的特征能通过一组原型(也就是典型案例)来刻画[④],这与

① Davies D, Bouldin Donald. A Cluster Separation Measure. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1979.

② 黄荣贵. 网络场域、文化认同与劳工关注社群: 基于话题模型与社群侦测的大数据分析. 社会, 2017(2): 26-50.

③ Jain A. Data Clustering: 50 Years Beyond K-means. Pattern Recognition Letters, 2010, 31(8): 651-666.

④ 周志华. 机器学习. 北京: 清华大学出版社, 2016.

韦伯对“理想类型”有异曲同工之处。在某种程度上，原型聚类的思路与社会科学思维有很强的亲和性，这也大大增强了其结果在社科领域的可解释性。

在原型聚类中，K-means 是最具生命力也最为经典的聚类算法。在计算社会科学中，它仍然被各领域广泛使用，并作为大量聚类算法的基础或工序存在。[①②] 理解 K-means 算法，有助于我们厘清聚类算法的基本逻辑，也为我们理解更新的算法奠定基础。我们也将在本节加入数学过程，虽然难免有些枯燥，但读者应当理解数字背后的计算思想。

在 K-means 算法中，我们用聚类中心(Cluster Center)来表示原型，其本质是群体特征的均值。聚类中心的数量 k 依赖于研究者的主观设定，该值意味着对数据集中潜在群体数量的假定，它通常依赖于研究者直接观察或基于指标的探索来确定。K-means 算法将基于设定的 k 值来确定恰当的聚类中心，对于具体的成员则把距离最近(也就是相似度最高)的聚类中心视作其原型，以此达到聚类的效果。

那么如何来确定一个恰当的聚类中心呢？从逻辑上，我们实际上希望划分出来的群体在内部相似度最高，这意味着要使所有群体与对应聚类中心的平均距离最小，于是当我们使用欧氏距离作为距离度量时，我们实际上希望从数据集 $\chi_{m\times n}$ 划分所得群体$\{C_1, C_2, \cdots, C_k\}$的“误差平方和”(Sum of Squared Errors，SSE)[③]最小，我们将某个群体 C_i 的聚类中心(也就是类内均值)视作 μ_i，则有：

$$\mathrm{SSE}=\sum_{i=1}^{k}\sum_{x\in C_i}\| x-\mu_i \|_2^2 \tag{5.9}$$

其中，$\| x-\mu_i \|_2$ 指的是 C_i 中某个点与其聚类中心 μ_i 的 L_2 距离，也就是欧氏距离。在理想状态下，我们一定希望直接获得使 SSE 取得全局最小值的群体划分，但这意味着要遍历数据集中所有聚类中心为 k 的可能划分，而当数据量达到一定规模，这几乎是不可能完成的任务。[④] 因此，K-means 算法其实提供了一种使聚类结果的 SSE 尽可能小的思路。接下来，我们具体看看 K-means 是如何实现该目标的。

K-means 算法的训练总体分为三个阶段：初始化、参数更新以及收敛，整个流程为(a)输入数据；(b)初始化聚类中心并确定群体划分；(c)(d)在迭代中更新聚类中心与群体划分；(e)聚类结果收敛。(见图 5.3)。[⑤] 在训练开始前，研究者需要先指定潜在的聚类中心数量 k，这将会决定最终获得的群体数量。研究者们同样可以指定的是训练终止条件，如指定最大迭代的次数、规定参数更新的最小幅度等。在完成以上设置后，K-means 就可以开始自动运行。K-means 首先会从数据集中选择 k 个成员作为最初的“原型”k，即聚类中心。虽然这种选择可以是完全随机的，但是聚类结果对此极为敏感[⑥]，故很多时候需要对 K-means 进行多次运行，以获得更好的聚类结果。

① Jain A, Murty M N, Flynn P. Data Clustering: A Review. ACM Computing Surveys, 1999, 31(3): 264-323.

② Jain A. Data Clustering: 50 Years Beyond K-means. Pattern Recognition Letters, 2010, 31(8): 651-666.

③ 也称为“惯性”(inertia).

④ Aloise D, Deshpande A, Hansen P. NP-hardness of Euclidean Sum-of-Squares Clustering. Machine Learning, 2009, 75: 245-248.

⑤ Jain A. Data Clustering: 50 Years Beyond K-means. Pattern Recognition Letters, 2010, 31(8): 651-666.

⑥ Celebi E. Improving the Performance of K-Means for Color Quantization. Image and Vision Computing, 2011, 29(4): 260-271.

图 5.3 K-means 算法的整个流程

在进行了最初的随机初始化后，K-means 会依照两个步骤进行逐次迭代。在第一步，算法会计算每个成员与所有聚类中心的距离，并将距离最近的聚类中心设定为该点的原型。当根据现有的聚类中心为所有成员分配了“标签”后，则进入第二步。此时，算法会把所有标签相同的成员视为同一群体，并计算这些群体的类型内均值作为新的聚类中心，对参数进行更新。在每次步骤一与步骤二完成后，K-means 会对迭代次数或者参数更新的幅度进行判断，当达成终止要求时，则算法停止并输出最终的聚类标签；反之，则继续重复步骤一与步骤二，直到达成终止条件。

K-means 是一种有效的算法。当数据本身的群体异质性足够强时，也就是数据的自然分组较为明显时，它在大概率上可以达到局部最优的结果。[①] 然而，K-means 仍然具有三个不足：

（1）在群体划分上对超球面形状的偏好；

（2）由于欧氏距离带来的异常值敏感；

（3）初始值会显著地影响聚类结果。[②]

针对以上不足，我们不仅可以通过多次运行模型以及剔除离群值解决，而且还可以通过选用其他聚类算法来替换。总之，K-means 仍然是研究者在处理聚类分析时的首选算法，在积极调整以及保持审慎的数据处理后，能为我们提供充分的启发。

5.3.2 其他原型聚类算法

原型聚类是聚类分析中最重要的一类方法，但其内部的分类较多。本节我们将为读者介绍其他主流的原型聚类方法。具体如下：

高斯混合模型（Gaussian Mixture Model，GMM）与 K-means 算法使用均值来刻画群体的原型不同，高斯混合模型使用概率分布来描述原型的特征。通过把数据结构拆分为多个高斯分布，该模型实际上对数据特征进行了建模，并最终通过预测的方式来获得样本从属于每个高斯分布的可能性。总的来说，该模型既能对模型复杂程度进行精细的人为设定，又能解决样本的多元归属问题，相较于传统的划分模型具有更强的灵活性与自由度。实际上，K-means 模型可以视作高斯混合模型在混合成分方差相等、且每个成员只能归属于同类时的特例。[③]

谱聚类（Spectral Clustering）也是近年来非常流行的聚类方法。在图论思想的引导下，

① Meila M. The Uniqueness of a Good Optimum for K-Means. Proceedings of the 23rd international conference on Machine learning(ICML06),2006.

② Celebi E, Kingravi H, Vela P. A Comparative Study of Efficient Initialization Methods for the K-Means Clustering Algorithm. Expert Systems with Applications,2013,40(1): 200-210.

③ 周志华. 机器学习. 北京：清华大学出版社，2016.

谱聚类首先是使用成员两两间的相似度方阵来替换特征矩阵，这就获得了具有图属性的数据。然后，借助拉普拉斯特征映射（Laplacian Eigenmaps）对图数据进行降维，该方法能很好地保留原始数据的相似性特征。最后，借助传统聚类方法，如 K-means 对转化后的数据重新聚类，来获得最终结果。与传统的聚类方法相比，谱聚类的主要优势在于其基于相似度方阵进行运算，可以较好地规避数据的稀疏问题；同时，由于其中使用了 LE 方法进行了降维，在处理大规模高维数据时有明显优势；最后，图论的引入使得谱聚类对于不同数据特性的适用性要更强。[①] 当然，谱聚类的问题在于当期望划分的群体数量较多时，效果可能并不理想，另外由于谱聚类倾向于给出体量均衡的划分，当不同群体间规模过于悬殊时则并不适用。

下面我们将以中国村庄的类型划分为例，详细介绍聚类算法如何帮助研究者基于量化数据识别农村发展过程中所可能存在的村庄类别分化。

5.3.3 案例 1：强基建与促发展——中国村庄发展的类型差异

众所周知，中国的村庄发展长期存在着不均衡的现象。为了减少这种发展不均衡，国家动用大量的财政力量，在基层农村大力开展基础设施建设，试图扭转农村发展不平衡的局面。既有研究已经发现，虽然政府推动的强基建政策确实产生了正向效应，但由于地方性问题，在农村的经济发展与基础设施建设间仍存在着不平衡问题。[②] 本案例基于中国劳动力动态调查（CLDS）2016 年的村庄数据，试图通过聚类分析对于现有理论发现的类型学进行检验。我们选择了经济发展水平与基础设施建设相关的变量作为分类的标准，在剔除缺失值与异常值后保留了 182 个有效的村庄样本。接下来，我们将基于几种原型聚类算法对中国村庄在经济发展与基础设施层面的自然类型进行识别：

1. 数据导入与预处理

```
1.    # 1.导入 pandas 库并创建别名 pd,后续我们使用 pandas 库中的 DataFrame 类作为数据容器
2.    import pandas as pd
3.    # 2.sklearn 库是一个泛用性很强的机器学习工具箱,提供大多数主流机器学习算法的实现
      # 从 sklearn 库导入 preprocessing 模块,用于后续数据预处理
4.    from sklearn import preprocessing
5.    # 3.从 sklearn 库中的 cluster 模块导入 KMeans 类
6.    from sklearn.cluster import KMeans
7.    # 4.从流形模块导入 TSNE 方法,TSNE 是一种降维算法,特别适用于将高维数据降至 2 维、3
      # 维,以便可视化。我们之后将借助 TSNE 的映射功能对高维数据的聚类结果进行可视化。
      # 当维数非常高时可以先进行 PCA,再使用 TSNE 降维并可视化
8.    from sklearn.manifold import TSNE
9.    # 5.sklearn 的 metrics 模块提供了大量模型评估工具,我们从中选用了三种常用的聚类结
      # 果评估指标:
10.   #     (1)轮廓系数取值在[-1,1],该数值越接近 1,则代表聚类效果越好
11.   #     (2)calinski-harabasz 指数没有值域限制,该数值越大代表聚类效果越好
12.   #     (3)davies-bouldin 指数越接近 0 代表聚类效果越好
13.   from sklearn.metrics import silhouette_score,calinski_harabasz_score,davies_bouldin_score
14.   # 6.从 matplotlib 中导入 pyplot 模块用于绘图,创建别名 plt
```

① Von Luxburg U. A Tutorial on Spectral Clustering. Statistics and Computing，2007，17：395-416.

② 梁玉成，刘河庆. 新农村建设：农村发展类型与劳动力人口流动. 中国研究，2015(1)：6-25.

```
from matplotlib import pyplot as plt
# 7.设置 pandas 的最大显示列数,行数
pd.set_option('display.max_columns', 50)
pd.set_option('display.max_rows', 20)
# 8.设置 pyplot 的绘图默认参数,使绘图正常显示
plt.rcParams['font.sans-serif'] = ['SimHei']                    # 使中文编码在图中正确显示
plt.rcParams['axes.unicode_minus'] = False                      # 使负号在图中正确显示
# 9.统一设置绘图参数: 透明度、大小、轮廓粗细、色谱。cmap 中具有与数值对映的颜色映
# 射,可视化时可以为不同的类别标签分别赋予颜色
plot_kwds = {'alpha' : 0.7, 's': 40, 'linewidths':0,'cmap':'Set3'}
# 10.读取 CLDS 中村庄的数据,查看数据规模,并使用 head 函数查看前 10 行的数据。将
CID2016 这一变量设置为索引,用于唯一标识数据
Data = pd.read_csv("ComTypeNew.csv",index_col = "CID2016")
Data.shape
Data.head(10)
# 11.部分变量间量纲差异较大,对数据进行标准化,以减少极端影响。标准化函数会洗掉索
# 引与列名,为了后续查看数据方便,我们从原始数据复制了这些信息
Data.astype(float)                  # 将数据转化为浮点数格式
DataDup = Data.copy()               # 暂存未标准化的数据,以便后续数据拼接
Data = pd.DataFrame(preprocessing.scale(Data),index = Data.index,columns = Data.
columns)
Data.head(10)
```

2. 聚类模型选择

```
# 1.对于 K-means 算法使用"肘部法则"选择参数。对于 K-means 算法而言,SSE 会随着类
# 别的增加而降低,但对于有一定区分度的数据,在达到某个临界点时 SSE 会得到极大改善,
# 之后缓慢下降,这个临界点就可以考虑为聚类性能较好的点,由于指标改善的曲线形似手
# 肘,故这种选择聚类中心数量的方法称为"肘部法则"。我们接下来计算了不同 n_clusters
# 下,K-means 结果的 SSE,并绘制了出来,见图 5.4
SSE = []                                        # 空列表,用于存放每次聚类结果的 SSE
for k in range(1, 9):
  clf = KMeans(n_clusters = k)                  # 更换不同的 k 值构建聚类模型
  clf.fit(Data)
  SSE.append(clf.inertia_)                      # 使用 inertia_接口获取 SSE
k = range(1, 9)
plt.xlabel('n_clusters')
plt.ylabel('SSE')
plt.plot(k, SSE,"o-",c = 'red')
plt.show()
# 从图 5.4 中我们发现,本例的"肘部法则"并不明显,我们接下来使用更为通用的聚类结果
# 评估指标来选取聚类中心数
# 2.我们综合考量轮廓系数、CH 值以及 DB 值来评估聚类结果的质量。我们同样对这些指标
# 随聚类中心数的变化进行了绘制,见图 5.5

def cluster_scores(n,data,label):
  s1 = silhouette_score(data,label) * 100     #越大越好(为了便于绘图进行了倍乘)
  s2 = calinski_harabasz_score(data, label)   #越大越好
  s3 = davies_bouldin_score(data, label) * 5  #越大越好(为了便于绘图进行了倍乘)
  return([n,s1,s2,s3])

scores = []
labels = []
for k in range(2, 10):
  clf = KMeans(n_clusters = k)                  #更换不同的 k 值构建聚类模型
```

```
    clf.fit(Data)
    label = clf.labels_
    scores.append(cluster_scores(k,Data,label))
    labels.append(label)
scores = pd.DataFrame(scores,columns=["n","s1","s2","s3"])
labels = pd.DataFrame(labels).T
# 累积绘制,将三条线画在一张图里
plt.plot(scores["n"],scores["s1"],"o-",alpha=0.7,c='green',label="S")
plt.plot(scores["n"],scores["s2"],"o-",alpha=0.7,c='blue',label="C-H")
plt.plot(scores["n"],scores["s3"],"o-",alpha=0.7,c='pink',label="D-B")
plt.legend()                                          #显示图例
plt.show()
# 综合考虑三种指标,我们选定 k=5 为最终的模型。但在社会科学中,我们应把模型的解释
# 性放在显要位置,我们进而对聚类结果进行可视化
```

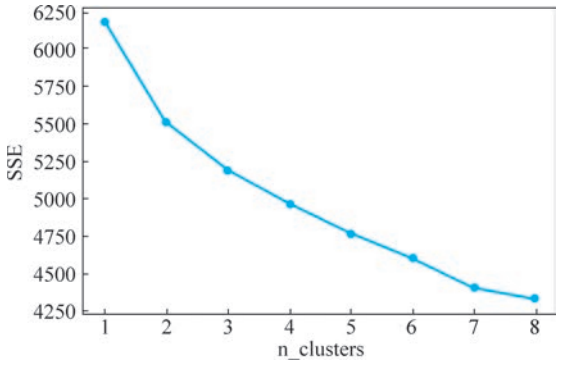

图 5.4 不同 n_clusters 下,K-means 结果的 SSE

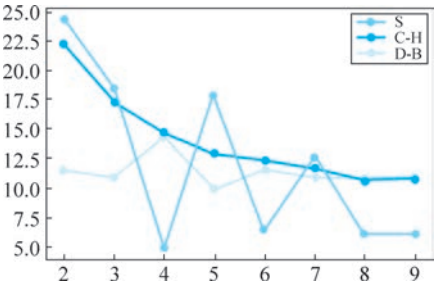

图 5.5 不同 n_clusters 下,K_means 结果的各指标变化

3. 聚类结果可视化

```
# 1.对于高维数据进行可视化是比较困难的,我们通常要用到降维的办法。比较常见的降维
# 办法如 PCA 并不一定能把数据映射到适合进行图示的维度。近年来比较流行的一种降维
# 算法 TSNE 则可以比较好地完成这个任务。TSNE 的主要参数是 n_component 和 perplexity,
# 前者决定目标维数(可选 2 或 3),后者则会影响降维后聚簇形状的紧密性,该数值一般设定
# 在[5,50],也有经验值认为设定在样本量的 1/20 较好。TSNE 的映射每次不尽相同,一般能
# 较好地展示出聚簇边界即可
tsne = TSNE(n_components=2,perplexity=100,n_iter=3000)
proj = tsne.fit_transform(Data)
# 2.由于 TSNE 的计算量通常较大,一般在获得了满意的映射后,我们可以将 TSNE 的系数进行
# 保存,以便日后的可视化
DataDup['tsne_x'] = proj.T[0]
DataDup['tsne_y'] = proj.T[1]
DataDup.to_csv("ComTypeTsne.csv")
# 3.我们接下来通过不同形状来标识聚类结果,见图 5.6
marker_list = ["X","v","h","o","s"]
label = labels[4].ravel()
DataDup['label'] = label
for i in range(0,5):
    data = DataDup[DataDup['label']==i]
    plt.scatter(data['tsne_x'],data['tsne_y'],marker=marker_list[i],
                alpha=i*0.2+0.05,s=40,c="black",label="cluster{}".format(i))
plt.legend()
```

```
plt.show()
# 4.在聚类结果进行存储。由于我们事先进行了标准化,变量数值不再有意义。我们可以根
# 据生成的类别对原始数据计算分组均值,来看类簇的行征
DataDup['ClusterLabel'] = labels[3].ravel()
DataDup['ClusterLabel'].value_counts()
DataDup.to_csv("ComTypeLabel.csv")
DataDup.groupby('ClusterLabel').mean()
Result = pd.DataFrame(Data.groupby('ClusterLabel').mean())
Result.to_csv("ComTypeClusterResult.csv")
```

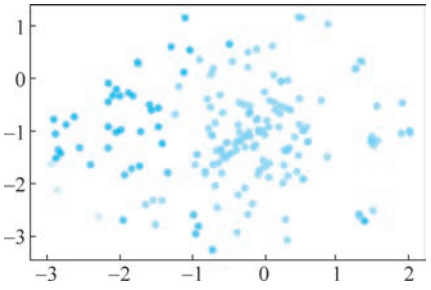

图 5.6 聚类结果染色示意图

结合既有发现,我们根据以上所生成的概览表对聚类结果进行解读。我们不难发现,经济水平高、基础设施较好型的村庄数据最多(127 个),同时由于国家对农村的投入,经济水平一般但基础设施建设好的村庄数量同样较多(25 个),这印证了既有的经验发现[①]。

5.4 密度聚类与层次聚类

在上一节中,我们主要对原型聚类方法进行了系统介绍。然而,除了假定数据中存在作为典型案例的原型聚类外,我们还要考虑到数据疏密的密度聚类,以及数据样本存在层级化的树状结构的层次聚类。

5.4.1 密度聚类

密度聚类认为,如果数据中存在潜在群体就会呈现出数据密度的变化。由此,理论上只要从一个点出发,保持对密度的监控,并进行穷举的搜索,就有可能对数据中所有的群体进行发现,即完成群体划分。在具体的算法实现上,它基于数据密度来测量样本间的可连接性,连续拓展同类的群体来完成聚类任务。[②] 最经典的密度聚类算法当属 DBSCAN 算法,它擅长在具有噪声的大规模数据中高效地发现潜在的群体,尤其出色的是该算法可以发现任何形状的群体。[③] 相比于大多数的原型聚类方法,以 DBSCAN 为代表的密度聚类算法并不需要指定潜在的原型数量,而这在大多数研究中通常是个难题。

当然,DBSCAN 同样也依赖于研究者对某些参数的指定。正如它基于密度搜索的思路,研究者控制算法判定样本间是否具有连接性的标准。DBSCAN 假定当两个样本共存在

① 梁玉成,刘河庆.新农村建设:农村发展类型与劳动力人口流动.中国研究.2015(1):6-25.

② 周志华.机器学习.北京:清华大学出版社,2016.

③ Ester M, Kriegel H P, Sander J, et al. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96), 1996: 226-231.

具有一定数据密度的区域内时，它们之间就是相互连接的，因而我们使用一组邻域参数(ε，MinPts)来限制数据密度。这组参数不仅限定了以 ε 为半径的圆形区域，而且保证该圆形区域保持连接性时具有的最少样本数 MinPts。也就是说，我们借助邻域参数规定了搜索过程继续进行的密度下限。在限定了邻域参数后，DBSCAN 会以数据集中每个样本为圆心画半径为 ε 的圆，并把圆周内点的数量大于等于 MinPts 的样本标记为核心对象，并以核心对象为起点继续画圆来“寻找同类”，直到连续搜索到所有可及的成员，并将它们划入同一群体(见图 5.7[①])。当然，密度聚类在抗噪声和不限定群体形状上表现优异，但在处理高维数据时面临较大挑战。

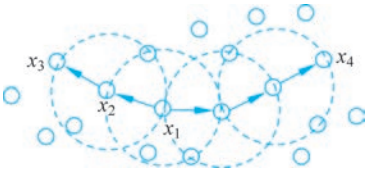

图 5.7 父母教养方式 K-means 聚类在不同 n-clusters 下的 SSE 结果

5.4.2 层次聚类

层次聚类在面对数据时假设数据本身具有层级结构，进而采用树状来刻画数据的群体划分。该算法通常分为由上而下的分裂式与由下而上的凝聚式两种：前者的代表算法是 DIANA 算法，先将所有成员划分入同一群体，在采取分拆的方式逐渐生成子群；后者的代表算法则是 AGNES 算法，它从单个成员为起点，根据成员间距离的远近逐渐合并，直到所有样本都合并到一类。对层次聚类而言，一个非常有趣的做法是对拆分或合并的过程绘制完整的树状图(Dendrogram)。图 5.8 展示的是基于 CLDS 的 AGNES 算法聚类结果。

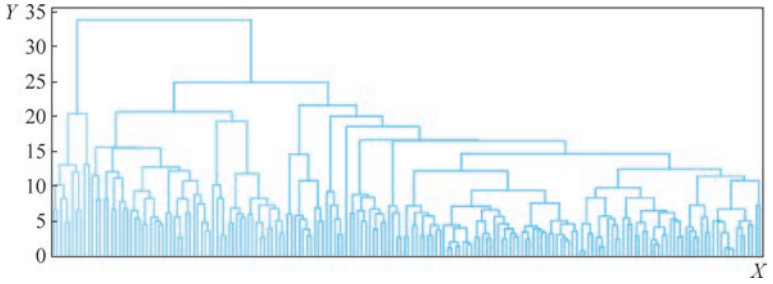

图 5.8 AGNES 算法中凝聚过程的树状图展示

那么，如何来理解层次聚类的树状图呢？首先，我们看到从 X 轴向上伸展出了竖线。在最初阶段，这些竖线并未合并，每条竖线都代表了数据集中一个独立成员。竖线开始向上延伸，Y 轴的坐标代表进行凝聚时同类样本间的最大距离，显然，由于聚类将一定距离内的样本都聚合进一类，伴随着该距离的升高，所有样本都会被逐渐吸纳进同一类中。每当子群体进行合并，树状图都会以横线连接子群体并合入一条竖线。最后，我们通过“剪树”来获得可用的聚类结果，在特定的距离值上横向“切一刀”，有多少竖线断裂就分作了几类。

层次聚类的优势之一是结果可视性强。借助于树状图，我们可以很好地看到随着距离变化，组与组间的合并/拆分过程，并进行适当的“剪树”。另外，在样本量不大时，我们借助树状图可以发现凝聚过程中的异常点。当出现凝聚距离很大时才被收纳的孤立点，我们就有理由认为这是一个奇异值，可以从聚类过程中剔除以提高聚类质量。最后，由于层次聚类的树状结构，我们有可能解读出不同层级群体间更深层的理论关系，这对于社会科学研究而

① 周志华. 机器学习. 北京：清华大学出版社，2016.

言尤为重要。

5.4.3 案例 2：关注与忽略——中国家庭教育的不同模式

对于每个中国家长而言，家庭教育都是个“欲说还休”的话题。自古以来就有“万般皆下品，唯有读书高”的观念，这也使得教育在中国社会处于尤为重要的位置。虽然都认同教育的重要性，但父母对教育的关注方式仍然存在很大的差异，是传统的“严师出高徒”，还是西式的“放任得自由”呢？是……还是……？本案例试图利用聚类分析来回答中国家庭教育有哪些不同模式。我们使用的是 2014—2015 年中国教育追踪调查（CEPS）数据，从中筛选了学生和家长问卷中与教育方式有关的变量[①]，接下来让我们利用聚类分析来探究中国家庭教育的基本类型。

1. 数据导入与预处理

```
# 1.与案例 1 相似，我们同样使用 pandas 库中的 DataFrame 类作为我们的数据容器。同时我
# 们从 sklearn 库中导入 preprocessing 模块用于数据预处理
import pandas as pd
from sklearn import preprocessing
# 2.本节我们将引入其他聚类算法来比较他们的差异。在 K-means 之外，我们还将简单介绍
# 高斯混合模型以及层次聚类。我们从 sklearn、scipy 库中导入这些模块。sklearn 同样提
# 供了层次聚类方法（见 AgglomerativeClustering）。这里换用 scipy 库是为了便于树状图
# 的绘制
from sklearn.cluster import KMeans
from sklearn.mixture import GaussianMixture
from scipy.cluster import hierarchy
# 3.导入 PCA 与 TSNE 用于数据可视化。导入轮廓系数、CH 指数、DB 指数用于聚类效果测量
from sklearn.decomposition import PCA
from sklearn.manifold import TSNE
from sklearn.metrics import silhouette_score ,calinski_harabasz_score,davies_bouldin_score
# 4.从 matplotlib 库中导入 pyplot 模块用于数据可视化，并创建别名 plt
from matplotlib import pyplot as plt
# 5.如案例一进行基本的 pandas、pyplot 的参数配置
pd.set_option('display.max_columns', 50)                    # 配置 pandas 的最大显示列数
pd.set_option('display.max_rows', 20)                       # 配置 pandas 的最大显示行数
plt.rcParams['font.sans-serif'] = ['SimHei']                # 使中文编码在图中正确显示
plt.rcParams['axes.unicode_minus'] = False                  # 使负号在图中正确显示
plot_kwds = {'alpha' : 0.15, 's': 40, 'linewidths':0}       # 统一设置绘图参数
# 6.读取 CEPS 中有关子女教养方式的数据，查看数据规模，并使用 head 函数查看前 10 行的
# 数据。将 ids 这一变量设置为索引，用于唯一标识数据
Data = pd.read_csv("Data/EduPat.csv",index_col = 'ids')
Data.shape
Data.head(10)
# 7.部分变量间量纲差异较大，对数据进行标准化，以减少极端影响。标准化函数会洗掉索
# 引与列名，为了后续查看数据方便，我们从原始数据复制了这些信息
Data.astype(float)              # 将数据转化为浮点数格式
DataDup = Data.copy()           # 暂存未标准化的数据，以便后续数据拼接
Data = pd.DataFrame(preprocessing.scale(Data),index = Data.index,
                    columns = Data.columns)
Data.head(10)
```

① 网址为 http://ceps.ruc.edu.cn/。本案例使用的 47 个变量在 CEPS 数据中对应的变量名为 ids、w2a18、w2a19、w2a2001-w2a2006、w2a2101a、w2a2101b、w2a2102a、w2a2102b、w2a2103a、w2a2103b、w2a2104a、w2a2104b、w2a22-w2a29、w2ba04、w2ba05、w2ba11～w2ba13、w2ba1701～w2ba1706、w2ba18-24、w2ba2601、w2ba2604、w2ba29。

2. 聚类模型选择

```
# 1.我之前已经展示过了 K-means 算法的基本参数选择办法,也就是,通过计算不同聚类中心
# 的 SSE 值,并通过"肘部法则",挑选位于"肘部"的取值。当然,"肘部法则"有时并不明显,
# 此时我们可以结合通用的聚类评估指标、可视化结果以及聚类结果的可解释性来综合选择
# 参数(这适用于大部分聚类算法)。我们来看一下对 CEPS 的父母教养方式数据进行 K-means
# 聚类的结果,见图 5.9
```

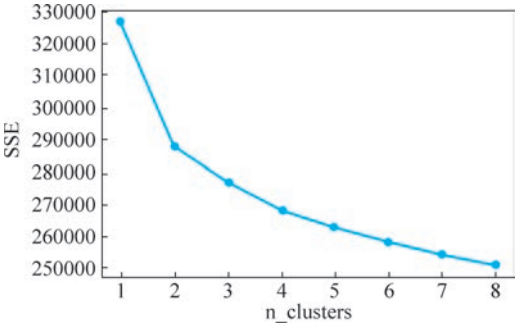

图 5.9 父母教养方式 K-means 聚类在不同 n-clusters 下的 SSE 结果

```
SSE = []                                    # 空列表,用于存放每次聚类结果的 SSE
for k in range(1, 9):
  clf = KMeans(n_clusters = k)              # 更换不同的 k 值构建聚类模型
  clf.fit(Data)
  SSE.append(clf.inertia_)                  # 使用.inertia_接口获取 SSE
k = range(1, 9)
plt.xlabel('n_clusters')
plt.ylabel('SSE')
plt.plot(k, SSE,"o-",c = 'red')
plt.show()
# 根据"肘部法则",我们大概可以确定 n = 2 或 n = 4 时,聚类结果可能有更好的表现
# 2.我们接下来用高斯混合模型(GaussianMixtureModel)来对我们的数据进行聚类。高斯混
# 合模型将我们的观测数据视为 n 个多元高斯分布的混合,因而他实际上是一种基于概率的
# 估计算法。在完成了模型的估计后,我们可以通过预测的方式来确定样本的类别,这就达
# 到了聚类的效果。通常使用 BIC 准测来评估高斯混合模型的拟合质量。(我们在社会统计
# 中也常用 BIC 来评估 Logistic 回归模型拟合的好坏)与另一种常用的指标 AIC 准则(评估
# 预测能力)相比,BIC 准则(评估拟合质量)更偏好简约(参数较少)的模型而已
BIC = []                                    # 空列表,用于存放每种 GMM 的 BIC 准则数值
for k in range(1, 9):
  clf = GaussianMixture(n_components = k)   # 更换不同多元高斯分布数量 k 值来构建模型
  clf.fit(Data)
  BIC.append(clf.bic(Data))                 # 使用.bic(data)接口获取 BIC 准则数值
k = range(1, 9)
plt.xlabel('n_components')
plt.ylabel('BIC')
plt.plot(k, BIC ,"o-",c = 'purple')
plt.show()
# 观察 GMM 拟合结果的 BIC 准测变化,见图 5.10。我们可以看到 n = 2 或 n = 5 时,聚类结果
# 可能有更好表现
# 3.我们最后介绍层次聚类(Hierarchy Clustering)。层次聚类是一种从下而上的聚类方
# 法,其根据点与点间的距离由小到大,逐渐生成小的类簇并对类簇进行合并,直到所有的类
# 簇合并为一类,见图 5.11。在实际使用中,层次聚类能比较好地识别出离群值。在"剪数"
# 时,通常对于点间距离接近的类簇,应该尽可能都划分出来。当样本量较大时,我们一般不
# 绘制完全的树状图,此时设置 dendrogram()中的参数 truncate_mode = 'lastp'并设置 p 值
```

```
    # 为我们想要观察的合并节点数量
55. clf3 = hierarchy.linkage(Data, method = 'ward',metric = 'euclidean')
56. hierarchy.dendrogram(clf3,truncate_mode = 'lastp',p = 40,no_labels = True)
57. plt.show()
58. # 后续为了获得聚类的类别,我们需要对“剪”树,也就是在一定的高度(height)对树形图进
    # 行截断来获得类别。我们想象在树状图上横切一刀,划破几根竖线就会分出多少类别。在
    # 本例中,我们选择 height = 100 作为聚类结果,此时会分出四个聚簇
```

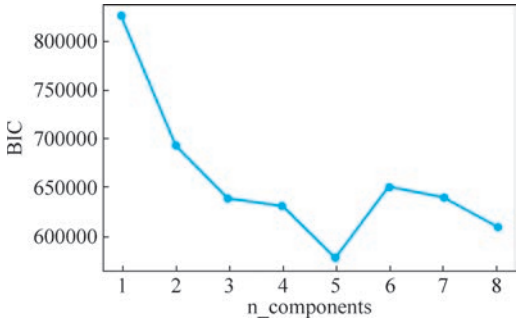

图 5.10 不同 n_components 下模型的 BIC 值

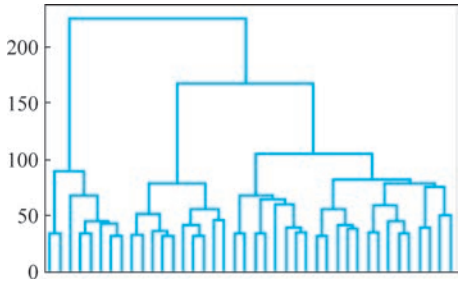

图 5.11 层次聚类示意

3. 聚类结果可视化

```
59. # 1.接下来,我们首先获取了这些模型的聚类结果
60. # K-means 算法获得的最佳聚类结果
61. clf1 = KMeans(n_clusters = 4)
62. clf1.fit(Data)
63. clf1_labels = clf1.labels_.ravel()
64. # GMM 算法获得的最佳聚类结果
65. clf2 = GaussianMixture(n_components = 5)
66. clf2_labels = clf2.fit_predict(Data).ravel()
67. # GMM 算法获得的最佳聚类结果
68. clf3_labels = hierarchy.cut_tree(clf3 ,height = 150).ravel()
69. # 2.我们接着使用几个通用的聚类结果评估指标对聚类结果进行评估。我们定义了函数用
    # 来打印这些指标
70.
71. def cluster_scores(method,n_clusters,data,label):
72.     s1 = silhouette_score(data,label)                    #越大越好
73.     s2 = calinski_harabasz_score(data, label)            #越大越好
74.     s3 = davies_bouldin_score(data, label)               #越大越好
75.     print(method,"n_clusters:",n_clusters)
76.     print("silhouette_score:", s1,"calinski_harabasz_score:", s2,"davies_bouldin_
        score:",s3)
77.
78. cluster_scores("kmeans",4,Data,clf1_labels)
79. cluster_scores("gmm",5,Data,clf2_labels)
80. cluster_scores("hierarchy",3,Data,clf3_labels)
81. # 3.TSNE 降维效果好,但计算量很多。由于变量较多,我们结合 PCA 与 TSNE 对数据进行可视
    # 化,来降低计算的时间成本
82. from sklearn.decomposition import PCA
83. pca = PCA(n_components = 30,svd_solver = 'full')
84. proj = pca.fit_transform(Data)
85. pca.explained_variance_ratio_.sum()                    # 计算总的解释方差
86. proj.shape
87. tsne = TSNE(n_components = 2,perplexity = 300,n_iter = 2000,n_jobs = -1)
```

```
    # 在一些情况下,n_jobs = -1 可能导致错误,此时可以删去这一参数
    proj = tsne.fit_transform(proj)
    # 4.在结合 PCA 与 TSNE 的降维图上,对于 3 种聚类结果进行可视化,见图 5.12
      marker_list = ["o","X","v","d","s"]
    def label_plot(method,n_cluster,label):
      plot = pd.DataFrame(proj,columns = ['tsne_x','tsne_y'])
      plot['label'] = label
      plot.figure(figsize = (10,10))
      plt.title(method + " " + "n_cluster = " + str(n_cluster))
      for i in range(0,n_cluster):
          data = plot[plot['label'] = = i]
          plt.scatter(data['tsne_x'],data['tsne_y'],marker = marker_list[i]),
                    alpha = i * 0.2 + 0.05,s = 10,c = "black",label = "cluster{}".format(i))
      plt.legend()

    label_plot("kmeans",4,clf1_labels)
    label_plot("gmm",5,clf2_labels)
    label_plot("hierarchy",3,clf3_labels)
    # 从图 5.12 中我们可以大概看到几种聚类算法获得的聚类结果的特点,显然 K-means 更偏
    # 好给出形状更为规整的聚簇,而 GMM 则倾向于给出复合的聚簇,层次聚类在大部分情况下
    # 与 K-means 结果相近
    # 5.最后,我们保存这些降维与聚类的结果。为了对结果进行解释,我接着计算聚类结果的
    # 类内均值,对这些聚簇进行描述
    DataDup['tsne_x'] = proj.T[0]
    DataDup['tsne_y'] = proj.T[1]
    DataDup['KmeansLabel'] = clf1_labels
    DataDup['GmmLabel'] = clf2_labels
    DataDup['HierarchyLabel'] = clf3_labels
    DataDup.to_csv("Data/EduPatLabel.csv")
```

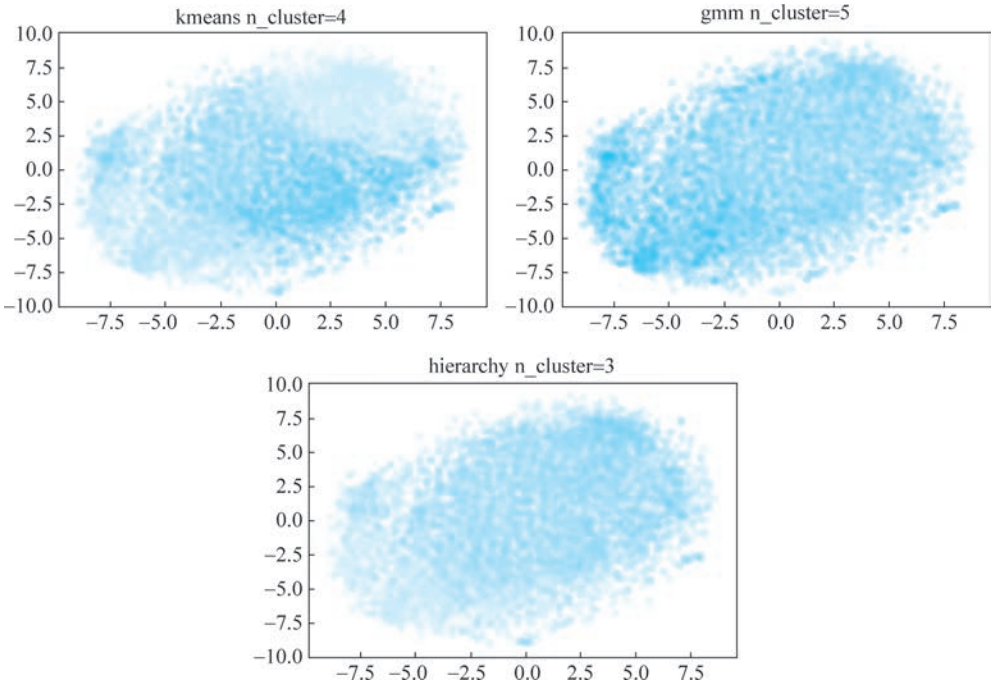

图 5.12 不同聚类方法下聚类结果的染色比较

综合考量我们的理论关照以及可视化结果，我们最后选择 K-means 算法获得的 $k=4$ 的模型用于我们最后的理论解释。针对“陪伴”“沟通”“管教”以及“期望”对父母教育子女模型进行分组后，我们发现中国父母的教育模式在方式上分为严格与放任型，另一个核心差异在于期望的高低，最终是一个二维的类型学模型。同样的，当大家获得如用户在知乎问答、豆瓣影视评价以及传感器中的行为信息等特征维度较高的数据时，也可以将上述介绍的聚类方法用于用户子群体识别、用户社群变迁等任务中，助力计算社会科学研究。

5.5 聚类分析展望

在聚类分析中，当定义了恰当的距离度量后，我们就可以使用相关算法来获得异质性群体的划分，并可以通过性能评估指标、结果启发性以及可解释性来选择最终的模型。具体到聚类算法的选择，其实没有一定之规。由于数据结构间的巨大差异，大部分聚类算法在性能上的差异其实并不大，关键在于我们的研究目的以及数据特征。我们往往针对数据结构来选择恰当的算法，要衡量的因素可能包括：数据体量、特征数量、离散聚集趋势以及稀疏性等等。在这点上来说，是否能够从特定数据集中发现真正有趣的聚类，对聚类分析更为关键。在研究中，我们大可放手尝试各种类型的聚类算法，在理解算法适用的状况后，以性能评估指标作为客观参照，再结合研究经验就能获得适切的模型。

近年来，聚类分析的算法发展的越发深入，在适应大规模非结构化数据上有了长足的进展。[①] 但大多数传统聚类方法仍然对聚类对象中的结构和关系重视不足。相对而言，更具灵活性的潜在语义分析（Latent Semantic Analysis，LSA）以及隐含狄利克雷分布模型（Latent Dirichlet Allocation，LDA），也都提供了提炼文本深层结构的初步方法。前者基于 SVD 分解，可以将词频数据投射到维度更低的空间，以达到提炼语义的目的；后者则通过贝叶斯方法获取词袋数据中的词汇在文档中的共现关系，以提取文档的主题，这些都已经在计算社会科学领域得到了广泛应用。[②③]

此外，伴随着深度学习的发展，表示学习（Representation Learning）愈发重要，借助自编码器（Autoencoder）的广泛应用，研究者可以对具有复杂特征数据进行更深入的分析，而无须担心过量的信息损耗。[④] 以词向量（word2vec）为代表的文本特征自动提取技术，利用了自编码器的优势，很好地保留文本数据中的语境信息，为我们理解社会与文化提供了巨大的支持。[⑤] 目前，国内外一些研究已经开始利用词向量模型来开展计算社会科学研究。[⑥⑦]

① Jain A. Data Clustering：50 Years Beyond K-means. Pattern Recognition Letters，2010，31(8)：651-666.

② Grimmer J. Appropriators Not Position Takers：The Distorting Effects of Electoral Incentives on Congressional Representation. American Journal of Political Science，2013，57(3)：624-642.

③ 黄荣贵. 网络场域、文化认同与劳工关注社群：基于话题模型与社群侦测的大数据分析. 社会，2017(2)：26-50.

④ Xie J W，et al. Representation Learning：A Statistical Perspective. Annual Review of Statistics and Its Application，2020，7：1-33.

⑤ Evans J A，Aceves P. Machine Translation：Mining Text for Social Theory. Annual Review of Sociology，2016，42：21-50.

⑥ Garg N，Schiebinger L，Dan J，et al. Word Embeddings Quantify 100 Years of Gender and Ethnic Stereotypes. Proceedings of the National Academy of Sciences，2018，115(16)：3635-3644.

⑦ 刘河庆，梁玉成. 政策内容再生产的影响机制：基于涉农政策文本的研究，社会学研究，2021(1)：115-136.

本章小结

聚类分析(Cluster Analysis)是一类从量化数据中自动提炼异质性群体的方法,可以帮助研究者探索和发现感兴趣数据集中的结构和模式。为更好地评估数据中的相似性和异质性,聚类分析需要引入距离度量来统一数据集内相似度的测算,对聚类结果进行评估则包括轮廓系数等常用指标。在对聚类分析的基础知识进行介绍后,本章分别重点对 K-means 为代表的原型聚类以及密度聚类、层次聚类等主要聚类算法的基本原理进行介绍,并结合 CLDS 村庄数据以及 CEPS 数据,分别对中国村庄发展的类型以及中国家庭教育模式进行了探索和分析。

聚类分析是传统量化研究方法的重要组成部分,它常常被用于发现调查数据中新社会群体、新社区类型和新治理模式等。在数字社会,社会科学所拥有数据的变量越来越丰富和多样,仅使用几个变量来确定数据中的群体分类并不现实。如何挖掘数以百万乃至数以千万计的文本数据、图片数据和视频数据中的群体类属,并发现新的有趣的群体类别是计算社会科学家的重要研究内容,而以聚类分析为代表的无监督机器学习方法无疑将发挥基础作用。

习 题 5

1. 在本章案例 1 中,选择了 CLDS2016 村庄数据中的经济发展水平与基础设施建设相关的变量作为分类的标准,你认为还有哪些变量可以进一步作为村庄分类的标准?原因是什么?请进一步思考不同类型村庄中的人口迁移模式存在何种差异?

2. 使用不同的聚类算法、不同的参数设置对你感兴趣的群体进行划分,选出你认为最好的模型,对各个类别进行进一步描述和解读。

3. 请使用聚类分析方法,尝试对文本数据(如人民日报文本数据、中国知网上的文献摘要数据等非结构化数据)进行分析,并比较上述聚类方法在分析结构化数据和非结构化数据中的异同。