聚类算法对比研究

学院：计算机学院

专业：大数据与人工智能

学号：21215122

姓名：何峙

授课老师：梁上松、王昌栋

摘 要 聚类是一种广泛应用于数据挖掘的算法。现实中的很多数据的类别标记可能是未知的，要揭示数据蕴含的性质和内在规律，则需要通过一定的数据分析技术，这一般被称为“无监督学习”的技术。而聚类则是其中占用重要地位的技术，其研究最多，应用最广，衍生出许许多多的算法，每种算法又分为多种形式以适用在不同场景下。本文通过对这些聚类算法进行对比，阐述各种聚类算法的效用，让读者了解其在特定场景下优缺点。

关键词 聚类，数据挖掘，无监督学习

**目录**

**[第一章 引言 1](#_Toc7470)**

**[第二章 性能度量 1](#_Toc21362)**

[一、外部指标 2](#_Toc21021)

[二、内部指标 3](#_Toc8814)

**[第三章 聚类算法 4](#_Toc31346)**

[一、基于划分的聚类 4](#_Toc19887)

[二、基于密度的聚类 5](#_Toc24460)

[三、基于层次的聚类 7](#_Toc28387)

**[第四章 研究前沿 8](#_Toc31495)**

**[第五章 总结 9](#_Toc19796)**

**[参考文献 10](#_Toc17074)**

1. 引言

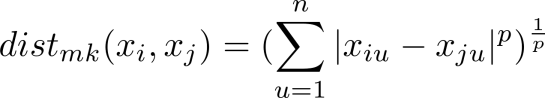
聚类是一种重要的无监督学习技术，它可以将没有类别标记的数据集划分为若干个互不相相交的子集，从而使数据样本分别位于不同的“簇”当中，这些“簇”也就是互不相交的类别，使得数据可进一步用于下游的数据分析任务。例如，在一些图片应用中需要根据用户照片的内容进行类别划分，譬如海滩的照片归为一类，山脉的照片归为另一类，等。但用户并没有人工划分这些照片的类别，此时往往先对这些照片进行特征提取，根据图片特征进行聚类，再根据聚类结果将每个簇定义为一个类别，然后再基于这些类别训练模型，以进行下一步的图片细粒度识别等功能应用。这时，往往会考虑：使用哪种聚类算法更适合于特定场景？还有，如何衡量这些聚类算法的性能好坏？接下来本文将从如下几个方面进行相关阐述：

* 第二章，阐述聚类算法所使用的性能度量，说明数据的相似度等概念；
* 第三章，列举三种不同场景下常用的聚类算法概念及其设计思想；
* 第四章，在传统的聚类算法基础上，展示某些前沿的聚类算法研究；
* 第五章，聚类算法总结。

1. 性能度量

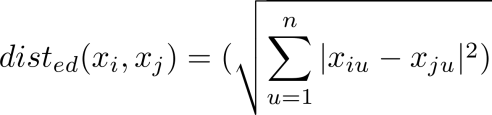
聚类算法的好坏直接影响着下游数据分析任务的优劣，那么如何衡量聚类算法的性能就相当关键。一般来说，聚类的核心性能指标是：使簇内相似度尽可能高，使簇间相似度尽可能低。相似度有时也被称为“数据距离”，是一种衡量数据之间距离的度量。假设xi, xj是两个样本对应的向量，以下列举的聚类中常用的相似度计算方法：

* 闵可夫斯基距离：



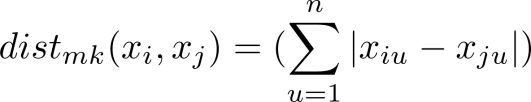
* 欧氏距离：

当闵可夫斯基距离的p=2时，即为欧氏距离：

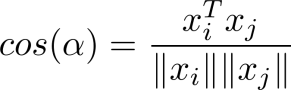


* 曼哈顿距离：

当闵可夫斯基距离的p=1时，即为曼哈顿距离：

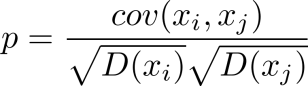


* 余弦相似度：



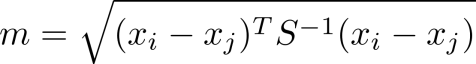
计算的结果是两个向量之间的夹角。所以值越小，两向量呈夹角，相似度越高。否则呈钝角，相似度越低。

* 皮尔逊相关系数：



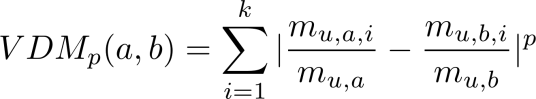
其中cov(xi, xj)是两个向量的协方差，D(xi)是向量的方差。皮尔逊相关系数的取值范围在[-1,1]之间，值越趋于0表面向量越不相关，越趋于-1表明向量越负相关，而越趋于1表明向量越正相关。

* 马氏距离：

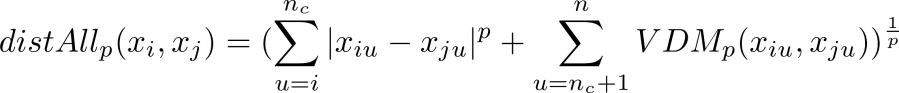


其中S为两个向量对应的协方差矩阵。

以上这些度量常用于连续的属性，或者更确切的称为“有序的属性”。而对于样本中离散的、无序的属性，譬如属性的取值为学科类别={物理，化学，法律，经济}，又该如何衡量它们之间的相似度？可以考虑采用VDM数值（Value Difference Metric）衡量：



其中mu,a表示在属性u上取值为a的样本数，mu,a,i表示在第i个簇中在属性u上取值为a的样本数。这时，可以将闵可夫斯基距离和VDM相结合，把含有有序属性和无序属性的样本相似度计算统一起来，如下计算：



有了以上这些距离度量，就可以进行性能度量。

性能度量指标又分为外部指标和内部指标两大类。外部指标是将聚类结果跟某个参考模型相比较，譬如人类专家定义的参考模型。内部指标是不利用外部参考模型而直接考察聚类结果。

一、外部指标

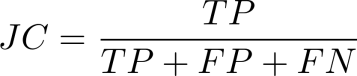
由于有外部参考模型的介入，聚类算法将各样本划分到对应簇的情况可能与外部参考模型不一致。先定义表1的混淆矩阵以方便说明问题。

Table 1 Definition of Confusion Matrix for binary classification

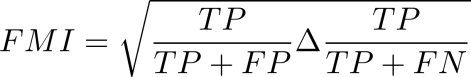
|  |  |  |  |
| --- | --- | --- | --- |
|  | Acutal Value | | |
| Predicted Value |  | Positive | Negative |
| Positive | TP | FP |
| Negative | FN | TN |

则有如下外部衡量指标：

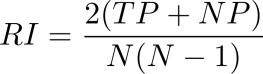
* Jaccard指数，也称Jaccard相似度：



* FM指数：



* Rand指数：



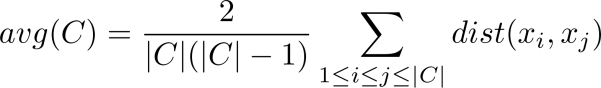
其中N为样本的数量。

显然上述性能度量的结果均在[0,1]区间，取值越大越好。

二、内部指标

假设聚类结果的簇划分为C={C1, C2, ..., Ck} ,先定义如下计算量：

* 簇C内样本的平均距离：



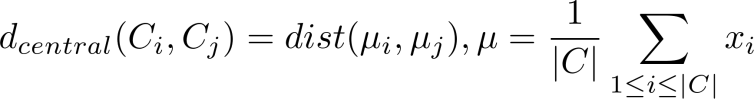
* 簇C内样本间的最远距离

wpsoffice

* 两个簇最近样本间的距离

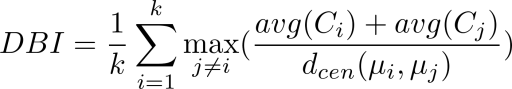
wpsoffice

* 两个簇中心点的距离



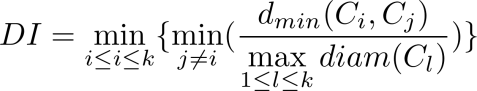
由此可导出以下常用的内部指标：

* DB指数：



显然值越小越好。

* Dunn指数



Dunn指数则相反，值越大越好。

以上提到的距离度量和性能度量只是聚类中比较常用的指标，实际研究中不仅仅只有这些，譬如F值、互信息等等也是不可忽略的指标，此外指标也不一定预先指定，甚至可以嵌入到聚类过程中进行学习，如Xing等[1]就提到了一种距离度量学习方法，这里不另做介绍了。

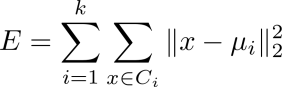
1. 聚类算法

聚类算法种类有很多，以下将根据数据聚类进行中的组织方式回顾三大类常用的聚类算法。

一、基于划分的聚类

这类聚类算法一般采用贪心策略，通过定义一个最优化的目标函数，通过迭代的方式使样本点不断归类于这K个代表点所属的类别，同时逐步提高聚类的效果。

K-means[2]是其中的典型代表。其目标函数是最小化所有样本点的平方误差：



算法首先需要指定要划分的簇类数量K，随机选择K个代表点作为初始向量，然后不断迭代的判断每个样本点分别与这些代表点的距离，把与距离小的代表点所属的类别分配给迭代中的样本点，并且每迭代一次，重新计算代表点为其所归属的样本点的中心点为新的代表点，一直迭代到所有样本点均未更新为止。所以，从图1可见，K-means是一个不断将样本聚集在最佳簇中心点（代表点）的过程。K-means的优点有：欧氏距离容易计算、算法不依赖于数据的先后排序，等。其缺点也是相当明显，主要有：

* 聚类簇数量要预先指定，但是又难以指定合适的簇类数量K；
* 对异常点敏感；
* 对于线性不可分的数据效果较差。这些通常也是基于划分的聚类算法的固有缺点。

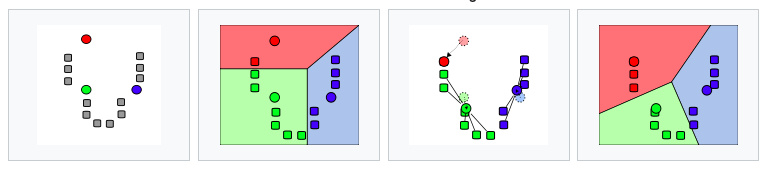
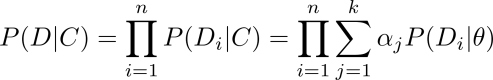


Fig.1. Procedure of K-means, from left to right. (picutre by wikipedia).

K-medoids是K-means算法的一个改进，其与K-means的最大区别是选择中心点的方法不同：K-medoids每次选取的中心点必须是样本点，而K-means选取的中心点可以是样本点以外的点。K-medoids选取中心点时，需要遍历簇中所有的点，在前簇中选择所有其他点到中心点距离之和最小的点为中心点。通过这样的方式K-medoids可以大大减少异常点对聚类结果影响。

还有一种常用的基于概率统计的方法，也属于基于划分的聚类算法。可以假定数据空间中有一个隐藏的类别分布，分布的类别数量为K，数据样本就是通过这K个类别即簇产生的：先以一定概率选取一个簇，然后从簇中以一定的概率选取一个样本。所以接下来就是计算这K个簇和它们以概率产生数据集的似然。假定样本是独立采样的，因此对于数据集D={D1, D2, D3, …}有：



其中 wpsoffice 是选择每个簇的概率， 所以选择K个簇的总概率为1，即 ，wpsoffice 是指定概率分布的参数，譬如实际应用中可选择高斯分布，则参数为n维均值向量和n x n阶的协方差矩阵。进行参数估计可使用EM算法[3]或其他方法。得出参数后即可通过后验概率P(Ci | xi) 得出样本所属的簇类。

表2总结了基于划分的聚类算法的时间复杂度，其中GMM[4]为高斯混合分布聚类算法，k表示聚类的数量，t为迭代次数，n为样本数量。

Table.2. Time complexity for partition alorithm

|  |  |  |
| --- | --- | --- |
| K-means | K-mediods | GMM |
| O(k\*t\*n) | O(k(n-k)2) | O(k\*t\*n2) |

二、基于密度的聚类

这类算法以样本之间的紧密程度如可连接性为考察重点，基于可连接性将样本不断扩展聚类簇以获得最终的聚类效果。其中DBSCAN[5]是此聚类算法典型代表。

DBSCAN需要预定参数（wpsoffice, MinPts），首先定义如下几个概念：

* wpsoffice-邻域：wpsoffice；
* 核心对象：若xj的wpsoffice-邻域至少包含MinPts个样本，则xj是一个核心对象；
* 密度直达：若xj位于xi的wpsoffice-邻域中，且xi是核心对象，则称xj由xi密度直达；
* 密度可达：对于xi与xj，若存在样本序列p1,p2,...,pn，其中p1=xi，pn=xj，且pi+1有pi密度可达，则称xj由xi密度可达；
* 密度相连：对xi与xj，若存在xk使得xi与xj均由xk密度可达，则称xi与xj密度相连。

基于以上定义，具体来说，该算法根据给定的参数wpsoffice，MinPts）找出所有核心对象，以任一核心对象作为出发点，找出由其密度可达的样本生成聚类簇，不断迭代至所有核心对象都被访问过为止。如图2所示，该算法可看作是一个由种子点（即核心对象）不断扩大连通区域以形成簇的过程。DBSCAN的优点主要是其可用于线性不可分的数据集，其聚类效果一般更加优秀，而且不需要指点聚类的簇数量。但其需要指定两个难以适用于样本数据集的参数（wpsoffice，MinPts），甚至数据集的不同部分可能需要指定不同的参数以达到良好的聚类效果，所以其对于参数过于敏感。



Fig.2. Procedure of DBSCAN.(picture by wikipedia)

OPTICS[6]是DBSCAN的改进算法，其也需要指定相同的两个参数（wpsoffice，MinPts），但对wpsoffice不再敏感，只要确定MinPts，wpsoffice 的轻微变化并不影响聚类结果。除了继承DBSCAN的定义，OPTICS还多了两个定义：

* 核心距离：使一个样本点成为核心点（即核心对象）的最小半径，在给定邻域半径 wpsoffice 和MinPts参数的前提下，核心距离可以比给定的 wpsoffice 更小；
* 可达距离：样本点到核心点的距离。

其算法过程如下：

1. 创建两个队列：待处理队列用于存储核心点及其密度直达的点, 并按可达距离升序排列；结果队列用于存储样本点的输出次序。结果队列中的点为处理之后的样本；
2. 选取一个未处理的核心点, 将其放入结果队列，同时计算邻域内样本点的可达距离，按照可达距离升序将样本点依次放入待处理队列。
3. 若待处理队列为空，则回到步骤（2）以重新选取处理数据。否则，从待处理队列中提取第一个样本，如果为核心点, 则计算可达距离，将可达距离最小的点放入结果队列。如果不是核心点 则重复步骤（2）；如果待处理队列中已经存在直接密度可达点，如果此时新的可达距离小于旧的可达距离，则用新可达距离取代旧可达距离，待处理队列重新排序（因为一个对象可能有多个核心对象可达）；如果待处理队列中不存在该直接密度可达样本点，则插入该点，并对有序队列重新排序；
4. 迭代（2）、（3），直到数据集中所有点都处理完成，则算法结束，输出结果队列中有序样本点。

OPTICS不显式产生聚类簇，而是生成一个簇的排序，这个排序表示了个样本点基于密度的聚类结构。

DBSCAN在暴力迭代下的时间复杂度是O(n2)，其作者在论文[5]中提到可使用R\*-树[7]作为数据结构，这是一种多维空间下的平衡树，所以其与OPTICS的时间复杂度可以优化为O(n\*logn)。

三、基于层次的聚类

层次聚类算法在不同层次上对数据集进行划分，采用自底向上的聚合策略，或者采用自顶向下的分拆策略，使数据集形式树形结构，图3展示了一个对字符串进行层次聚类的例子。

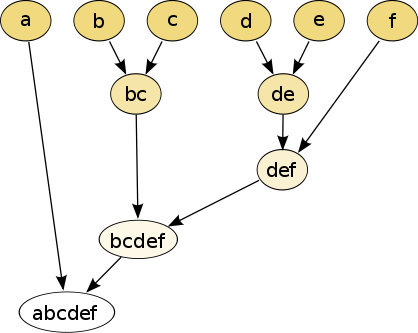


Fig.3. Example of hierarchical clustering for strings

AGNES[8]是一种经典的自底向上的层次聚类算法。它先指定要聚类的簇个数，接着将数据集中的每个样本看作是一个个初始的簇，然后在算法运行过程中的每一步找出距离最近的两个聚类簇进行合并，不断重复迭代直到达到预设的聚类簇个数。关于簇之间距离的计算可参考第二节，一般可用最小距离（由两个簇最近的样本决定），最大距离（由两个簇最远的样本决定），平均距离等衡量（由两个簇所有样本决定）。层次聚类的优点一般为：

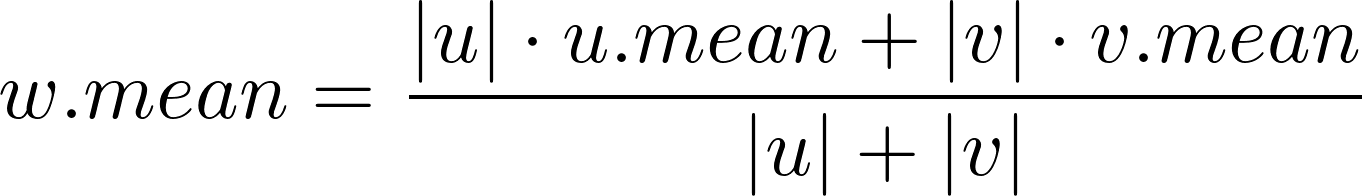
* 可解析性较强，形成树型的聚类分析图可以了解聚类的层次；
* 能应用与任意的相似度和距离度量；
* 对任意的数据类型适用性强，容易扩展。

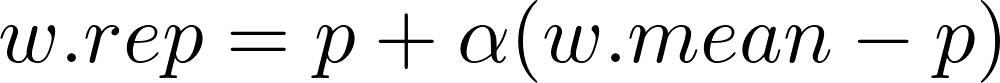
其缺点一般有：

* 难以选择适当的停止迭代时机，因为也需要人工指定聚类的簇个数；
* 一旦簇被合并便不能撤销；
* 时间复杂度较高。

CURE算法[9]也是对AGNES的一种改进，其与AGNES的区别是不使用所有点或中心点加上距离来表示一个簇，而是从每个簇中抽取固定数量、分布较好的点作为此簇的代表点。算法过程如下：

1. 开始也是每个样本点单独称为一个簇；
2. 从每个簇中，选择一小部分作为簇的代表点，选出的点之间尽量相距较远；
3. 将每个代表点移动一段距离：距离其位置到簇中心的距离乘以一个收缩因子，如0.2，使它们更加靠近中心点；簇的中心点及代表点的计算公式如下：





其中w为合并好的簇，u和v为待合并的两个簇，wpsoffice 为收缩因子，p为待考察的样本点；

1. 簇之间的距离定义为移动后的任意两个代表点之间的最短距离；
2. 当两个簇的某对代表点之间的距离小于指定的阈值，就将两个簇合并，重复该过程直至没有足够接近的簇为止；
3. 遍历每个样本点，将其与代表点比较，将点分配给最近的簇；

由于代表点的使用，使模型增加了一定的随机性，可以减少噪音数据对聚类的影响，其收缩特性可以调整模型匹配非高斯分布的应用场景，但缺点仍然是较高时间复杂度，与AGNES一样都为O(n2)。

1. 研究前沿

随着大数据的发展，数据量越来越大，样本空间也更加高维，这要求聚类算法的设计需要在保证一定时间复杂度的情况下完成对数据的处理。最近一些年来，一些新颖的聚类算法越来越引起人们的关注。

核聚类算法是利用非线性映射将样本数据从输入空间映射到一个更加高维的特征空间进行聚类分析。将数据映射到高维空间的好处是数据由原来线性不可分可以变得线性可分，因此传统聚类算法跟核方法结合后，可能会使算法更加高效，可解析性也更加强。 MacDonald D等[10]提出的kernel SOM算法，是核方法与传统SOM[11]聚类方法的集合。还有Zhong-dong Wu等[12]提出kernel FCM也是核方法与FCM[13]结合的核聚类方法。再如Bin Zhao等[14]提出的MKC方法，是MMC[15]方法的一种改良。MMC尝试找到一个最大间隔的超平面进行聚类，而MKC可以同时找到最大间隔超平面、最优聚类标签和最优核。

集成聚类算法借鉴了机器学习中的集成学习，为了提高聚类结果的准确性与稳定性，通过集成多个基聚类算法结果可以产生一个较优的结果。一个聚类算法模型对特定的数据集可产生良好的效果，但可能对其他数据集效果则很差，这时需要用多个独立的基聚类器分别对原始数据集进行聚类，然后使用某种集成方法对进行聚类结果进行提升，最终获得一个更好的聚类效果。集成聚类算法流程如图4所示，可见集成聚类的关键是找到合适的Consensus函数，即结合策略，将多个聚类结果映射到一个。Consensus函数的选择有多种方案，如基于关联矩阵，基于图划分，基于重标记与投票表决，基于遗传算法，等等。Evgenia Dimitriadou,等[16]提出的VM模型就是基于投票表决的集成聚类方法，每个数据点都有一个对类簇的从属程度（belongingness）,最终选取从属程度最大的聚类结果作为最终结果。ALN Fred等[17]提出的EAC方法将不同聚类划分的结果形成一个邻近度矩阵（proximity matrix），矩阵的元素衡量了样本之间对于不同聚类划分中同属一个簇的平均次数，最后对这个矩阵进行层次聚类算法得到最终的聚类结果。Dong Huang等[18]提出了一种可以利用多种度量以组合多个随机子空间的聚类方法——MDEC，它通过随机化一个具有比例指数的相似核（scaled exponential similarity kernel）产生大量不同的度量，这些度量组合为多个随机子空间从而形成大的度量与子空间配对集合，然后使用谱聚类算法对这些集合进行聚类。

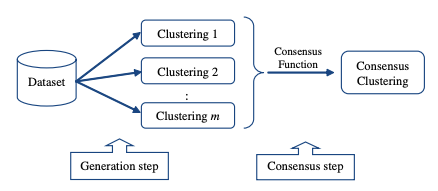


Fig.4. general process of ensemble clustering.[19]

除此之外，当前聚类算法与其他学科的结合也产生了许多特定领域内的新型算法，如基于量子理论的聚类，基于图谱的聚类，基于群体智能的聚类，等，都十分具有前景。

1. 总结

聚类是机器学习中算法繁多而且发展最快的领域，一个重要原因是数据类标识的缺失导致不存在客观的分类标准。对于特定的应用场景，给定数据集，采用不同的表征样本分布的紧密程度，总能设计出新的聚类算法。本文的主要目的是阐述各种经典的聚类算法的基本的、核心的设计思想，通过对比各种算法的优缺点，让读者了解聚类在数据挖掘中的重要性和价值，这对数据挖掘理论的系统性和通用性研究是十分有帮助的。

参考文献

[1] E. P. Xing, A. Y. Ng, M. I. Jordan, and S. Russell, “Distance metric learning, with application to clustering with side-information,” p. 8.

[2] J. A. Hartigan and M. A. Wong, “Algorithm AS 136: A K-Means Clustering Algorithm,” *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 28, no. 1, pp. 100–108, 1979, doi: 10.2307/2346830.

[3] G. J. McLachlan and T. Krishnan, *The EM Algorithm and Extensions*. John Wiley & Sons, 2007.

[4] C. Rasmussen, “The Infinite Gaussian Mixture Model,” in *Advances in Neural Information Processing Systems*, 1999, vol. 12. Accessed: Jul. 08, 2022. [Online]. Available: https://proceedings.neurips.cc/paper/1999/hash/97d98119037c5b8a9663cb21fb8ebf47-Abstract.html

[5] M. Ester, H.-P. Kriegel, and X. Xu, “A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise,” p. 6.

[6] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander, “OPTICS: Ordering Points To Identify the Clustering Structure,” p. 12.

[7] N. Beckmann, R. Schneider, B. Seeger, and P. Informatlk, “The R-Tree: An Efficient and Robust Access Method for Points and Rectangles,” p. 10.

[8] L. Kaufman and P. J. Rousseeuw, Eds., “Agglomerative Nesting (Program AGNES),” in *Wiley Series in Probability and Statistics*, Hoboken, NJ, USA: John Wiley & Sons, Inc., 1990, pp. 199–252. doi: 10.1002/9780470316801.ch5.

[9] S. Guha, R. Rastogi, and K. Shim, “CURE: an efficient clustering algorithm for large databases,” *SIGMOD Rec.*, vol. 27, no. 2, pp. 73–84, Jun. 1998, doi: 10.1145/276305.276312.

[10] D. MacDonald and C. Fyfe, “The kernel self-organising map,” in *KES’2000. Fourth International Conference on Knowledge-Based Intelligent Engineering Systems and Allied Technologies. Proceedings (Cat. No.00TH8516)*, 2000, vol. 1, pp. 317–320 vol.1. doi: 10.1109/KES.2000.885820.

[11] T. Kohonen, *Self-Organizing Maps*. Springer Science & Business Media, 2012.

[12] Z. Wu, W. Xie, and J. Yu, “Fuzzy C-means clustering algorithm based on kernel method,” in *Proceedings Fifth International Conference on Computational Intelligence and Multimedia Applications. ICCIMA 2003*, Sep. 2003, pp. 49–54. doi: 10.1109/ICCIMA.2003.1238099.

[13] J. C. Bezdek, R. Ehrlich, and W. Full, “FCM: The fuzzy c-means clustering algorithm,” *Computers & Geosciences*, vol. 10, no. 2–3, pp. 191–203, Jan. 1984, doi: 10.1016/0098-3004(84)90020-7.

[14] B. Zhao, J. T. Kwok, and C. Zhang, “Multiple Kernel Clustering,” in *Proceedings of the 2009 SIAM International Conference on Data Mining*, Apr. 2009, pp. 638–649. doi: 10.1137/1.9781611972795.55.

[15] E. Stone and J. Ayroles, “Modulated modularity clustering as an exploratory tool for functional genomic inference.,” *PLoS Genet*, vol. 5, no. 5, p. e1000479, 2009.

[16] E. Dimitriadou, A. Weingessel, and K. Hornik, “Voting-Merging: An Ensemble Method for Clustering,” in *Artificial Neural Networks — ICANN 2001*, vol. 2130, G. Dorffner, H. Bischof, and K. Hornik, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 217–224. doi: 10.1007/3-540-44668-0\_31.

[17] A. L. N. Fred and A. K. Jain, “Combining multiple clusterings using evidence accumulation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 6, pp. 835–850, Jun. 2005, doi: 10.1109/TPAMI.2005.113.

[18] D. Huang, C.-D. Wang, J.-H. Lai, and C.-K. Kwoh, “Toward Multidiversified Ensemble Clustering of High-Dimensional Data: From Subspaces to Metrics and Beyond,” *IEEE Transactions on Cybernetics*, pp. 1–14, 2021, doi: 10.1109/TCYB.2021.3049633.

[19] S. Vega-Pons and J. Ruiz-Shulcloper, “A SURVEY OF CLUSTERING ENSEMBLE ALGORITHMS,” *Int. J. Patt. Recogn. Artif. Intell.*, vol. 25, no. 03, pp. 337–372, May 2011, doi: 10.1142/S0218001411008683.