

k-means聚类算法

大连理工大学模式识别与机器学习研究小组
(此版为初稿, 欢迎提出修改意见!)

2019年8月6日

第三章 k-means聚类算法

聚类是一个古老的研究问题，在《周易》中就有“方以类聚，物以群分”的记载。在人类的文明发展中，人类在同大自然的互动与反馈中学习大自然的规律，也正是通过这种无监督的学习方式，人类的文明与知识也从无到有建立起来。在本章中，我们将重点讲解一种无师自通的学习方法，这就是聚类分析。在所有的聚类方法中，k-means聚类算法是一种最基础的聚类方法，其聚类的过程包含了聚类最基本的思想，通过对k-means聚类算法的学习，可以了解聚类最一般性的原理。在余下的内容中，我们首先阐述聚类的一般性问题；其次，重点介绍k-means聚类算法，并分析了k-means聚类算法的优点和缺点；再次，我们介绍了几种针对k-means算法的扩展算法，最后我们对本章的内容进行了总结。

0.1 聚类问题的阐述

在我们日常生活中，分类问题经常接触到。例如：我们可以依据攻读的学位把全校的学生分为本科生、硕士生和博士生，我们也可以依据性别将全校的学生划分为男生和女生。也就是说，我们在对全校的学生进行划分之前，事先知道了要将全校的学生划分为本科生、硕士生和博士生或男生和女生。由此可见，分类是一种有监督的学习过程，即实现知道事物的类标签。聚类不同于分类，在执行聚类任务时我们并不知道事物的类标签，也就是说，对于将全校的学生进行划分这件事，我们并不知道要将全校学生划分为本科生、硕士生和博士生或男生和女生，我们只能依据学生之间的相似或相异程度，来把相似的学生划分到同一类，把不相似的学生归划分到不同类；至于每个类簇所代表的实际意义，我们事先并不知道，因此，聚类是一个无监督的学习过程。

对于聚类的基本概念，Han等人给出了形式化的定义 [1]：聚类是把一个对象集划分成子集的过程，每个子集称着为簇，使得簇内的对象具有很高的相似性，但与其他簇中的对象不相似。图1展示了一个数据集的聚类结果。

聚类尽管是一种无监督的学习方法，但是却有巨大的应用价值。例如，Facebook每天处理的数据超过500 TB，阿里巴巴拥有的数据量超过100 PB(1 PB=1024 TB)，新浪微博用户数超过5亿，每天产生的微博数超过1亿条。然而，由于现实条件的限制，获取这些海量数据的类标签往往是一件十分困难的事，即使利用专家的领域知识来对数据进行标记也不是一件不容易的事，因为人工标记数据的效率可能远远无法满足现实的需要，

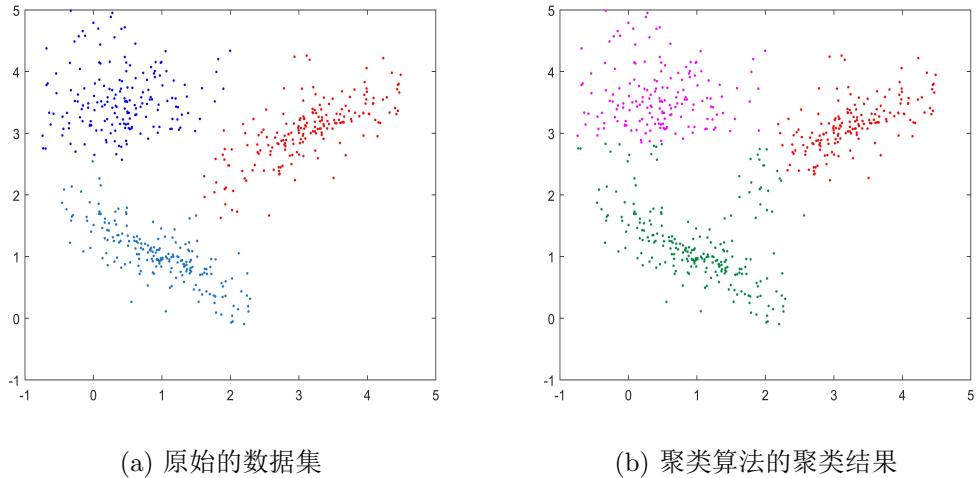


图 1: 数据集的聚类结果示意图

而且可能产生的巨大代价也难以让人承受，因此如何利用无标记的数据来指导机器学习过程，是一个受到广泛关注的问题。在2011年，美国《科学》杂志推出“数据处理”专刊，重点阐述了数据处理的基本方法和对社会发展的影响；这一切都显示了聚类分析对我们的生活产生了重要的影响。

到目前为止，广大学者们对聚类问题进行了深入地研究，并提出了许多的聚类算法[2]。依据这些聚类算法的特点，这些聚类算法主要可以划分成四类：

(1) 基于划分的聚类算法，也就是把数据划分为 k ($k \geq 2$)个簇，每一簇至少包含一个对象，最典型的即是本章重点阐述的k-means算法[3]；

(2) 基于密度的聚类算法，利用样本点周围点的密度来发现任意形状的簇，代表性的算法有DBSCAN算法[4]、CFSFDP算法[5]；

(3) 层次聚类算法，根据数据集的分裂顺序可以分为自顶向下的聚类算法和自底向上的聚类的算法；对于自顶向下的聚类算法，即最先把整个数据集看做一个类簇，然后将整个数据集不断分裂为子集，直至子集符合终止条件为止；对于自底向上的聚类算法，最先将每一个数据点看做一个类簇，不断合并相似的类簇，直至所获得的聚类结果符合终止条件为止；代表性的层次聚类算法有BIRCH算法[6]、Chameleon算法[7]；

(4) 基于网格的聚类算法，即把数据所在的空间划分成一个个网格，聚类都在这个网状结构上进行，通过网格内样本的统计信息来发现类簇，代表性的算法有STING算法[8]、CLIQUE算法[9]。

由此可见，聚类算法多种多样，每一种算法都有不同的特点，也有其自身适用的场景，因此针对同一数据集，不同的算法可能会得出不同的聚类结果。同时，这也表明，聚类问题是一个开放性问题，需要依据具体的问题来设计或选择合适聚类算法，我们也很难找出一种聚类算法能够适用所有问题，所以，聚类领域具有非常多需要进一步研究的问题。

0.2 k-means算法的基本原理

0.2.1 数据间相似性度量

由上一节的内容可以知道，k-means算法是要把彼此相似的样本划分到一起，因此，如何度量数据间的相似程度是包括k-means算法在内的所有聚类所面临的一个重要问题。为了阐述这个问题，我们引入如下的例子。在表1中，我们列出了几份诊断报告，其中 $\{s_1, s_2, s_3, s_4\}$ 为4名患者，而 $\{WBC, HGB, PLT, RDW, MCV, MCH, HCT\}$ 为相应的生化检测指标。

表 1: 几份诊断报告的结果

人物/指标	WBC	HGB	PLT	RDW	MCV	MCH	HCT
s_1	11.80	125	237	12.8	91.0	30.5	37.30
s_2	11.35	153	284	13.8	74.0	23.9	47.62
s_3	9.47	173	237	12.4	89.5	30.3	51.20
s_4	8.70	135	405	17.8	67.0	22.0	41.50

如果要分析这四位患者的病情，必定会面临如何度量这四位患者病情的相似程度。为了便于数学表达，我们可以用向量来表示患者，例如，对于 s_1 患者，我们记表示的向量为 $s_1 = \{11.80, 125, 237, 12.8, 91.0, 30.5, 37.30\}$ ；同理， $s_2 = \{11.35, 153, 284, 13.8, 74.0, 23.9, 47.62\}$ ， $s_3 = \{9.47, 173, 237, 12.4, 89.5, 30.3, 51.20\}$ ， $s_4 = \{8.70, 135, 405, 17.8, 67.0, 22.0, 41.50\}$ 。因此，对于 s_1, s_2, s_3, s_4 之间的相似性，即可以转化为向量之间的相似性。在高中阶段，我们已学习过，对于任意两个向量 $\mathbf{x} = [x_1, x_2, \dots, x_m] \in \mathbb{R}^{1 \times m}$ 与 $\mathbf{y} = [y_1, y_2, \dots, y_m] \in \mathbb{R}^{1 \times m}$ 而言，它们之间的相似性可以用向量之间的夹角余弦来表示，即相似度 $sim(\mathbf{x}, \mathbf{y})$ 可以表示为：

$$sim(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{x}\| \cdot \|\mathbf{y}\|} \quad (1)$$

其中 $\mathbf{x} \cdot \mathbf{y}$ 表示向量 \mathbf{x} 与 \mathbf{y} 的内积， $\|\mathbf{x}\|$ 与 $\|\mathbf{y}\|$ 分别表示向量 \mathbf{x} 与 \mathbf{y} 的模。 $sim(\mathbf{x}, \mathbf{y})$ 值越大，表示向量 \mathbf{x} 与向量 \mathbf{y} 就越相似。(1)式中的方法所获得相似度，也被称为余弦相似度。在实际应用中，除了余弦相似度，基于距离的相似度量方法也比较常见，即用两个向量之间的距离来度量相似程度。基于距离的相似度量所用的距离主要有如下几种：

(1) 欧几里得距离(Euclidean Distance):

$$d(\mathbf{x}, \mathbf{y}) = \left(\sum_{i=1}^m (x_i - y_i)^2 \right)^{\frac{1}{2}} \quad (2)$$

(2) 曼哈顿距离(Manhattan Distance):

$$d(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^m |x_i - y_i| \quad (3)$$

其中: $|x_i - y_i|$ 表示 $x_i - y_i$ 的绝对值。

(3) 切比雪夫距离(Chebyshev Distance):

$$d(\mathbf{x}, \mathbf{y}) = \lim_{p \rightarrow \infty} \left(\sum_{i=1}^m (x_i - y_i)^p \right)^{\frac{1}{p}} = \max_i (x_i - y_i) \quad (i = 1, 2, 3, \dots, m) \quad (4)$$

(4) 马氏距离(Mahalanobis Distance):

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{(\mathbf{x} - \mathbf{y}) \Sigma^{-1} (\mathbf{x} - \mathbf{y})^T} \quad (5)$$

其中: Σ 为随机变量 \mathbf{x} 与 \mathbf{y} 所服从同一分布的协方差矩阵。

(5) 明可夫斯基距离(Minkowski Distance):

$$d(\mathbf{x}, \mathbf{y}) = \left(\sum_{i=1}^m |x_i - y_i|^p \right)^{\frac{1}{p}} \quad (6)$$

其中: p 为一个参数; 当 $p=1$ 时, (6) 式为曼哈顿距离, 当 $p=2$ 时, (6) 式为欧几里得距离, 当 $p \rightarrow \infty$ 时, (6) 式则为切比雪夫距离。

在以上的几种距离中, 欧几里得距离由于形式简单, 且便于计算和求导, 在实际应用中使用得最为广泛。除了以上几种距离, 对于任意一个非空集合 \mathbf{X} , \mathbf{X} 中任意两点 $x, y \in \mathbf{X}$, 都有一实数 $d(x, y)$ 与之对应且满足:

(1) 非负性: $d(x, y) \geq 0$, 当且仅当 $x = y$ 时, 有 $d(x, y) = 0$;

(2) 对称性: $d(x, y) = d(y, x)$;

(3) 传递性 (三角不等式): 对于 $\forall z \in \mathbf{X}$, 都有 $d(x, z) \leq d(x, y) + d(y, z)$ 。

凡是符合以上定义的 $d(x, y)$ 均被称为距离, 且都可以用来度量样本之间的相似或相异程度。例如: $f(\mathbf{x}, \mathbf{y}) = \exp(-(\mathbf{x} - \mathbf{y}) \cdot (\mathbf{x} - \mathbf{y})^T)$ 也满足非负性、对称性和传递性, 因此, $f(\mathbf{x}, \mathbf{y})$ 也可以看做一种距离的度量。

由表1可知, 如果采用欧几里得距离, 单纯只计算样本之间的距离, 而不考虑其他方面问题, 则有:

$$d(\mathbf{s}_1, \mathbf{s}_2) = \sqrt{0.45^2 + (-28)^2 + (-47)^2 + (-1)^2 + 17^2 + 6.6^2 + (-0.32)^2} = 57.6790;$$

$$d(\mathbf{s}_1, \mathbf{s}_4) = \sqrt{3.1^2 + (-10)^2 + (-168)^2 + (-5)^2 + 24^2 + 8.5^2 + (-4.2)^2} = 170.3658;$$

因此可得 $d(\mathbf{s}_1, \mathbf{s}_2) < d(\mathbf{s}_1, \mathbf{s}_4)$, 所以 \mathbf{s}_1 与 \mathbf{s}_2 之间的相似程度更高。

通过以上的知识和实例可知, 对于任意两个样本, 若他们属性的取值在连续值区间内, 即可使用 (1) 式- (6) 式来度量它们之间的相似程度。然而, 在实际应用中, 我们并不能保证所有的属性取值都在连续值区间内, 在很多应用场景中, 有时候样本的属性取值可能是有限个, 即属性的取值在离散值区间内, 我们把这种属性称作为分类型属性, 若一个数据集所有的属性都是分类型, 则称这个数据集为分类型数据。对于分类

型数据，使用（1）式-（6）式并不能很好地度量它们的属性取值。为了便于说明这种情况，引入如下的实例。在表2中，为几位患者的症状表，其中： $\{s_1, s_2, s_3, s_4\}$ 为4名患者，而{体温，咳嗽，头痛，肌无力，鼻塞}为相应的症状。

表 2: 患者的症状表

人物/症状	体温	咳嗽	头痛	肌无力	鼻塞
s_1	高	是	是	否	是
s_2	正常	否	否	否	是
s_3	高	是	是	是	是
s_4	低	否	是	是	否

表 3: 患者的症状表

人物/症状	体温	咳嗽	头痛	肌无力	鼻塞
s_1	2	1	1	0	1
s_2	1	0	0	0	1
s_3	2	1	1	1	1
s_4	0	0	1	1	0

在表2中，属性值采用文字表述并不易于分析，为了便于描述，我们可以将体温的低、正常、高分别用数字0、1、2来代替，其他的属性同理用0与1来代替相应的属性值否和是，因此表2可以转换为表3。很显然，在表3中，属性值不同只是代表症状的不同，并没有大与小的关系。如果采用上述的欧几里得距离等方式来度量，这些方法会把属性值看着具有大小关系，这并不符合实际情况。为了度量出分类型数据的相似性，一般采用匹配距离(Matching Distance)。对于任意两个样本 $\mathbf{x} = [x_1, x_2, \dots, x_m] \in \mathbb{R}^{1 \times m}$ 与 $\mathbf{y} = [y_1, y_2, \dots, y_m] \in \mathbb{R}^{1 \times m}$ ，则 \mathbf{x} 与 \mathbf{y} 匹配距离的表达形式为：

$$d(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^m \varphi(x_i, y_i) \quad \text{且} \quad \varphi(x_i, y_i) = \begin{cases} 0 & \text{若 } x_i = y_i \\ 1 & \text{若 } x_i \neq y_i \end{cases} \quad (7)$$

在表3中，若使用匹配距离来计算 s_1 与 s_4 之间的聚类，则 $d_1(s_1, s_4) = \sqrt{1+1+1+1} = 2$ ；若采用欧式距离来计算，则 $d_2(s_1, s_4) = \sqrt{4+1+1+1} = 2.6457$ 。相对而言，采用匹配距离计算出来的结果会更符合实际情况。

除了以上的匹配距离，当前对分类型数据已经发展出很多其他的相似性度量方法，其中一种受到广泛关注和研究的方法即为不确定性度量 [10]，最常见的不确定性度量方法有粗糙集的方法 [11, 12]、粒计算的方法 [13]和形式概念分析的方法 [14]等，这些方法在处理分类型数据时，具有较为明显的优势，而且已经发展出较为成熟的理论。关于这类方法，由于涉及到较多近世代数方面的内容，对于这方面内容，感兴趣的同学可以查阅相关文献进行了解。

一般而言，分类型数据不具有几何结构，分类型属性的距离采用匹配距离计算，连续型属性可采用各种几何距离来计算；若数据同时包含分类型属性和连续型属性，一种最简单的思路即可把两种属性分别采用各自对应的方式来计算距离，最终的距离可设为两种属性距离之和。考虑到属性的重要性，例如对感冒诊断而言，体温对诊断结果的重要性一般要高于性别，因此也可以采用属性加权的方式来计算距离，即对每一个属性赋予一个权值，权值的大小代表了属性的重要程度。对于 \mathbf{x} , \mathbf{y} 加权的欧几里得距离和加权的匹配距离，其表达形式如下所示：

$$d_e(\mathbf{x}, \mathbf{y}) = \left(\sum_{i=1}^m w_i (x_i - y_i)^2 \right)^{\frac{1}{2}} \quad d_c(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^m w_i \varphi(x_i, y_i) \quad (8)$$

其中： $\mathbf{W} = [w_1, w_2, \dots, w_m]$ 为各个属性的权值，且有 $\sum_{i=1}^m w_i = 1, w_i \geq 0$ 。

在聚类任务，数据的类簇划分是依据对象之间的相似性来进行的，如果相似性的度量方法不合理，会极大影响聚类算法的性能，因此，选择出一个合适的相似性度量方法，对聚类算法具有非常重要的意义。

0.2.2 k-means算法的基本步骤

k-means算法是一种基于划分的基本算法，它是通过距离将数据划分到 k 个类簇中去，也就是每个样本划分到与其最近的类簇中去，并且使得同一类簇内样本之间的距离尽可能小，不同类簇样本之间的距离尽可能大。

对于一组 n 个样本 $\mathbf{X} = [\mathbf{x}_1^T, \mathbf{x}_2^T, \dots, \mathbf{x}_n^T]^T \in \mathbb{R}^{n \times m}$ 的数据，假设其聚类的类簇数目为 k 并采用欧几里得距离作为距离度量，根据聚类的目标，其目标函数可以表达为如下的形式：

$$F(\mathbf{U}, \mathbf{V}) = \sum_{i=1}^n \sum_{j=1}^m \sum_{r=1}^k u_{ir} (x_{ij} - v_{rj})^2 \quad \text{并且有 } u_{ir} \in \{0, 1\} \quad \sum_{r=1}^k u_{ir} = 1 \quad (9)$$

其中： $\mathbf{v}_r = [v_{r1}, v_{r2}, \dots, v_{rm}] \in \mathbb{R}^{1 \times m}$ 为第 r 个类簇的中心点， $\mathbf{U} \in \mathbb{R}^{n \times k}$, $\forall u_{ir} \in \mathbf{U}$ 表示第 i 个样本是否属于第 r 个类簇，若第 i 个样本是属于第 r 个类簇，则 $u_{ir} = 1$ ；否则， $u_{ir} = 0$ 。对(8)式， $F(\mathbf{U}, \mathbf{V})$ 代表的是各个样本点到所在类簇中心点的聚类，我们的目标是希望其值越小越好，因为其值越小，意味着类簇的内部越紧凑。

对于每一个样本，我们是将其划分到最近的类簇中去，因此，计算出当前样本到各个类中心的距离，把最近的类簇作为样本的归属，这种划分方法可以表达为如下的数学形式：

$$r = \operatorname{argmin}_r \sum_{j=1}^m (x_{ij} - v_{rj})^2 \quad (r = 1, 2, \dots, k) \quad (10)$$

在(10)式中，通过计算距离，确定了 \mathbf{x}_i 所在的类簇，因此 $u_{ir} = 1$ ，并且 $u_{ir_0} = 0$ ($r_0 = 1, 2, \dots, k; r_0 \neq r$)。

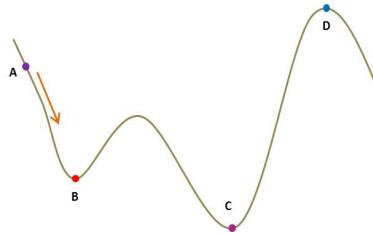


图 2: 贪婪算法的局部最优解情况（寻找函数最小值的解）

在确定 \mathbf{U} 后，即每个样本所归属的类簇也就确定了，由于类簇的中心点为类簇所有样本的均值，因此，每个类簇的中心点的更新方式如下：

$$\mathbf{v}_r = \frac{\sum_{i=1}^n u_{ir} \mathbf{x}_i}{\sum_{i=1}^n u_{ir}} \quad (r = 1, 2, \dots, k) \quad (11)$$

更新完类簇的中心点后，获得一组新的类簇中心点，利用这组新的类簇中心点，又可以重新来对类簇的数据进行划分，从而再次更新类簇中心点，重复以上的过程，直到相邻两次更新的中心点变化很小，即算法达到收敛状态，此时所获得的聚类结果为最终的聚类结果。对于k-means 算法的具体流程可以总结为如算法1所示 [3]。

算法 1 k-means算法.

输入: 一个需要聚类的数据 $\mathcal{X} \in \mathbb{R}^{n \times m}$; 聚类的类簇数目 k 。

输出: 数据的划分矩阵 \mathbf{U} 和类簇中心点 \mathbf{V} .

随机地选择 k 个中心点，获得中心点矩阵 $\mathbf{V} \in \mathbb{R}^{k \times m}$;

确定每个数据所属的类簇，获得划分矩阵 $\mathbf{U} \in \mathbb{R}^{n \times k}$;

while 当算法不收敛 **do**

 依据 (11) 式更新每个类簇的中心点;

 依据 (10) 式确定每个数据的类簇;

end while

算法1描述了k-means算法的基本步骤，从基本步骤可以看出，k-means算法属于一种贪婪算法，即算法迭代沿着使目标函数值减小的方向寻找解，一旦目标函数达到收敛状态，算法终止迭代。然而，这种采用贪婪的策略，也会带来一些问题，会使得算法的解处于局部最优解，所获的解可能并不一定是全局最优。如图2显示了贪婪算法的局部最优问题，在图2中，函数的全局最小解为点C处的解，如果算法从A点开始贪婪搜索，将会在B点处终止，即算法会把B点作为最小值的解，但是，很明显C点的解比B更优。所以，如何避免k-means算法陷入局部最优解，这也是一个值得研究的问题。

为了展示如何使用k-means算法，我们给出一个实例来具体讲解k-means算法的聚类过程。在如图3中为5个点，假设k-means算法聚类的类簇数目为2，算法的距离度量采用欧几里得距离，聚类中心点的变化终止条件为0.00001，算法的具体聚类过程如下所示：

(1)选择初始中心点，对于初始中心点可以随机选择，在本实例中，为了便于计算，我们

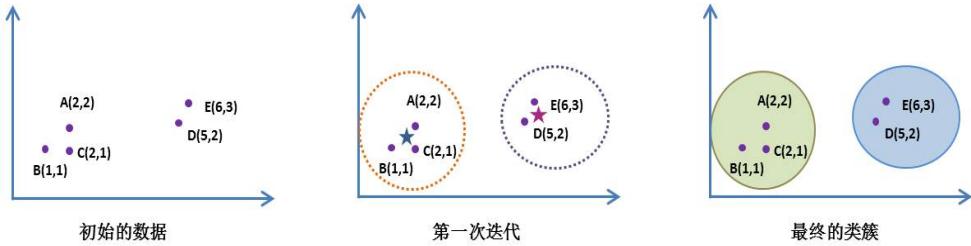


图 3: k-means 算法聚类的实例

把B(1,1)与D(5,2)作为聚类的初始中心点；

(2)计算各点到B与D点的距离，并把其分配到相应的类簇中去；对于A点有， $d(A, B) = \sqrt{1+1} = 1.414$, $d(A, D) = \sqrt{9} = 3$, 所以A点应归属到B点所在类簇；对于C点， $d(B, C) = \sqrt{1+0} = 1$, $d(C, D) = \sqrt{9+1} = 3.162$, 所以C点应归属到B点所在类簇；对于E点， $d(E, B) = \sqrt{25+4} = 5.385$, $d(E, D) = \sqrt{1+1} = 1.414$, 所以E点应归属到D点所在类簇；本次划分形成的类簇为 $\{\{A, B, C\}, \{D, E\}\}$ ；

(3)更新类簇的中心点， $v_1 = \frac{A+B+C}{3} = (1.667, 1.333)$, $v_2 = \frac{D+E}{2} = (5.5, 2.5)$ ；由于 $d(v_1, B) + d(v_2, D) = \sqrt{(1.667 - 1)^2 + (1.333 - 1)^2} + \sqrt{(5.5 - 5)^2 + (2.5 - 2)^2} = 1.453 > 0.00001$ ，所以算法不终止，进行下一次迭代；

(4)计算A点到 v_1 和 v_2 的距离， $d(A, v_1) = 0.746$, $d(A, v_2) = 3.536$, 所以A点归属于 v_1 所在的类簇；对于B点， $d(B, v_1) = 0.746$, $d(B, v_2) = 4.743$, 所以B点归属于 v_1 所在的类簇；对于C点， $d(C, v_1) = 0.471$, $d(C, v_2) = 3.808$, 所以C点归属于 v_1 所在的类簇；对于D点， $d(D, v_1) = 3.399$, $d(D, v_2) = 0.707$, 所以D点归属于 v_2 所在的类簇；对于E点， $d(E, v_1) = 4.643$, $d(E, v_2) = 0.707$, 所以E点归属于 v_2 所在的类簇；本次划分形成的类簇为 $\{\{A, B, C\}, \{D, E\}\}$ ；

(5)更新类簇的中心点， $v_1 = \frac{A+B+C}{3} = (1.667, 1.333)$, $v_2 = \frac{D+E}{2} = (5.5, 2.5)$ ；由于 v_1 和 v_2 两个中心点未发生变化，即中心点移动的距离为 $0 < 0.00001$ ，此时，算法满足终止条件，算法终止；

(6)输出最终的聚类结果 $\{\{A, B, C\}, \{D, E\}\}$ 。

0.2.3 k-means 算法的特点

k-means 算法作为一种典型的聚类算法，曾被评为数据挖掘十大算法之一 [15]，足以可见其在数据挖掘领域的影响力，但是作为一种常见和应用较为广泛的聚类算法，k-means 算法由于自身的特点，具有一些优点和缺点。对于 k-means 算法，主要具有以下的优点：

(1) 算法的原理较为简单，能够很容易被理解，并且易于实现；在 k-means 算法中，主要就是两步：更新中心点和更新数据的划分；随着迭代的进行，算法所搜寻的中心点会逐步向真实的中心点靠近，最终的达到收敛状态。由此可见，算法部分的步骤可以重复

调用，实现起来并不需要编写太多的代码；

(2)该算法时间复杂度为 $\mathcal{O}(tkmn)$ ，其中， t 为迭代次数， k 为簇的数目， n 为数据的数目， m 为数据的维数，算法的时间复杂度与样本数量线性相关，所以，对于处理大数据集合，该算法同样也非常高效，这表明k-means算法具有良好的伸缩性。

同样，k-means算法的缺点也比较明显，在余下的内容里，将重点讲解k-means算法的一些缺点，并对这些缺点进行分析。k-means算法的缺点主要有如下几点：

(1)算法的 k 值不容易确定，k-means算法的 k 值选择对先验知识比较依赖，不同的 k 值也将会产生不同的结果，当前还没有一种完美的方法能够精确地确定 k 值。对 k 值的确定，绝大多数方法还是依赖于先验知识。图4展示了 k 值对k-means算法的影响，当 k 值从2取到4时，对于不同的 k 值，数据的聚类结果也明显不同。因此，如何选择合适的 k 值，这是一个值得研究的问题。

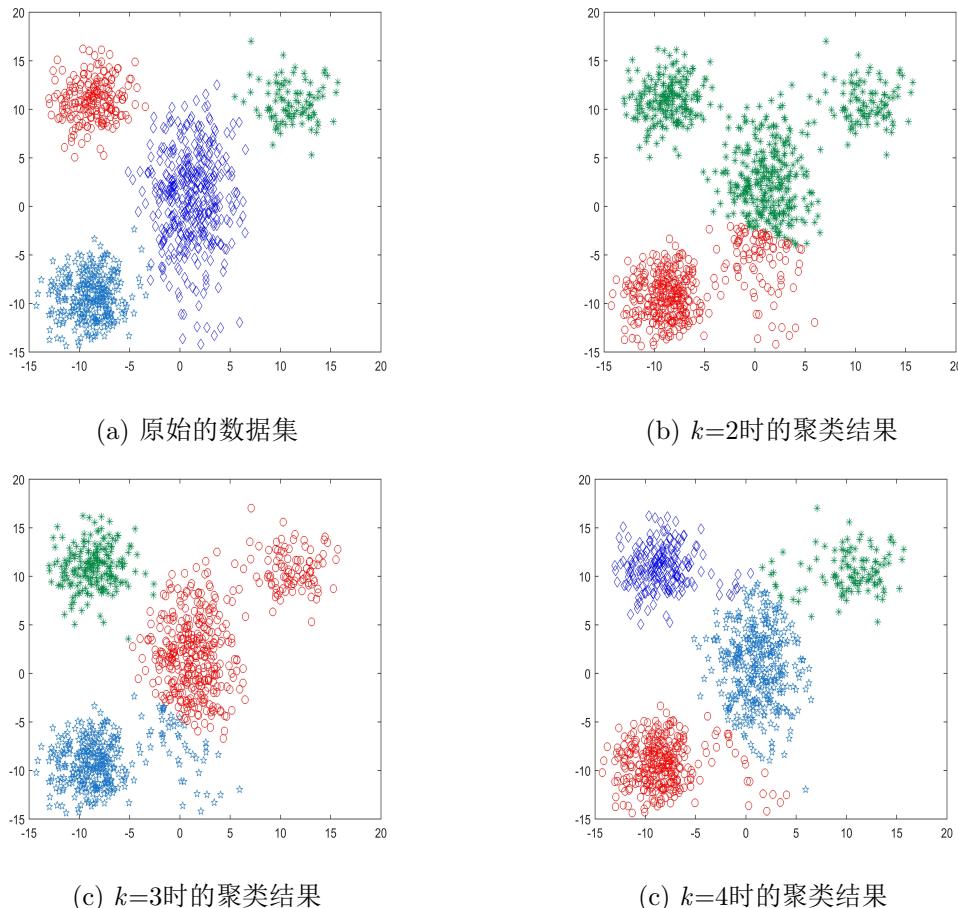


图 4: 不同 k 值对k-means算法聚类结果的影响

(2)k-means算法对球状（凸状）数据能够取得较好的结果，但是对于非球状数据并不能取得令人满意的聚类结果。图5展示了非球状数据对k-means算法的影响，我们分别

运行了k-means算法和谱聚类算法[16], 结果如图5所示, 从运行的结果来看, k-means算法几乎把数据在空间内进行三等分, 相比较而言, 谱聚类所获得结果与原始数据的分布更为接近。这表明, k-means算法并不能很好地应对非球状数据的聚类任务。

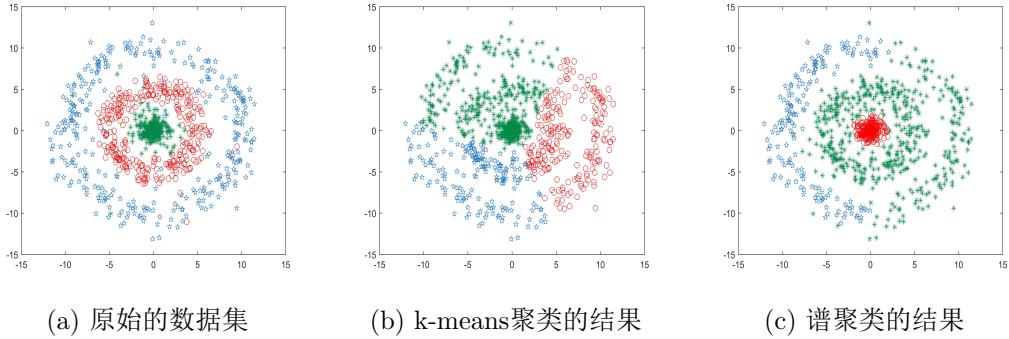


图 5: 数据的形状对k-means算法聚类结果的影响

(3)由于初始中心点是随机选择的, 初始中心点大的随机性, 可能会导致聚类结果也具有一定的随机性。图6展示了随机性对k-means算法聚类结果的影响, 在实验中, 我们将k-means算法运行在图5的数据集上3次, 可以看出, 3次的结果都有一些差异, 这也表明k-means算法容易受到随机性的影响, 因此, 在评价聚类算法的时候, 为了更全面地反映出聚类算法的性能, 我们一般需要多次运行算法, 最终的结果为多次实验结果的均值。

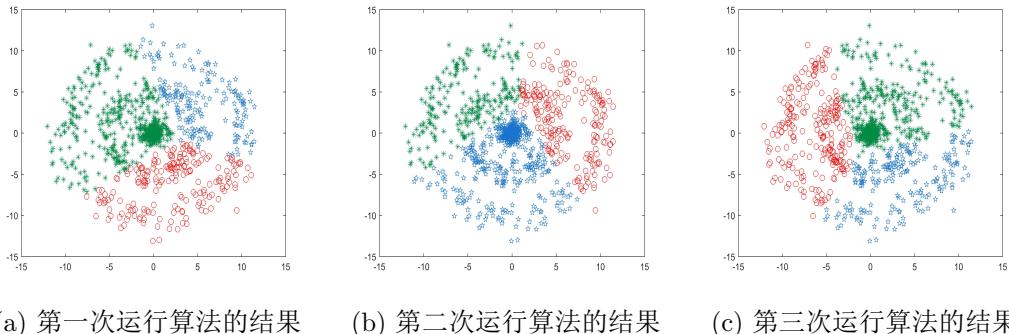


图 6: 随机性对k-means算法聚类结果的影响

(4)由于k-means算法需要所有点参与到聚类的过程中去, 这使得算法较易受到噪声的影响。图7展示了噪声对k-means算法聚类结果的影响, 在实验中, 我们选择图4的数据集作为实验数据, 这是因为通过实验发现, 图4数据集实验结果的随机性水平较低, 用此数据集进行实验, 能够减小初始中心点对聚类结果的影响。我们将k-means算法运行在不同的噪声水平下, 实验结果如图7所示。从图7的实验结果可以看出, 当噪声水平增加时, 数据被划分错误划分的概率也开始增加, 低水平噪声下的聚类结果明显要好于高水平噪声下的聚类结果。因此, 如何提高k-means算法的鲁棒性, 这也是一个受到广泛关注的问题。

题。

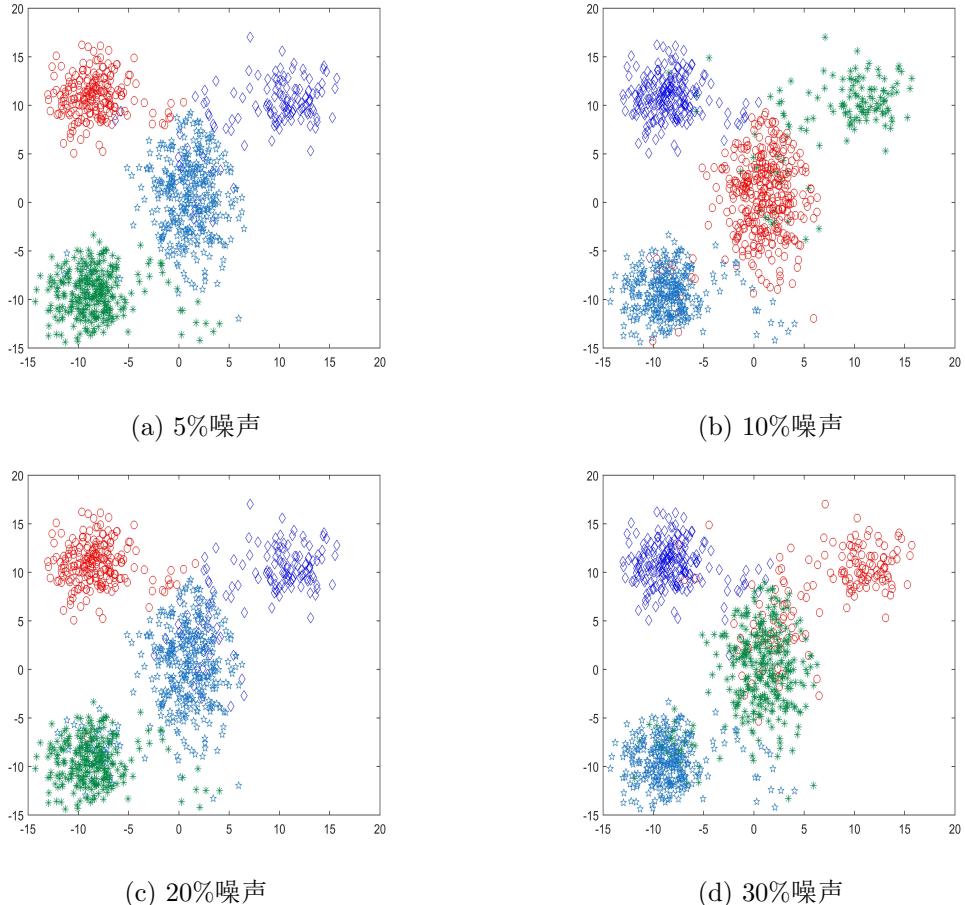


图 7: 不同噪声水平下对k-means算法聚类结果的影响

(5)在前一小节介绍k-means算法基本步骤中，我们分析了k-means算法是一个贪婪算法，因此，k-means算法与像其他的贪婪算法那样，可能会陷入局部最优解中，这也是k-means算法的一个不足之处。

由上述内容可知，k-means算法尽管简单有效，并且在有些情况下能够取得较好的效果，但是，由于其自身的一些特点，使得它也有不少的不足之处，在实际应用中，我们需要根据具体的任务，结合k-means算法特点来设计出具体的算法来解决具体问题。

0.3 基于k-means算法的扩展算法

由上一节的内容可知，k-means算法具有一些固有的缺点，因此，不少的学者针对这些不足之处，来对k-means算法进行改进，从而扩展k-means算法的应用范围 [17, 18]。从已有的算法来看，针对k-means算法的改进基本是从已有的缺点来入手或者根据特定应用

场景来提出新的算法。下面将从以下几个方面来介绍已有的改进型算法。

0.3.1 能够自主决定类簇数目的k-means算法

对于k-means算法而言，需要用户预先输入类簇的数目，在无监督学习任务中，类簇的数目往往无法预先知道，因此，许多学者提出了一系列能够自主决定类簇数目的k-means算法 [19]。在这些算法中，最具代表性的即是X-means算法 [20]和G-means算法 [21]。

X-means算法采用一种正则化的框架来学习 k 值，该算法通过设置一些列不同的候选 k 值，并使用一种贝叶斯信息准则(Bayesian Information Criterion,BIC)来评价这些聚类模型，对于一个聚类模型C而言，其BIC的定义如下：

$$BIC(C|\mathbf{X}) = \mathcal{L}(\mathbf{X}|C) - \frac{p}{2} \log n \quad (12)$$

其中： $\mathbf{X} \in \mathbb{R}^{n \times m}$ 为无标签的数据， n 为数据的数目， m 为数据的维数， $p = k(m+1)$ ， $\mathcal{L}(\mathbf{X}|C)$ 为在聚类模型C下数据集 \mathbf{X} 的log似然概率。利用BIC即可对每个不同 k 值的聚类模型进行打分，得分最高模型所对应的 k 值即为最终的结果。

G-means算法是一种基于假设检验来选择合适的 k 值，该算法认为数据在类簇空间内应服从的高斯分布，并有如下的两个假设： H_0 ：围绕在中心点附近的数据服从高斯分布； H_1 ：围绕在中心点附近的数据不服从高斯分布。在算法开始阶段，算法认为只有一个类簇即 $k=1$ ，并把整个数据集的均值作为初始的中心点，对于每一个中心点，尝试分裂该中心点，把 k 设置为2，运行k-means算法获得聚类结果并获得两个类簇中心点 \mathbf{c}_1 与 \mathbf{c}_2 ，然后由 \mathbf{c}_1 与 \mathbf{c}_2 可获得一个方向向量 $\mathbf{c} = \mathbf{c}_1 - \mathbf{c}_2$ ，并把此类簇的任意一个数据 \mathbf{x}_i 投影到此方向向量上，即 $x'_i = \frac{\langle \mathbf{x}_i, \mathbf{c} \rangle}{\|\mathbf{c}\|^2}$ ，令 $F(x'_i)$ ， $F(\cdot)$ 为一个服从标准正态分布的累积分布函数，并令 $A^2(z_i)$ 为：

$$A^2(z_i) = -\frac{1}{n} \sum_{i=1}^n (2i-1) [\log(z_i) + \log(1-z_{n+1-i})] - n \quad (13)$$

由 $A^2(z_i)$ 可得 $A_*^2(z_i)$ 为：

$$A_*^2(z_i) = A^2(z_i) \left(1 + \frac{4}{n} - \frac{25}{n^2}\right) \quad (14)$$

如果 $A_*^2(z_i)$ 在置信水平为 α 的非临界值范围内，则接受假设 H_0 ，并维持原中心点，并把新产生的中心点 \mathbf{c}_1 ， \mathbf{c}_2 丢弃，否则，则接受假设 H_1 ，用新产生的中心点 \mathbf{c}_1 与 \mathbf{c}_2 来替换原有的中心点。重复以上步骤，直至没有新的中心点加入为止。

0.3.2 鲁棒性的k-means算法

原始的k-means算法一般会较易受到噪声数据的影响，如何提高k-means算法的鲁棒性，这也是一个重要的研究问题。在当前，针对提高k-means算法鲁棒性的算法中，主要

分为两种方法：孤立点剔除法和子空间学习方法。孤立点剔除法即事先检测出数据中的离群点或噪声点，然后再对此数据进行聚类。子空间学习方法是通过学习一个新的特征空间来表示原有的数据 [22, 23]，常见的子空间学习方法有子空间分解法、稀疏学习和低秩学习等。

在孤立点剔除法中，最具代表性的是k-means--算法 [24]，该算法通过使用距离度量来检测数据中离群点。对于任意一点 \mathbf{x} 和类簇 \mathbf{C} 而言， \mathbf{x} 到最近的类簇中心点之间的距离 $d(\mathbf{x}|\mathbf{C})$ 为：

$$d(\mathbf{x}|\mathbf{C}) = \min_{\mathbf{c} \in \mathbf{C}} \{d(\mathbf{x}, \mathbf{c})\} = d(\mathbf{x}, c(\mathbf{x}|\mathbf{C})) \quad (15)$$

其中： $c(\mathbf{x}|\mathbf{C}) = \underset{\mathbf{c} \in \mathbf{C}}{\operatorname{argmin}} \{d(\mathbf{x}, \mathbf{c})\}$ 。对于k-means--算法，首先选择出 k 个初始中心点，计算出每一个数据 \mathbf{x} 到各自类簇中心之间的距离 $d(\mathbf{x}|\mathbf{C}_{i-1})$ ，然后，将各个点到类簇中心的距离进行降序排列，并把前 ℓ 个点 $\mathbf{L}_i \leftarrow \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_\ell\}$ 作为噪声点，最后利用非噪声数据 $\mathbf{X}_i \leftarrow \mathbf{X} - \mathbf{L}_i$ 来更新类簇中心点，重复以上的步骤，直至算法到达收敛状态为止。在k-means--算法，作者将此算法扩展到使用布雷格曼散度(Bregman Divergences)作为相异性度量的情况下。任意两个数据 $\mathbf{x}, \mathbf{y} \in \mathbf{X}$ ，给定一个严格的凸函数 Φ ， \mathbf{x} 与 \mathbf{y} 的布雷格曼散度 $d_\Phi(\mathbf{x}, \mathbf{y})$ 定义如下：

$$d_\Phi(\mathbf{x}, \mathbf{y}) = \Phi(\mathbf{x}) - \Phi(\mathbf{y}) - \langle \mathbf{x} - \mathbf{y}, \nabla \Phi(\mathbf{y}) \rangle \quad (16)$$

其中： $\nabla \Phi(\mathbf{y})$ 为 \mathbf{y} 上计算 Φ 的梯度。布雷格曼散度度量了两者变量之间差异的大小，但它只是作为类似距离度量的方式，并不满足对称性和传递性，因此，布雷格曼散度不符合距离的定义。在k-means--算法中，作者指出了此算法同样收敛于局部最优解。

子空间学习作为当前一个非常火热的研究方向，已经发展出非常成熟的理论与算法 [25]。子空间分解的方法是把原始的数据投影到不同的子空间来区分噪声和非噪声数据。SUBKMEANS算法就是一种子空间学习算法 [26]，该算法把数据的空间分为类簇空间和噪声空间，设 \mathbf{V} 使得原始特征中的前 m 个特征位于类簇空间且后 $d-m$ 个特征位于噪声空间的变换矩阵（ d 为数据的维数），因此，两个对应的投影矩阵 \mathbf{P}_C 和 \mathbf{P}_N 为：

$$\mathbf{P}_C = \begin{bmatrix} \mathbf{I}_m \\ \mathbf{0}_{d-m, m} \end{bmatrix} \quad \mathbf{P}_N = \begin{bmatrix} \mathbf{0}_{d-m, m} \\ \mathbf{I}_m \end{bmatrix} \quad (17)$$

其中： \mathbf{I}_m 为单位矩阵。对于任意一个 \mathbf{x} ，它在类簇空间的投影为 $\mathbf{P}_C \mathbf{V}^T \mathbf{x}$ ，它在噪声空间的投影为 $\mathbf{P}_N \mathbf{V}^T \mathbf{x}$ 。因此，SUBKMEANS算法的目标函数为：

$$J = \sum_{i=1}^k \sum_{\mathbf{x} \in \mathbf{C}_i} \|\mathbf{P}_C^T \mathbf{V}^T \mathbf{x} - \mathbf{P}_C^T \mathbf{V}^T \boldsymbol{\mu}_i\|_F^2 + \sum_{\mathbf{x} \in \mathbf{D}} \|\mathbf{P}_N^T \mathbf{V}^T \mathbf{x} - \mathbf{P}_N^T \mathbf{V}^T \boldsymbol{\mu}_D\|_F^2 \quad (18)$$

其中： $\|\cdot\|_F$ 为Frobenius范数， \mathbf{C}_i 为第*i*个类簇， \mathbf{D} 为原始的所有数据对象， $\boldsymbol{\mu}_i$ 与 $\boldsymbol{\mu}_D$ 分别

为 C_i 与 D 中数据的均值向量。通过一系列变换，作者将（18）式变换为如下式：

$$J = \text{Tr} \left(\mathbf{P}_C \mathbf{P}_C^T \mathbf{V}^T \left(\sum_{i=1}^k \mathbf{S}_i - \mathbf{S}_D \right) \mathbf{V} \right) + \text{Tr} (\mathbf{V}^T \mathbf{S}_D \mathbf{V}) \quad (19)$$

令 $\Sigma = \sum_{i=1}^k \mathbf{S}_i - \mathbf{S}_D$ ，作者指出了 Σ 前 m 个最小的特征值所对应的特征向量即组成类簇空间，剩余的 $(d-m)$ 个最小的特征值所对应的特征向量即组成噪声空间。

稀疏学习和低秩学习也是一种重要的机器学习方法 [27]，近几年来，不少了研究把这两种方法同k-means聚类相结合。其中Cohen等人提出了一种约束的低秩近似(Constrained Low Rank Approximation)的k-means算法 [28]。对于一个矩阵 $\mathbf{A} \in \mathbb{R}^{n \times d}$ ， $\mathbf{P} \in \mathbb{R}^{n \times n}$ 且 $\text{rank}(\mathbf{P}) = k < n$ ，其约束的低秩近似可以表达为如下的问题：

$$\mathbf{P}^* = \underset{\mathbf{P}}{\operatorname{argmin}} \|\mathbf{A} - \mathbf{PA}\|_F^2 \quad s.t. \text{rank}(\mathbf{P}) = k \quad (20)$$

在k-means聚类算法中，设 $\mathbf{F}_C \in \mathbb{R}^{n \times k}$ 为标识矩阵，若 \mathbf{X}_i 归属于类簇 C_i 中，则 $F_C(i, j) = \frac{1}{|C_i|}$ ， $|C_i|$ 为类簇 C_i 中数据的数目；否则， $F_{ij} = 0$ 。因此， $\mathbf{F}_C \mathbf{F}_C^T \mathbf{X}$ 为各个数据所归属类簇类簇的中心点，即第 i 行为第 i 个样本所对应类簇的中心点。所以，k-means算法可以写为如下的表达形式：

$$\min_{\mathbf{F}_C} \|\mathbf{X} - \mathbf{F}_C \mathbf{F}_C^T \mathbf{X}\|_F^2 \quad (21)$$

由于 $\mathbf{F}_C \mathbf{F}_C^T$ 是一个正交矩阵，且矩阵的秩为 k ，所以此k-means算法属于约束的低秩近似问题，（21）式可以转化为（20）式来进行求解。

文献 [29]所提出的鲁棒性的基于低秩表示的数据降维方法(Low Rank Representation,LRR)也具有代表性，尽管LRR方法并不是针对聚类问题，但是该算法具有很好的鲁棒性，其降维后的数据可以通过运行k-means算法来进行聚类。对于LRR方法，其可以表示为如下的优化问题：

$$\min_{\mathbf{Z}, \mathbf{E}} \|\mathbf{Z}\|_* + \lambda \|\mathbf{E}\|_{2,1} \quad s.t. \quad \mathbf{X} = \mathbf{XZ} + \mathbf{E} \quad (22)$$

其中： $\|\mathbf{Z}\|_*$ 为矩阵的核范数，其值为矩阵 \mathbf{Z} 的奇异值之和； $\|\mathbf{E}\|_{2,1} = \sum_{j=1}^n \sqrt{\sum_{i=1}^n E_{ij}^2}$ 为矩阵的 $\ell_{2,1}$ 范数。对于 $\mathbf{X} = \mathbf{XZ} + \mathbf{E}$ ，这相当于从给出的数据中把原始数据和噪声数据自动分割出来。关于（22）式，文献 [29]给出了具体的求解方法。

0.3.3 自主选择初始中心点的k-means算法

在原始的k-means算法中，初始中心点随机选择会导致算法的性能具有一定的随机性，如果能选择出质量较高的初始中心点，不但能加快算法的收敛速度，而且还能提高算法的性能。许多学者针对这一问题，提出了许多k-means改进算法。在这里，将主要介绍四种算法k-means++ [30]、Efficient k-means++ [31]、K-MC² [30]和AFK-MC² [30]。

k-means++是一种非常基础的方法，当前许多针对选择初始中心点的改进型算法都能看到此算法的思想。k-means++初始中心点选取的基本思想就是，初始的聚类中心点之间要尽可能的远离。因此，k-means++算法的基本思路如下：首先，从数据集 \mathbf{X} 中随机选择一个样本作为第一个中心点，并将其加入到类簇中心点集 C 中，然后，对于 $\mathbf{x} \in \mathbf{X}$ ，依据 $D^2(\mathbf{x}, C) / \sum_{\mathbf{x} \in \mathbf{X}} D^2(\mathbf{x}, C)$ 的概率来选择 \mathbf{x} 作为中心点，其中： $D(\mathbf{x}, C)$ 为 \mathbf{x} 距 C 中点最近的距离；最后，重复以上步骤，直至选择出 k 个中心点。k-means++聚类的过程同k-means基本类似，只不过其在聚类前加入了初始中心点选择的过程，由于高质量的聚类中心点能减少迭代次数和获得更好的聚类结果，因此k-means++具有较快的收敛速度和较好的性能。

Efficient k-means++是一种基于随机投影 [32]的聚类框架，在此框架下，生成一个随机投影矩阵 $\mathbf{R} \in \mathbb{R}^{d \times m}$ 且有 $\mathbf{R} \leftarrow N(0, 1) \times \frac{1}{\sqrt{d}}$ ，原始的数据 \mathbf{X} 被投影到一个低维的空间中 $\mathbf{X}_{proj} = \mathbf{X}\mathbf{R}$ 。若 \mathbf{X}_{proj} 生成于初始中心点选择步骤之前，则形成了kmFRP算法；若在每一次选择初始中心点时，都重新生成 \mathbf{X}_{proj} ，则形成了kmIRP算法；若形成若干个 \mathbf{X}_{proj} ，放入缓冲区中，每次选择初始中心点时，都从缓冲区中选择一个 \mathbf{X}_{proj} ，由此形成了kmBRP算法。

在k-means++算法中，初始中心点的选择需要预先获取所有的数据，这限制了k-means++算法在海量数据聚类中的应用。K-MC²算法引入了基于马尔科夫链(Markov Chain)的 D^2 采样方法，从而较好地解决了算法的伸缩性问题 [33]。设马尔科夫链的长度为 m ，KM-C²算法选择一个初始的类簇中心点 \mathbf{c}_1 ， $C_1 \leftarrow \{\mathbf{c}_1\}$ ，再从 \mathbf{X} 中按照均匀分布的概率抽取一个样本 \mathbf{x} ，计算出 $d_{\mathbf{x}} \leftarrow d^2(\mathbf{x}, C_{i-1}) = \min_{\mathbf{c} \in C} \|\mathbf{x} - \mathbf{c}\|_2^2$ ，然后，构建一条长为 m 的马尔科夫链进行取样，对每一个节点，从 \mathbf{X} 中按照均匀分布的概率抽取一个样本 \mathbf{y} ，计算出 $d_{\mathbf{y}}$ ，若 $d_{\mathbf{x}}/d_{\mathbf{y}} > Unif(0, 1)$ ，则 $\mathbf{x} \leftarrow \mathbf{y}$ ， $d_{\mathbf{x}} \leftarrow d_{\mathbf{y}}$ ，并把这条链的输出点 \mathbf{x} 作为一个选择的类簇中心点， $C_i \leftarrow C_{i-1} \cup \{\mathbf{x}\}$ ，按照以上方法，再次取样，直至选择出 k 个初始的聚类中心点。

在K-MC²算法中，假设数据的分布服从均匀分布，并且按照均匀分布来对数据进行取样，然而，有时候数据的分布可能未知或并不一定服从均匀分布，在文献 [34]中，提出了一种不带任何先验假设的KM-C²算法(Assumption-Free K-MC², AFK-MC²)，AFK-MC²算法的基本思路同KM-C²算法相类似，只不过采取了马尔科夫链蒙特卡洛(Markov Chain Monte Carlo)方法来对数据进行取样，对于任意 $\mathbf{x} \in \mathbf{X}$ ，其取样的概率 $q(\mathbf{x})$ 为：

$$q(\mathbf{x}) \leftarrow \frac{1}{2} \frac{d^2(\mathbf{x}, C)}{\sum_{\mathbf{x} \in \mathbf{X}} d^2(\mathbf{x}, C)} + \frac{1}{2n} \quad (23)$$

依照K-MC²算法相类似的思路，即可选择出 k 个初始聚类中心点。

0.3.4 基于平滑性距离的k-means算法

在第二节中，我们介绍了一些度量距离方法，在这些距离度量中，欧几里得距离是最常见的一种距离，然而，每种距离都不同的特点，所以不同的距离度量对聚类算法的性能也会产生一些影响。在本小节中，我们将介绍几种基于平滑距离的k-means算法。

对于数据 $\mathbf{X} = [\mathbf{x}_1^T, \mathbf{x}_2^T, \dots, \mathbf{x}_n^T]^T \in \mathbb{R}^{n \times d}$, 聚类的中心点 $\mathbf{V} = [\mathbf{v}_1^T, \mathbf{v}_2^T, \dots, \mathbf{v}_n^T]^T \in \mathbb{R}^{k \times d}$, 文献 [35] 提出了一种 k-harmonic means(KHM) 算法, 该算法使用了调和均值(harmonic means) 来代替欧几里得距离, 因此, k-harmonic means 算法的目标函数如下所示:

$$f(\mathbf{V}) = \sum_{i=1}^n \left(\frac{1}{k} \|\mathbf{x}_i - \mathbf{v}_j\|_2^{-2} \right)^{-1} \quad (24)$$

作者在文中指出了, 调和均值为 (24) 式提供了光滑代理(Smooth Proxy), 会产生更好的鲁棒性和更为精简的迭代过程。作者然后将 KHM 算法中的距离 $\|\mathbf{x}_i - \mathbf{v}_j\|_2^{-2}$ 扩展到更一般情况, 并提出了 KHM_p 算法[36], 其距离度量采用 $\|\mathbf{x}_i - \mathbf{v}_j\|_2^{-p}$, 因此, (24) 式的目标函数变为如下式所示:

$$f(\mathbf{V}) = \sum_{i=1}^n \left(\frac{1}{k} \|\mathbf{x}_i - \mathbf{v}_j\|_2^{-p} \right)^{-1} \quad (25)$$

作者在文中指出了, 如果认真调整 p 值, KHM_p 算法能够取得比 KHM 算法更好的聚类效果。

文献 [37] 提出了一种 power k-means 算法, 不同于传统的 k-means 算法, 该算法使用幂均值(Power Mean) 作为相异性度量。对于一对象 $\mathbf{y} = [y_1, y_2, \dots, y_k]$, 其科尔莫戈罗夫均值(Kolmogorov Mean) [38] 定义如下:

$$M_g(\mathbf{y}) = g^{-1} \left[\frac{g(y_1) + g(y_2) + \dots + g(y_k)}{k} \right] \quad (26)$$

当 $M_g(\mathbf{y}) = y^s$ 时, (26) 式即变为幂均值(Power Means)。基于幂均值 power k-means 算法的目标函数表示如下:

$$f(\mathbf{V}) = \sum_{i=1}^n M_s \left(\|\mathbf{x}_i - \mathbf{v}_1\|_2^2, \dots, \|\mathbf{x}_i - \mathbf{v}_k\|_2^2 \right) \quad (27)$$

power k-means 算法的参数更新过程同 k-means 算法相类似, 对于第 m 次迭代, 样本 \mathbf{x}_i 隶属于类簇第 j 个类簇的隶属度 $w_{m,ij}$ 和中心点 $\mathbf{v}_{m+1,j}$ 更新的方式如下:

$$w_{m,ij} \leftarrow \left(\sum_{l=1}^k \|\mathbf{x}_i - \mathbf{v}_{m,l}\|_2^{2s_m} \right)^{\frac{1}{s_m}-1} \|\mathbf{x}_i - \mathbf{v}_{m,j}\|_2^{2(s_m-1)} \quad (28)$$

$$\mathbf{v}_{m+1,j} \leftarrow \left(\sum_{i=1}^n w_{m,ij} \right)^{-1} \sum_{i=1}^n w_{m,ij} \mathbf{x}_i \quad (29)$$

在 (28) 式和 (29) 式中, $s_m \leftarrow \eta \cdot s_m$, η 为预先确定的参数。

除了以上所介绍的方法, 在文献 [39] 中, 还提出了一种利用无监督的信息论来建立二叉树的聚类方法, 该算法通过信息熵或信息增益来分裂树的叶子节点, 通过设置树的最大高度来控制无关属性对聚类结果的影响, 对于每对样本, 根据从根节点到各自叶节点公共路径的长度来计算相似性, 尽管该方法是针对谱聚类所提出的, 但是这种思路对 k-means 算法中度量两个样本之间的相似性或相异性也具有一定的启发作用。

0.3.5 基于模糊聚类的k-means算法

模糊聚类是一种重要的聚类方法，对于k-means算法，在(9)式中，划分矩阵的每一个元素 $u_{ir} \in \{0, 1\}$ ，这种聚类方法称着为硬聚类方法。在实际中，硬划分有时并不完全符合实际情况，例如，对于一个身高1.9m的人属于高个子，但对于1.75m的身高单纯用高或矮来划分并不合适，相对而言，用一个隶属度衡量更为合适。一般而言，如果 $u_{ir} \in [0, 1]$ 连续区间中的一个实数，则把(9)式所代表的聚类算法称为模糊C-Means算法(Fuzzy C-Means,FCM) [40]，也被称为软聚类算法。

文献[41]提出了一种面向高维数据的软子空间聚类算法，该算法是针对FCM算法的一种扩展，其目标函数采用加权距离，考虑到每个属性可能在不同的类簇空间中所起的作用也不同，因此，同一属性在不同的类簇空间中的权值也应该不同。对于该算法的目标函数如下式所示：

$$\begin{aligned} f(\mathbf{V}, \mathbf{W}, \mathbf{U}) = & \sum_{i=1}^n \sum_{j=1}^d \sum_{r=1}^k u_{ir}^m w_{rj} (x_{ij} - v_{rj})^2 + \gamma \sum_{r=1}^k \sum_{j=1}^d w_{rj} \log w_{rj} \\ & - \eta \sum_{r,t=1}^k \left(\sum_{i=1}^n u_{ir}^m \right) \sum_{j=1}^d (v_{rj} - v_{tj})^2 \quad (30) \\ s.t. \quad & 0 \leq u_{ir} \leq 1, \sum_{i=1}^n \sum_{r=1}^k u_{ir} = 1, 0 \leq w_{rj} \leq 1, \sum_{r=1}^k \sum_{j=1}^d w_{rj} = 1 \end{aligned}$$

在(30)式中，第一项为类簇的紧凑度，第二项为权值熵，加上这一项，是因为其具有一定的物理含义[42]，第三项为类簇中心点之间的距离， γ 和 η 为预先定义的参数。对于一个聚类结果而言，第一项和第二项的值越小越好，第三项的值越大越好。关于此优化问题的求解方法，一般是采用交替方向乘子法(Alternating Direction Method of Multipliers,ADMM) [43]。

在文献[41]中，作者提出了两种模糊聚类算法SCAD1和SCAD2(Simultaneous Clustering and Attribute Discrimination,SCAD)。这两种算法的基本流程相类似，只不过采用不同的损失函数。在SCAD1算法中，其定义的目标优化函数如下式所示：

$$\begin{aligned} f(\mathbf{V}, \mathbf{W}, \mathbf{U}) = & \sum_{i=1}^n \sum_{j=1}^d \sum_{r=1}^k u_{ir}^m w_{rj} d_{ijr}^2 + \sum_{r=1}^k \delta_r \sum_{i=j}^d w_{rj}^2 \quad (31) \\ s.t. \quad & 0 \leq u_{ir} \leq 1, \sum_{i=1}^n \sum_{r=1}^k u_{ir} = 1, 0 \leq w_{rj} \leq 1, \sum_{r=1}^k \sum_{j=1}^d w_{rj} = 1 \end{aligned}$$

其中： δ_r 相关的系数， m 为常数， $d_{ijr} = |x_{ij} - v_{rj}|$ 。在SCAD2算法中，其定义的目标优

化函数如下式所示：

$$\begin{aligned}
 f(\mathbf{V}, \mathbf{W}, \mathbf{U}) &= \sum_{i=1}^n \sum_{j=1}^d \sum_{r=1}^k u_{ir}^m w_{rj}^q d_{ijr}^2 \\
 \text{s.t. } 0 \leq u_{ir} &\leq 1, \sum_{i=1}^n \sum_{r=1}^k u_{ir} = 1, 0 \leq w_{rj} \leq 1, \sum_{r=1}^k \sum_{j=1}^d w_{rj} = 1
 \end{aligned} \tag{32}$$

关于（31）式与（32）式中的优化问题，也是采用ADMM方法来进行求解，文献[41]中给出了这两个问题详细的求解过程。

0.3.6 基于非负矩阵分解的聚类算法

非负矩阵分解(Non-negative Matrix Factorization,NMF) [44]是一种基于矩阵分解的方法，NMF算法将一个非负矩阵分解成两个子矩阵，并且这两个矩阵同时都保持非负性。对于一个非负矩阵 $\mathbf{A} \in \mathbb{R}^{n \times d}$ ，将其分解成两个非负的子矩阵 $\mathbf{B} \in \mathbb{R}^{n \times m}$ 与 $\mathbf{C} \in \mathbb{R}^{m \times d}$ ，NMF的数学表达形式如下：

$$\mathbf{A} \approx \mathbf{BC} \quad \text{s.t. } \mathbf{A} \geq 0, \mathbf{B} \geq 0, \mathbf{C} \geq 0 \tag{33}$$

关于（33）式有多种求解方法，其中一种方法采用梯度下降法来求解[45]， \mathbf{B} 与 \mathbf{C} 按照如下的方式迭代更新：

$$\mathbf{B} \leftarrow \mathbf{B} - 2\gamma (\mathbf{BCC}^T - \mathbf{AC}^T) \quad \mathbf{C} \leftarrow \mathbf{C} - 2\gamma (\mathbf{B}^T \mathbf{BC} - \mathbf{B}^T \mathbf{A}) \tag{34}$$

其中： γ 为学习的步长。由（33）式可知，如果 $d > m$ ， \mathbf{B} 相当于 \mathbf{A} 的一个低维表示。当 $m = k$ 时， $\mathbf{B} \in \mathbb{R}^{n \times k}$ ， $\mathbf{C} \in \mathbb{R}^{k \times d}$ ，如果 \mathbf{B} 被归一化(即有： $\mathbf{B}_{i,:} \mathbf{1}_{k \times 1} = 1$ 且 $0 \leq \mathbf{B}_{i,j} \leq 1$)，则 \mathbf{B} 相当于FCM算法中的隶属度矩阵， \mathbf{C} 相当于FCM算法中 k 个聚类中心点。文献[46, 47]已经证明，NMF算法与k-means算法是等价的。如果 $\forall B_{ij} \in \mathbf{B}$ 是一个在[0,1]连续区间内取值的实数，则NMF算法等价于FCM算法。关于利用NMF算法来进行聚类，已经发展不少的算法[48, 49, 50, 51]，并且有些已取得了不错的效果。

0.4 k-means算法的应用场景

随着信息社会发展，数据的价值越来越大，如何从数据中挖掘出有价值的信息，已经成为了亟待解决的问题。k-means算法作为一种重要的聚类方法，其原理简单，性能高效，目前受到许多学者的关注，并且已在众多领域得到广泛地应用。在本小节中，我们将介绍k-means算法在实际场景中的几种应用。

社会网络中的社团发现

随着社交媒体的发展，许多商业公司像Facebook、新浪微博等积累了大量的社交数据，不同于其他类型的数据，这些社交数据是以图形式组织的（如图8所示），并且组成了社会网络。在社会网络中，每个用户是网络的一个节点，人人之间通过边来连接，具有某些相似性的节点组成了一个社团。社团发现具有非常大的价值，但是，由于社交网络开放性的特点，会使得网络十分复杂、巨大，而k-means算法作为一种具有低复杂度的聚类算法，在社团发现任务中具有较明显的优势。不同于其他类型的数据，社会网络数据具有高纬、稀疏等特点，因此，在设计相关的聚类算法时，需要综合考虑这方面的问题。

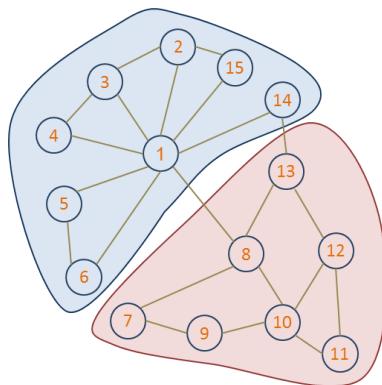


图 8: 社会网络中的社团发现

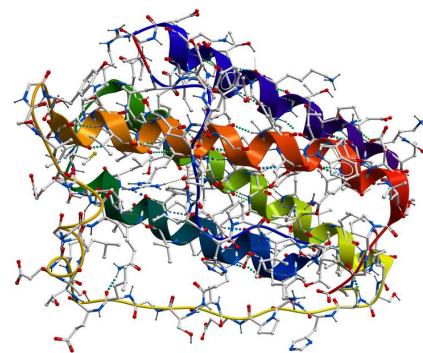


图 9: 蛋白质空间结构¹

商品推荐

在电子商务公司，购物网站积累了许多用户的购物数据，如果能从这些数据中，通过聚类算法，发现具有相似购物习惯的用户，就可以根据已有用户的购物记录，来定向推荐商品。定向商品推荐不但能帮助用户快速找到需要的商品，增加销售商品的机会，而且还能减少电子商务公司宣传成本，对提高公司的盈利能力具有非常重要的作用。

蛋白质结构预测

蛋白质是一种大分子化合物，在生命活动中起到重要作用，几乎每一项生命活动都有蛋白质的参与。在生命活动中，蛋白质种类多样，功能复杂，每种蛋白质的功能都由其独特的空间结构而决定。氨基酸经过脱水缩合形成肽链，若干条肽链经过在空间中的盘区和折叠后形成复杂的空间结构（如图9所示）。利用k-means算法来对蛋白质二级结构进行预测，对了解蛋白质在生命活动中的作用起着重要的作用。

¹http://tech.ifeng.com/a/20181207/45248935_0.shtml



图 10: 图像分割在自动驾驶中场景识别²

网络入侵检测

随着网络技术在生活中的大量应用，如何应对网络入侵的挑战，也成为信息安全领域一个重要的问题。最近几年各种计算机病毒的肆虐，让人们意识到，存储在计算机中的重要数据一旦丢失、损坏或者泄露，会给个人或企业带来的巨大的损失。通过对网络中的访问数据进行在线聚类分析，就可以发现当前网络中的入侵行为，这对保护用户隐私和保障企业的重要数据安全有着重要的作用，但是，网络入侵数据往往只占访问数据一小部分，因此，在设计相应的聚类算法时，需要综合考虑数据的非平衡性特点。

图像分割

图像处理作为人工智能一个热门方向，图像分割技术已在工业和生活中得到广泛地应用，如自动驾驶、人脸识别、智能交通等领域。图像分割就是把图像分成若干个特定的、具有独特性质的区域并提出感兴趣目标的技术和过程。它是由图像处理到图像分析的关键步骤。聚类算法先将图像空间中的像素用对应的特征空间点表示，根据它们在特征空间的聚集对特征空间进行分割，然后将它们映射回原图像空间，得到分割结果。

以上所列举的几种k-means聚类算法应用场景，表明了k-means聚类算法在我们的日常生活发挥着越来越重要的作用。最近几年，随着人工智能的火热，无监督学习正在越来越受到重视，自监督学习和无监督深度学习 [52, 53]等一些列相关的技术被提出，而k-means算法作为一种基础的聚类算法，我们相信在未来仍然有广阔的用武之地。

0.5 本章小结

在本章中，我们主要介绍一种基本的聚类算法——k-means算法。尽管k-means算法在1956年就被提出，但是至今为止，仍然受到许多学者的关注，即使在最近几年，关

²http://www.mindflow.com.cn/industries/autonomous_vehicles

于k-means算法的研究论文仍然有发表在ICML、NIPS、AAAI等计算机领域的顶级会议上[37, 34, 33]，这显示了这种古老的算法仍然有着较强的生命力。

在本章的第二节中，我们首先介绍了样本间的相似性度量方法，并给出数值型和分类型数据的各种距离度量，其实关于距离的度量远不止这些，更多的距离度量，可以参考文献[54, 55]；然后，我们进一步介绍了k-means 算法的基本原理和详细步骤，并用一个实例来讲解k-means算法的聚类过程。在本节的最后部分，我们分析了k-means算法的优点和缺点，并用实验来展示了这些缺点可能会对算法的性能产生一些不利的影响。

在第三节中，我们介绍了一些关于k-means算法的扩展算法。在前一节中，我们已经知道了k-means算法所具有的一些缺点会限制此算法的应用范围。在本章节的第一小节中，我们介绍了几种能自主决定类簇数目的算法，关于如何选择合适的类簇，这并不是一件容易的事；在本章节第二小节中，介绍了一些具有较好鲁棒性的k-means算法，在聚类算法中，能够应对噪声数据的聚类算法一般都能够看到DBSCAN算法或子空间学习方法的思想，尤其是低秩学习和稀疏学习，这几年属于一个比较热门的研究方向；在本章节第三小节中，主要是介绍了k-means++ 算法及其几种改进型算法，但是如何降低先验知识的依赖来选取合适的初始中心点，当前并没有最优的算法，关于聚类中心点的选择上，有一些算法能够自动寻找聚类中心点，如：均值漂移(Mean Shift) 算法 [56] 和密度峰值(Density Peaks) 聚类算法 [5]，但是这些算法仍然有一些需要用户指定的参数，并没有完全摆脱对用户经验的依赖；在本章节第四小节中，主要介绍了几种采用新的相异性度量来寻找数据中的类簇结构，从目前已有方法来看，关于采用新的相异性度量方法进行聚类的论文大量集中在分类型数据的聚类领域，关于数值型数据的聚类而提出新的距离度量，可以参考度量学习方法 [57, 58]；在本章节第五小节中，所介绍的模糊聚类方法，主要都属于软子空间聚类方法，在多年前，软子空间聚类属于一个热门方向，特征选择和降维都会带来信息的损失，而软子空间聚类通过分配一组权值使得同一特征在不同类簇空间中具有不同的重要性，因此，这种算法并没有带来信息损失，但却具有较好的聚类效果，关于更多的软子空间聚类方法可以参考文献 [59, 60]；在本章节第六小节中，我们主要介绍了非负矩阵分解的方法，已有的理论已经证明了，k-means算法与NMF算法具有等价性，在当前NMF仍然是属于一个较热门方向，尤其是NMF与谱聚类 [49]、半监督学习 [61]、多视角聚类 [62]、协同聚类 [63]等相结合，能够为当前算法的改进提供不少有益的思路。

在第四节里，我们主要介绍了几种k-means聚类算法应用的场景，可以看出k-means算法及其改进型算法，在实际应用中拥有广阔的前景。关于社会网络中社团发现，尽管k-means算法和NMF也得到一些应用，但是相对而言，基于谱聚类的算法更为常见，关于谱聚类的一些知识，可以参考文献 [16]；在推荐系统中，NMF是一种非常常见和重要的算法 [64]，在商业中得到较多的应用。蛋白质结构预测、网络入侵检测、图像分割等更是可以看到k-means算法的影响。最近几年，自监督学习和无监督深度学习的火热，证实了聚类是一个有着很大价值的问题，在未来仍然会持续受到学者们的关注。

参考文献

- [1] J. Han, J. Pei, M. Kamber, Data mining: concepts and techniques, Elsevier, 2011.
- [2] A. K. Jain, M. N. Murty, P. J. Flynn, Data clustering: a review, ACM Computing Surveys (CSUR) 31 (3) (1999) 264–323.
- [3] J. MacQueen, et al., Some methods for classification and analysis of multivariate observations, in: Proceedings of the Fifth Berkeley symposium on mathematical statistics and probability, Vol. 1, Oakland, CA, USA, 1967, pp. 281–297.
- [4] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, Density-based spatial clustering of applications with noise, in: Int. Conf. Knowledge Discovery and Data Mining, Vol. 240, 1996, pp. 226–231.
- [5] A. Rodriguez, A. Laio, Clustering by fast search and find of density peaks, Science 344 (6191) (2014) 1492–1496.
- [6] T. Zhang, R. Ramakrishnan, M. Livny, Birch: an efficient data clustering method for very large databases, in: ACM Sigmod Record, Vol. 25, ACM, 1996, pp. 103–114.
- [7] G. Karypis, E.-H. S. Han, V. Kumar, Chameleon: Hierarchical clustering using dynamic modeling, Computer (8) (1999) 68–75.
- [8] W. Wang, J. Yang, R. Muntz, et al., Sting: A statistical information grid approach to spatial data mining, in: International Conference on Very Large Data Bases, Vol. 97, 1997, pp. 186–195.
- [9] R. Agrawal, J. Gehrke, D. Gunopulos, P. Raghavan, Automatic subspace clustering of high dimensional data, Data Mining and Knowledge Discovery 11 (1) (2005) 5–33.
- [10] J. C. Baez, S. J. Olson, Uncertainty in measurements of distance, Classical and Quantum Gravity 19 (14) (2002) L121.
- [11] J. Liang, R. Li, Y. Qian, Distance: A more comprehensible perspective for measures in rough set theory, Knowledge-Based Systems 27 (2012) 126–136.

- [12] N. Parthalain, Q. Shen, R. Jensen, A distance measure approach to exploring the rough set boundary region for attribute reduction, *IEEE Transactions on Knowledge and Data Engineering* 22 (3) (2009) 305–317.
- [13] Y. Qian, J. Liang, C. Dang, Knowledge structure, knowledge granulation and knowledge distance in a knowledge base, *International Journal of Approximate Reasoning* 50 (1) (2009) 174–188.
- [14] N. Apollonio, M. Caramia, P. G. Franciosa, On the galois lattice of bipartite distance hereditary graphs, *Discrete Applied Mathematics* 190 (2015) 13–23.
- [15] X. Wu, V. Kumar, J. R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. Ng, B. Liu, S. Y. Philip, et al., Top 10 algorithms in data mining, *Knowledge and Information Systems* 14 (1) (2008) 1–37.
- [16] A. Y. Ng, M. I. Jordan, Y. Weiss, On spectral clustering: Analysis and an algorithm, in: *Advances in Neural Information Processing Systems*, Neural Information Processing Systems Foundation, 2002, pp. 849–856.
- [17] S. K. Popat, M. Emmanuel, Review and comparative study of clustering techniques, *International Journal of Computer Science and Information Technologies* 5 (1) (2014) 805–812.
- [18] A. K. Jain, Data clustering: 50 years beyond k-means, *Pattern Recognition Letters* 31 (8) (2010) 651–666.
- [19] T. M. Kodinariya, P. R. Makwana, Review on determining number of cluster in k-means clustering, *International Journal of Advance Research in Computer Science and Management Studies* 1 (6) (2013) 90–95.
- [20] D. Pelleg, A. W. Moore, X-means: Extending k-means with efficient estimation of the number of clusters, in: *Proceedings of the Seventeenth International Conference on Machine Learning*, Morgan Kaufmann Publishers Inc., 2000, pp. 727–734.
- [21] G. Hamerly, C. Elkan, Learning the k in k-means, in: *Advances in Neural Information Processing Systems*, Neural Information Processing Systems Foundation, 2004, pp. 281–288.
- [22] Z. Li, J. Liu, J. Tang, H. Lu, Robust structured subspace learning for data representation, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37 (10) (2015) 2085–2098.
- [23] K. Gregor, Y. LeCun, Learning fast approximations of sparse coding, in: *Proceedings of the 27th International Conference on International Conference on Machine Learning*, Omnipress, 2010, pp. 399–406.

- [24] S. Chawla, A. Gionis, k-means--: A unified approach to clustering and outlier detection, in: Proceedings of the 2013 SIAM International Conference on Data Mining, SIAM, 2013, pp. 189–197.
- [25] F. De La Torre, M. J. Black, A framework for robust subspace learning, *International Journal of Computer Vision* 54 (1-3) (2003) 117–142.
- [26] D. Mautz, W. Ye, C. Plant, C. Böhm, Towards an optimal subspace for k-means, in: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2017, pp. 365–373.
- [27] C. Lu, J. Feng, Z. Lin, T. Mei, S. Yan, Subspace clustering by block diagonal representation, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41 (2) (2019) 487–501.
- [28] M. B. Cohen, S. Elder, C. Musco, C. Musco, M. Persu, Dimensionality reduction for k-means clustering and low rank approximation, in: Proceedings of the Forty-seventh Annual ACM Symposium on Theory of Computing, ACM, 2015, pp. 163–172.
- [29] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, Y. Ma, Robust recovery of subspace structures by low-rank representation, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35 (1) (2012) 171–184.
- [30] D. Arthur, S. Vassilvitskii, k-means++: The advantages of careful seeding, in: Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, Society for Industrial and Applied Mathematics, 2007, pp. 1027–1035.
- [31] J. Y. Chan, A. P. Leung, Efficient k-means++ with random projection, in: 2017 International Joint Conference on Neural Networks (IJCNN), IEEE, 2017, pp. 94–100.
- [32] E. Bingham, H. Mannila, Random projection in dimensionality reduction: applications to image and text data, in: Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2001, pp. 245–250.
- [33] O. Bachem, M. Lucic, S. H. Hassani, A. Krause, Approximate k-means++ in sublinear time, in: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, Association for the Advance of Artificial Intelligence, 2016, pp. 1459–1467.
- [34] O. Bachem, M. Lucic, H. Hassani, A. Krause, Fast and provably good seedings for k-means, in: Advances in Neural Information Processing Systems, Neural Information Processing Systems Foundation, 2016, pp. 55–63.

- [35] B. Zhang, M. Hsu, U. Dayal, K-harmonic means-a data clustering algorithm, Hewlett-Packard Labs Technical Report HPL-1999-124 55.
- [36] B. Zhang, Generalized k-harmonic means–dynamic weighting of data in unsupervised learning, in: Proceedings of the 2001 SIAM International Conference on Data Mining, SIAM, 2001, pp. 1–13.
- [37] J. Xu, K. Lange, Power k-means clustering, in: Proceedings of the 36th International Conference on Machine Learning, Proceedings of Machine Learning Research, Long Beach, California, USA, 2019, pp. 6921–6931.
- [38] M. de Carvalho, Mean, what do you mean?, *The American Statistician* 70 (3) (2016) 270–274.
- [39] X. Zhu, C. Change Loy, S. Gong, Constructing robust affinity graphs for spectral clustering, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2014, pp. 1450–1457.
- [40] J. C. Bezdek, R. Ehrlich, W. Full, Fcm: The fuzzy c-means clustering algorithm, *Computers & Geosciences* 10 (2-3) (1984) 191–203.
- [41] Z. Deng, K.-S. Choi, F.-L. Chung, S. Wang, Enhanced soft subspace clustering integrating within-cluster and between-cluster information, *Pattern Recognition* 43 (3) (2010) 767–781.
- [42] L. Jing, M. K. Ng, J. Z. Huang, An entropy weighting k-means algorithm for subspace clustering of high-dimensional sparse data, *IEEE Transactions on Knowledge & Data Engineering* (8) (2007) 1026–1041.
- [43] N. Parikh, S. Boyd, et al., Proximal algorithms, *Foundations and Trends® in Optimization* 1 (3) (2014) 127–239.
- [44] D. D. Lee, H. S. Seung, Learning the parts of objects by non-negative matrix factorization, *Nature* 401 (6755) (1999) 788–791.
- [45] P. O. Hoyer, Non-negative matrix factorization with sparseness constraints, *Journal of Machine Learning Research* 5 (11) (2004) 1457–1469.
- [46] C. Ding, X. He, H. D. Simon, On the equivalence of nonnegative matrix factorization and spectral clustering, in: Proceedings of the 2005 SIAM International Conference on Data Mining, SIAM, 2005, pp. 606–610.
- [47] C. Bauckhage, K-means clustering is matrix factorization, arXiv preprint arXiv:1512.07548.

- [48] T. Li, C.-c. Ding, Nonnegative matrix factorizations for clustering: A survey, in: Data Clustering, Chapman and Hall/CRC, 2018, pp. 149–176.
- [49] S. Huang, H. Wang, T. Li, T. Li, Z. Xu, Robust graph regularized nonnegative matrix factorization for clustering, *Data Mining and Knowledge Discovery* 32 (2) (2018) 483–503.
- [50] Y. Sheng, M. Wang, T. Wu, H. Xu, Adaptive local learning regularized nonnegative matrix factorization for data clustering, *Applied Intelligence* 49 (6) (2019) 2151–2168.
- [51] H. Kameoka, T. Higuchi, M. Tanaka, et al., Nonnegative matrix factorization with basis clustering using cepstral distance regularization, *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)* 26 (6) (2018) 1025–1036.
- [52] P. Sermanet, C. Lynch, Y. Chebotar, J. Hsu, E. Jang, S. Schaal, S. Levine, G. Brain, Time-contrastive networks: Self-supervised learning from video, in: 2018 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2018, pp. 1134–1141.
- [53] K. Lin, J. Lu, C.-S. Chen, J. Zhou, M.-T. Sun, Unsupervised deep learning of compact binary descriptors, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41 (6) (2018) 1501–1514.
- [54] Y. Qian, Y. Li, J. Liang, G. Lin, C. Dang, Fuzzy granular structure distance, *IEEE Transactions on Fuzzy Systems* 23 (6) (2015) 2245–2259.
- [55] L. H. Son, Generalized picture distance measure and applications to picture fuzzy clustering, *Applied Soft Computing* 46 (2016) 284–295.
- [56] D. Comaniciu, P. Meer, Mean shift: A robust approach toward feature space analysis, *IEEE Transactions on Pattern Analysis & Machine Intelligence* (5) (2002) 603–619.
- [57] B. Kulis, et al., Metric learning: A survey, *Foundations and Trends in Machine Learning* 5 (4) (2013) 287–364.
- [58] F. Wang, J. Sun, Survey on distance metric learning and dimensionality reduction in data mining, *Data mining and knowledge discovery* 29 (2) (2015) 534–564.
- [59] Z. Deng, K.-S. Choi, Y. Jiang, J. Wang, S. Wang, A survey on soft subspace clustering, *Information sciences* 348 (2016) 84–106.
- [60] H.-P. Kriegel, P. Kröger, A. Zimek, Subspace clustering, *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 2 (4) (2012) 351–364.

- [61] H. Lee, J. Yoo, S. Choi, Semi-supervised nonnegative matrix factorization, *IEEE Signal Processing Letters* 17 (1) (2009) 4–7.
- [62] L. Zong, X. Zhang, L. Zhao, H. Yu, Q. Zhao, Multi-view clustering via multi-manifold regularized non-negative matrix factorization, *Neural Networks* 88 (2017) 74–89.
- [63] H. Ma, W. Zhao, Q. Tan, Z. Shi, Orthogonal nonnegative matrix tri-factorization for semi-supervised document co-clustering, in: *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, Springer, 2010, pp. 189–200.
- [64] X. Luo, M. Zhou, Y. Xia, Q. Zhu, An efficient non-negative matrix-factorization-based approach to collaborative filtering for recommender systems, *IEEE Transactions on Industrial Informatics* 10 (2) (2014) 1273–1284.