聚类算法研究综述

陈新泉[1,2] 周灵晶[1] 刘耀中[3]

[1](重庆三峡学院智能信息处理与控制重点实验室 重庆 404100)
[2](电子科技大学大数据研究中心 成都 611731)
[3](中国石油塔里木油田分公司 库尔勒 841000)

摘 要 聚类是数据挖掘研究领域的一种重要数据预处理方法，其目的是从无标签数据集中获得有价值数据集的内在分布结构，进而简化数据集的描述。历经几十年的研究，针对不同应用和数据特性已出现了千余种不同的聚类算法，但不同的聚类算法都有其特定的适用范围和不足。传统的聚类算法大致可分为划分聚类方法、层次聚类方法、密度聚类方法、网格聚类方法、模型聚类方法等。通过对传统聚类方法的回顾和总结，文章重点介绍了近年来出现的同步聚类算法、信念传播聚类算法和密度峰值聚类算法，并针对以上聚类算法的应用及发展方向进行了论述。

关键词 数据挖掘；聚类；信念传播；同步聚类；密度峰值
中图分类号 TP 181 **文献标志码** A

Review on Clustering Algorithms

CHEN Xinquan[1,2] ZHOU Lingjing[1] LIU Yaozhong[3]

[1](*Key Laboratory of Intelligent Information Processing and Control, Chongqing Three Gorges University, Chongqing* 404100, *China*)
[2](*Big Data Research Center, University of Electronic Science and Technology of China, Chengdu* 611731, *China*)
[3](*Tarim Oilfield Company of PetroChina, Koler* 841000, *China*)

Abstract Clustering is an important research topic in data mining domain for data preprocessing. Clustering is an unsupervised learning method that tries to find out some obvious clusters in the unlabeled data. It is usually performed by maximizing the similarity of inner-clusters and minimizing the similarity of inter-clusters. A lot of clustering algorithms have been proposed to solve various tasks and data properties in the past decades. However, all existing clustering methods have their own pros and cons, and there still lack of a clustering method with universality. Traditional clustering methods are usually classified into partitioning methods, hierarchical methods, density-based methods, grid-based methods and model-based methods. With a brief review to classical clustering methods, we put emphasis on introducing some recent emerging clustering methods like synchronization clustering algorithm, affinity propagation algorithm and density peaks algorithm. Based on the analysis and comparison of these algorithms, their potential applications and research directions are also discussed.

Keywords data mining; clustering; affinity propagation; synchronization clustering; density peak

收稿日期：2016-10-30 **修回日期**：2017-03-24
基金项目：重庆市基础与前沿研究计划项目(cstc2016jcyjA0521、cstc2016jcyjA2033)；重庆三峡学院科学研究项目计划资助(16PY08)；重庆市高校市级重点实验室资助项目(C16)
作者简介：陈新泉(通讯作者)，博士，教授，研究方向为数据挖掘，E-mail：chenxqscut@126.com；周灵晶，主管护师，研究方向为医疗信息分析等；刘耀中，工程师，研究方向为地球勘探数据处理。

1 引 言

21 世纪是一个信息化、数据化和知识化的时代，信息技术正改变着人类社会的方方面面。当前，人们已经认识到，只有将数据转化成信息或从数据中挖掘出知识才能发挥数据的更大价值。传统的数据挖掘算法随着大数据时代的到来，表现得越来越力不从心。随着第二代 Web 的发展及物联网、云计算和大数据技术的兴起，我们需要开发更为高效的数据挖掘工具和算法来处理不同类型、不同属性及不同维度的海量数据以支持正确的决策和行动。

在统计学领域，与聚类分析功能相似的是多元统计分析。其分析对象一般是数值型数据集，其目的是通过计算出数据集的一些统计参数信息，获知数据集的分布特性及分布状况。从数据挖掘的角度来看，聚类分析的目的是获得那些有意义、有实际价值的数据集的内在分布结构，进而简化数据集的描述。与纯粹的理论研究不同，数据挖掘领域的聚类分析更多的是属于一种应用驱动型的智能数据分析技术。

从机器学习的视角来看，作为数据挖掘的一个重要分支——聚类，属于一种非监督学习方法，它试图在无标签的数据集中发现其分布状况或模式。通常，我们认为同一聚类中的数据点比不同聚类的数据点具有更大的相似性[1]。随着数据挖掘的兴起，聚类方法开始用于分析实际问题中经常出现的具有多种类型属性和复杂分布结构的数据集。聚类被广泛应用到多个领域，如机器学习、模式识别、图像处理、信息检索等。

聚类算法的研究已经开展了几十年，到目前为止，已公开发表了近千种聚类算法，但没有一种聚类算法敢声称是通用的、普适的，几乎所有的聚类算法都存在某种缺点。例如，一些聚类算法更适合或只能处理一定类型的数据；一些聚类算法擅长处理具有某种特殊分布结构的数据而不能很好地处理其他分布的数据。现实世界中的数据，或是具有复杂的分布，或是具有多种数据类型，或是数据量巨大，或是含有噪声，或是含有孤立点等。为应对不同背景下的不同聚类任务，研究人员一直在研究可处理不同数据类型，适合于不同任务的、更先进的聚类方法。

2 聚类算法简介

由于聚类分析属于一个交叉研究领域，融合了多个学科的方法和技术，故可以从多种角度、多个层次来分析现有的聚类分析算法。Agarwal 等[1]关于数据聚类的经典长文从统计模式识别的视角总结了 1999 年之前的经典模式聚类方法。该文对许多著名聚类算法的发展背景、应用状况（如图像分割、对象识别及信息检索）等作了一个很好的总结性综述。到目前为止，这篇回顾性论文在聚类分析领域依然具有非常重要的地位和参考价值。Qian 和 Zhou[2]从聚类标准、聚类表示及算法框架角度分析了多个流行的聚类算法，该文在国内聚类算法领域具有一定的地位。Grabmeier 和 Rudolph[3]从数据挖掘的角度（如相似度和距离度量的严格区分、应用到聚类中的相关优化标准等）分析了一些聚类方法，还讨论了 IBM 公司的智能挖掘器（Intelligent Miner）中聚类算法的使用演示。Arabie 和 Hubert[4]的论文是一篇很好的关于聚类方面的参考文献。

传统的聚类算法大致可以分为划分聚类方法[5,6]、层次聚类方法[7-9]、密度聚类方法[10-12]、网格聚类方法[13,14]、模型聚类方法[15]等。近年来，量子聚类方法[16]、谱聚类方法[17,18]、粒度聚类方法、概率图聚类方法、同步聚类方法[19-23]等也流行起来。

2.1 基于划分的聚类算法

基于划分的聚类算法通过构造一个迭代过程来优化目标函数，当优化到目标函数的最小值

或极小值时，可以得到数据集的一些不相交的子集，通常认为此时得到的每个子集就是一个聚类。

多数基于划分的聚类算法都是非常高效的，但需要事先给定一个在聚类分析前难以确定下来的聚类数目。*k*-means 算法[5]和 FCM(Fuzzy C Means)算法[6]是该类型中最著名的两个算法。另外，*k*-中心点算法 PAM(Partitioning Around Medoid)和 CLARA(Clustering LARge Applications)[24]、*k*-模算法(聚类分类数据)和 *k*-原型算法(聚类混合数据)[25]也属于这种类型的算法。

基于划分的聚类算法易于实现，聚类速度快，其时间复杂度与数据点数目 n、数据的维度 d 及预先设定的聚类数目 k 成线性关系。例如，*k*-means 算法和 FCM 算法的时间复杂度为 $T=O(ndkt)$，其中 t 为算法的迭代次数。由于 *k*-means 算法和 FCM 算法的目标函数优化是个 NP(Non-deterministic Polynomial)难问题，要搜索到最小值，所花费的时间代价非常高。采用迭代重定位的目标函数优化方法很容易陷入到局部极小值。对于有些数据集，即使优化到目标函数的全局最小值，此时对应的聚类簇也未必与数据集的实际分布结构相吻合。这类算法通常适合那些具有近似(超)球体形状、簇半径相近的数据集；而对于极不规则、大小相差很大的数据集，基于划分的聚类算法显得力不从心。因为这类算法具有这样的优点和缺点，所以自 *k*-means 算法和 FCM 算法发表以后，一直有大量的研究人员从事这方面的理论改进及扩展研究。例如，对于一些局部分布稀疏不均、聚类区域的形状及大小很不规整的数据集，*k*-means 和 FCM 算法常常不能很好地探测出其聚类分布结构。为克服 *k*-means 算法和 FCM 算法与初始值有关的两个重大缺陷(聚类数目 k 的确定、初始中心点集的选择)，许多研究者进行了更为深入的研究。

在聚类中心点集的初始化方向上，Arthur 和 Vassilvitskii[26]提出了复杂的 *k*-means++改进算法，但实际上，该算法的改进效果并不是十分明显；Zalik[27]提出的算法较好地解决了 k 值和初始聚类中心的选择问题，有较好的应用参考价值；Cao 等[28]提出了一种利用数据点的邻居信息来确定初始聚类中心的方法，也具有一定的参考价值。在确定合适的聚类数目方向上，许多研究者在聚类有效性函数的构造方面开展了研究，取得的成果有：Hubert 和 Arabie[29]提出的基于对象三元组的指标度量、Davies 和 Bouldin[30]提出的基于簇相似性的度量指数、Dunn[31]提出的 Dunn 指数、Bezdek 和 Pal[32]提出的 Dunn 推广指数等。

2.2　层次聚类算法

层次聚类方法[33]使用一个距离矩阵作为输入，经过聚类后得到一个反映该数据集分布状况的聚类层次结构图，其时间复杂度至少为 $T=O(n^2\log n)$。

层次聚类算法通常分为两种。第一种是凝聚的层次聚类算法，它首先把每个数据点看作是一个聚类，然后以一种自底向上的方式通过不断地选择最近邻居聚类对的合并操作，最终可以构造出一棵代表着该数据集聚类结构的层次树。第二种是分裂的层次聚类算法，它首先把所有的数据点看作是一个聚类，然后以一种以自顶向下的方式通过不断地选择最松散簇进行分裂操作，最终可以构造出一棵代表着该数据集聚类结构的层次树。

早期的层次聚类方法有 Kaufman 和 Rousseeuw[24]提出的 AGNES(AGglomerative NESting，凝聚的嵌套)聚类算法和 DIANA(DIvisive ANAlysis，分裂的分解)聚类算法。后来，Zhang 等[9]提出的 BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies)算法是一个更为著名的改进的凝聚型层次聚类算法，它采用 CF(Cluster Feature)树来进行层次聚类以达到改进聚类质量的目的。用 CF 树来描述聚类结构的方法后来在数据流聚类中又得到进一

步的发展和应用。

Guha 等提出的 CURE（Clustering Using REpresentative）算法[7]和 ROCK（RObust Clustering using linKs）算法[34]及 Karypis 等[8]提出的变色龙（CHAMELEON）算法也是三个有名的层次聚类算法。其中，CURE 算法是为解决海量数据集的聚类问题而开发出来的一种高效聚类方法，它使用多个点来表示一个簇，可以较好地过滤孤立点，解决非球状簇、簇大小不等的数据集的聚类问题。在处理大数据集时，CURE 算法采用随机抽样和划分分区的方法，因此可以获得较好的时间效率。ROCK 算法是一种健壮的用于类别属性数据集的凝聚型聚类算法。ROCK 算法也采用随机抽样技术，在计算两个数据点的相似度时，考虑它们共同邻居的数量，因此可以获得较好的健壮性。变色龙算法利用动态建模技术，首先从数据集中构造出一个 k-最近邻图 G_k，然后使用图划分算法将图 G_k 划分成大量的子图，每个子图代表一个初始子簇，最后用一种凝聚型的层次聚类算法反复合并子簇，找到真正的结果簇。

尽管层次聚类方法的时间代价高于划分聚类方法，但多数层次聚类算法并不需要事先设定一个难以确定的聚类数目这个参数，而且这类方法可以获得一种具有多个粒度的多层次聚类结构，这是它区别于划分聚类方法的最大优点。

2.3 基于密度的聚类算法

基于划分的聚类算法通常更适合于发现凸形聚类簇，但对于任意形状的聚类簇，它就显得有些力不从心了。基于密度的聚类算法试图通过稀疏区域来划分高密度区域以发现明显的聚类和孤立点，主要用于空间型数据的聚类。DBSCAN（Density-Based Spatial Clustering of Applications with Noise）算法[10]就是一个最为著名的基于密度的聚类算法。在该算法中，“密度可达性”被用来连接一些在某个距离范围内满足一定密度阈值的数据点。可见，DBSCAN 算法需要两个参数：距离参数和密度阈值参数。DBSCAN 算法不采用空间索引结构时，其时间复杂度为 $T=O(dn^2)$。如果使用合适的空间索引结构，如 R^* 树、k-d 树等，在低维空间中可以获得 $T=O(dn\log n)$ 的时间代价。OPTICS（Ordering Points to Identify the Clustering Structure）算法[11]是 DBSCAN 算法的一个推广，据说能比 DBSCAN 算法更好地处理不同密度的数据集。EnDBSCAN 算法[12]则是 DBSCAN 算法在效率上的一个变种。

2.4 基于网格的聚类算法

基于网格的聚类算法是一种基于网格的具有多分辨率的聚类方法。它首先将数据集的分布空间划分为若干个规则网格（如超矩形单元）或灵活的网格（如任意形状的多面体），然后通过融合相连的带数据概要信息的网格来获得明显的聚类。显然，几乎所有的基于网格的聚类算法都属于近似算法，它们能处理海量数据。这类算法的优点是处理时间与数据点的数目无关、与数据的输入顺序无关，可以处理任意类型的数据。其缺点是处理时间与每个维度上所划分的单元数相关，一定程度上降低了聚类的质量和准确性。

STING（STatistical INformation Grid）算法[14]是基于网格的聚类算法的典型代表。该算法利用属性空间的多维网格数据结构，将空间划分为多个单元网格。它针对不同级别的分辨率，存储多个级别的单元网格，这些单元网格形成一个层次结构，每个高层的单元网格被划分为多个低一层的单元网格。CLIQUE（CLustering In QUEst）算法[13]结合了网格和密度聚类的思想，它能聚类大规模高维数据。

2.5 基于模型的聚类算法

基于模型的聚类算法借助于一些统计模型来获得数据集的聚类分布信息。该方法假定数据集是由有限个概率分布模型共同作用生成的。在这种方法中，多变量的高斯分布混合模型应

用最为广泛。其中，Fish[35]提出的 COBWEB、Gennarim 等[36]提出的 CLASSI、 Cheeseman 和 Stutz[37]提出的 AutoClass 是较为有名的几个模型聚类方法。其中，COBWEB 算法是一个常用的、简单的增量式概念聚类方法，它采用分类树的形式来表现层次聚类结果。CLASSI 算法是一种基于增量概念模型的聚类方法，它以概率混合模型为基础，集成了认知和学习过程，利用属性的概率分布来描述聚类，该方法能很好地处理混合型数据。AutoClass 算法是一种基于经典的混和模型的聚类方法，该算法使用贝叶斯方法来确定最优类别，因此具有良好的数学理论基础。

在实际应用中，有时使用基于模型的聚类算法或其他聚类算法来获取数据集的聚类中心点集，然后再用学习向量化方法来构造分类器。

2.6　基于图的聚类算法

采用图聚类方法[15]进行聚类分析时，首先是建立与具体问题相适应的图。图的结点代表被分析数据的基层单元，图的边代表基层单元数据之间的相似性度量(或相异性度量)。通常，每个基层单元数据之间都会有一个度量表达，这样可以保持数据集的局部分布特性。图聚类方法是以数据集的局部连接特征作为聚类的主要信息源，因而易于处理局部数据的特性。

Karypis 等[8]提出的变色龙算法也可看作是一种图聚类算法。

2.7　其他聚类算法

除了以上提及的传统聚类算法外，近些年也发展起来不少新的聚类算法，具体如下。

量子聚类方法借用了量子学理论，先是从源数据中创建一个基于空间尺度的概率函数，接着使用一些分析操作来获得一个根据极小值来确定聚类中心的势函数，最终通过调整尺度参数来搜索聚类结构。

谱聚类(Spectral Clustering)方法通过源数据的相似度矩阵来计算特征值，进而可以发现明显的聚类区域。许多谱聚类算法都易于实现，其效果优于传统的聚类算法，如 *k*-means，它们在许多应用中也都获得了成功的实现。用于图像划分的 Shi-Malik 算法[38]就是基于谱聚类方法开发出来的。

基于粒度的聚类方法[39]，是从信息粒度角度上发展起来的一个新的聚类研究方向。目前这种聚类方法的研究还不太成熟，尤其是对粒度计算语义的研究还比较少。

概率图聚类方法是近年来流行起来的一种聚类方法。最著名的概率图聚类方法要数 2007 年发表在《Science》上面的 AP(Affinity Propagation)聚类算法[40]。

2010年，Böhm 等[19]受大自然普遍存在的同步原理启发，提出了一种新颖的同步聚类算法——SynC (Synchronization Clustering)算法。该算法不仅可以在不知道数据集的任何分布情况下，通过动态的同步过程来发现它的内在结构，并能很好地处理孤立点，还能使用最小描述长度原理来实现自动聚类。

3　聚类算法的研究现状及发展趋势

十年来，出现了几篇原创性的聚类算法论文。例如，Frey 和 Dueck[40]于 2007 年发表在《Science》上的 AP 聚类算法，引领了新型的基于概率图模型的聚类算法研究方向。据我们所知，Böhm 等[19]第一次将自然界中普通存在的同步现象引入聚类领域，在 KDD2010 上发表了第一篇同步聚类算法论文。这篇开拓性论文首先将 Kuramoto 模型进行了适当地推广，得到可应用于聚类算法中的扩展 Kuramoto 模型，提出了 SynC 聚类算法。同时该文还将最小描述长度原理应用于 SynC 算法中，提出了一种自动优化参数的方法。基于同步模型的聚类算法可以缓解聚类分析和噪声检测在传统数据上的某些难题，具

有动态性、局部性及多尺度分析等特性，可以在一定程度上解决大规模数据的聚类分析所面临的困难。自此之后，新型的基于同步模型的聚类算法形成了一个研究热潮。Rodriguez 和 Laio[41]于 2014 年发表在《Science》上的 DP（Density Peak）聚类算法，为聚类算法的设计提供了一种新的思路，估计能引领一种新的聚类算法研究方向。

除了这些原创聚类算法外，大量发表的是一些改进型聚类算法论文、多种技术结合型聚类算法论文及聚类应用论文。例如，Bouguettaya 等[42]利用一组相连的数据点子集的中心点来建立一个层级结构，提出了一种高效的凝聚的层次聚类算法。Garcia 等[43]将 *k*-means 算法扩展后，应用到功能型数据的聚类分析中，提出了一种适合于功能数据的 *kk*-means 聚类算法。Ozturk 等[44]提出了一种基于二元人工蜜蜂群体算法的动态聚类算法。这是一个将群体智能方法应用到聚类算法中的典型例子，该算法不仅能自动确定最优的聚类数目，而且还可以获得较好的聚类质量。Ghassabeh[45]通过引入 Lyapunov 函数，发现 Mean Shift 算法的平衡点是渐近稳定的。这意味着在 Mean Shift 算法中，从一个平衡点的近邻区域内任一点开始迭代，所产生的序列最终收敛到该平衡点，该项成果推进了 Mean Shift 算法在理论上的收敛性分析研究。Kolesnikov 等[46]提出了一种基于量子纠错的参数建模方法来确定最优聚类数目的方法，该方法目前主要应用于数值型数据集中。Villalba 等[47]比较分析了图像领域中的聚类技术，为实时图像处理软件的开发提供了参考方法。Ritter 等[48]提出了一个基于统计的简易近邻聚类算法，该算法能够消除背景噪声、孤立点，并且能够从一些数据集中检测出具有不同密度的聚类区域。

自 Böhm 等[19]在 KDD2010 上发表了基于 Kuramoto 同步模型的聚类算法后，吸引了一些研究人员的注意。之后，一些研究人员从不同视角、不同应用领域发表了多篇同步聚类方面的论文。例如，邵俊明等从多个角度研究基于同步模型的数据挖掘方法，发表了多篇高水平论文[21-23,49,50]：为了更自然地从真实复杂数据集中检测出孤立点，从同步遗漏角度提出了一种新颖的孤立点检测算法[49]；为了发现高维稀释数据集在子空间中的多个聚类簇，提出了一种新颖有效的子空间聚类算法，即 ORSC 算法[50]；为了从层次聚类结构中探测出有价值的层次信息，提出了一种基于同步原理和最小描述长度原理的新颖的动态层次聚类算法，即 hSync 算法[21]；为了更好地从数据流中提取有价值的信息，提出了一种用于检测概念漂移的基于原型学习的数据流挖掘算法[22]。2013 年，黄健斌等[20]在 Böhm 等[19]基础上，提出了一种基于同步的层次聚类方法。2017 年，陈新泉[51]在 Böhm 等[19]基础上，提出了一种基于 Vicsek 模型的线性版本的同步聚类算法；Hang 等[52]基于重力学中的中心力优化方法，提出了一种局部同步聚类算法（G-Sync 算法）。这两篇较新的同步聚类论文分别从不同的角度扩展了原始的同步聚类算法研究。

4 近邻思想及同步模型在聚类分析中的应用

在数据挖掘领域，近邻思想在提高算法的时间效率方面发挥了很大的作用。

许多基于图的聚类方法，因需要获知并存储任意两点的相异性度量，所以其时间和空间复杂度往往会达到 $O(n^2)$，这样的时空代价不太适合海量数据集的聚类分析。为改善这类算法的性能，根据由多个相邻的局部可以近似构造全局的原则，可以认为，某个空间区域内的数据集的聚类分析并非一定要在全局范围内进行任意两个数据点之间相异性度量的计算及存储。事实上，全局的聚类分析也可通过局部区

域范围内数据点之间的相异性度量计算，进而构造出其局部拓扑结构，由多个相邻的局部拓扑结构来近似推知全局范围内数据集的空间分布结构，最终得到数据集的多层次聚类分布状况描述，该观点是某些改进型聚类算法的一个核心思想。例如，ICNNI(Improved Clustering based on Near Neighbor Influence)算法[53]、FSynC(Fast Synchronization Clustering)算法、IESynC(Improved Effective Synchronization Clustering)算法[51]和 ISSynC(Improved Shrinking Synchronization Clustering)算法等就是一类通过寻找并设计出合适的数据结构而得到的能改善时间代价的有效算法。

尽管目前在基于同步模型的聚类方法方面已出现了一些研究成果，但该方法由于自身具有的平方时间复杂度，目前还不能推广应用到面向大数据的聚类分析中。Chen[54]展示了如何通过近邻思想和方法的运用来降低聚类算法的时间代价。这种基于近邻的思想依旧可以应用到基于同步模型的聚类方法中。因此，我们认为这个问题具有一定的研究意义。

5　结束语

到目前为此，尽管在大数据存储、大数据搜索、大数据挖掘等方面已取得了一些研究成果，但当前的大数据挖掘理论、方法及技术水平仍满足不了大数据时代的需要。当前，研究实时、高鲁棒的面向大数据的新型高效聚类算法，已成为深入挖掘大数据中的隐含价值需要解决的一项关键任务。

在数据挖掘领域，许多聚类算法的最终结果敏感于参数的正确设定，该缺陷导致这些算法远不能称为成熟的、实用的智能型机器学习算法。在大数据环境下的搜索引擎软件设计中，需要开发出更为高效的智能型自动聚类算法，所以建立一套聚类算法的参数自适应优化理论及方法仍是一项紧迫的任务。

同步聚类算法的理论也还有待进一步研究。例如，为什么同步机制能应用于聚类分析中？这里面是否存在着一种深层次的理论基础？这些问题都有待深入研究。

参 考 文 献

[1] Agarwal PK, Guibas LJ, Edelsbrunner H. Data clustering: a review [J]. ACM Computing Surveys, 1999, 31(3): 264-323.

[2] Qian WN, Zhou AY. Analyzing popular clustering algorithms from different viewpoints [J]. Journal of Software, 2002, 32(5): 432-445.

[3] Grabmeier J, Rudolph A. Techniques of cluster algorithms in data mining [J]. Data Mining and Knowledge Discovery, 2002, 6(4): 303-360.

[4] Arabie P, Hubert LJ. An overview of combinatorial data analysis [M] // Clustering and Classifications. World Scientific Publishing Co Pte Ltd, 2003: 5-63.

[5] Bezdek JC. Pattern Recognition with Fuzzy Objective Function Algorithms [M]. New York: Plenum Press, 1981.

[6] Macqueen JB. Some methods for classification and analysis of multivariate observations [C] // Proceedings of Berkeley Symp on Mathematical Statistics and Probability, 1967: 281-297.

[7] Guha S, Rastogi R, Shim K. CURE: an efficient clustering algorithm for clustering large databases [C] // ACM SIGMOD International Conference on Management of Data, 1998: 73-84.

[8] Karypis G, Han EH, Kumar V. CHAMELEON: a hierarchical clustering algorithm using dynamic modeling [J]. Computer, 1999, 32(8): 68-75.

[9] Zhang T, Ramakrishnan R, Livny M. BIRCH: an efficient data clustering method for very large databases [C] // Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data, 1996: 103-114.

[10] Ester M, Kriegel HP, Sander J, et al. A density-based algorithm for discovering clusters in large

spatial data sets with noise [C] // International Conference on Knowledge Discovery and Data Mining, 1996: 226-231.

[11] Ankerst M, Breunig MM, Kriegel HP, et al. OPTICS: ordering points to identify the clustering structure [J]. ACM Sigmod Record, 1999, 28(2): 49-60.

[12] Roy S, Bhattacharyya DK. An approach to find embedded clusters using density based techniques [C] // ICDCIT 2005: Distributed Computing and Internet Technology, 2005: 523-535.

[13] Agrawal R, Gehrke J, Gunopolos D, et al. Automatic subspace clustering of high dimensional data for data mining application [C] // Proceedings of the 1998 ACM SIGMOD International Conference on Mangement of Data, 1998: 94-105.

[14] Wang W, Yang J, Muntz RR. STING: a statistical information grid approach to spatial data mining [C] // Proceedings of International Conference on Very Large Data Bases, 1997: 186-195.

[15] Theodoridis S, Koutroumbas K. Pattern Recognition (The Fourth Edition) [M]. Pittsburgh: Academic Press, 2008.

[16] Horn D, Gottlieb A. Algorithm for data clustering in pattern recognition problems based on quantum mechanics [J]. Physical Review Letters, 2002, 88(1): 018702.

[17] Luxburg UV. A tutorial on spectral clustering [J]. Statistics and Computing, 2007, 17(4): 395-416.

[18] Schölkopf B, Smola A, Müller KR. Nonlinear component analysis as a kernel eigenvalue problem [J]. Neural Computation, 1998, 10(10): 1299-1319.

[19] Böhm C, Plant C, Shao J. et al. Clustering by synchronization [C] // ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2010: 583-592.

[20] Huang JB, Kang JM, Qi JJ, et al. A hierarchical clustering method based on a dynamic synchronization model [J]. Science in China Series F: Information Sciences, 2013, 43(5): 599-610.

[21] Shao J, He X, Böhm C, et al. Synchronization inspired partitioning and hierarchical clustering [J]. IEEE Transactions on Knowledge & Data Engineering, 2013, 25(4): 893-905.

[22] Shao J, He X, Plant C, et al. Robust synchronization-based graph clustering [M] // Advances in Knowledge Discovery and Data Mining. Springer Berlin Heidelberg, 2013: 249-260.

[23] Shao J, Ahmadi Z, Kramer S. Prototype-based learning on concept-drifting data streams [C] // ACM SIGKDD, 2014: 412-421.

[24] Kaufman L, Rousseeuw PJ. Finding Groups in Data: an Introduction to Cluster Analysis [M]. New York: John Wiley & Sons, 1990.

[25] Huang Z. Extensions to the *k*-means algorithm for clustering large data sets with categorical values [J]. Data Mining and Knowledge Discovery, 1998, 2(3): 283-304.

[26] Arthur D, Vassilvitskii S. *K*-means++: the advantages of careful seeding [C] // Eighteenth ACM-SIAM Symposium on Discrete Algorithms, 2007: 1027-1035.

[27] Zalik KR. An efficient *k*-means clustering algorithm [J]. Pattern Recognition Letters, 2008, 29(9): 1385-1391.

[28] Cao F, Liang J, Jiang G. An initialization method for the *k*-means algorithm using neighborhood model [J]. Computers & Mathematics with Applications, 2009, 58(3): 474-483.

[29] Hubert LJ, Arabie P. Comparing partitions [J]. Journal of Classification, 1985, 2(1): 193-218.

[30] Davies DL, Bouldin DW. A cluster separation measure [J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 1979, 1(2): 224-227.

[31] Dunn JC. A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters [J]. Journal of Cybernetics, 1973, 3(3): 32-57.

[32] Bezdek JC, Pal NR. Some new indexes of cluster validity [J]. IEEE Transactions on Systems, Man & Cybernetics, 1998, 28(3): 301-315.

[33] Johnson SC. Hierarchical clustering schemes [J]. Psychometrika, 1967, 32(2): 241-254.

[34] Guha S, Rastogi R, Shim K. ROCK: a robust clustering algorithm for categorical attributes [J]. Information Systems, 1999, 25(5): 512-521.

[35] Fisher D. Improving inference through conceptual

clustering [C] // National Conference on Artificial Intelligence, 1987: 461-465.

[36] Gennari JH, Langley P, Fisher D. Models of incremental concept formation [J]. Artificial Intelligence, 1989, 40(1-3): 11-61.

[37] Cheeseman P, Stutz J. Bayesian classification (AutoClass): theory and results [M] // Advances in Knowledge Discovery & Data Mining. The Association for Computing Machinery, 1997: 153-180.

[38] Shi J, Malik J. Normalized cuts and image segmentation [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2002, 22(8): 888-905.

[39] 王国胤, 张清华, 胡军. 粒计算研究综述 [J]. 智能系统学报, 2007, 2(6): 8-26.

[40] Frey BJ, Dueck D. Clustering by passing messages between data points [J]. Science, 2007, 315(16): 972-976.

[41] Rodriguez A, Laio A. Clustering by fast search and find of density peaks [J]. Science, 2014, 344(6191): 1492-1496.

[42] Bouguettaya A, Yu Q, Liu XM, et al. Efficient agglomerative hierarchical clustering [J]. Expert Systems with Applications, 2015, 42(5): 2785-2797.

[43] Garcia MLL, Garcia-Rodenas R, Gomez AG. *K*-means algorithms for functional data [J]. Neuro Computing, 2015, 151: 231-245.

[44] Ozturk C, Hancer E, Karaboga D. Dynamic Clustering with improved binary artificial bee colony algorithm [J]. Applied Soft Computing, 2015, 28: 69-80.

[45] Ghassabeh YA. Asymptotic stability of equilibrium points of mean shift algorithm [J]. Machine Learning, 2015, 98(3): 359-368.

[46] Kolesnikov A, Trichina E, Kauranne T. Estimating the number of clusters in a numerical data set via quantization error modeling [J]. Pattern Recognition, 2015, 48(3): 941-952.

[47] Villalba LJG, Orozco ALS, Corripio JR. Smartphone image clustering [J]. Expert Systems with Applications, 2015, 42(4): 1927-1940.

[48] Ritter GX, Nieves-Vazquez JA, Urcid G. A simple statistics-based nearest neighbor cluster detection algorithm [J]. Pattern Recognition, 2015, 48(3): 918-932.

[49] Shao J, Böhm C, Yang Q, et al. Synchronization based outlier detection [M] // Machine Learning and Knowledge Discovery in Databases. Springer Berlin Heidelberg, 2010: 245-260.

[50] Shao J, Yang Q, Böhm C, et al. Detection of arbitrarily oriented synchronized clusters in high-dimensional data [C] // IEEE International Conference on Data Mining, 2010: 607-616.

[51] Chen XQ. An effective synchronization clustering algorithm [J]. Applied Intelligence, 2016, 46(1): 1-23.

[52] Hang W, Choi KS, Wang S. Synchronization clustering based on central force optimization and its extension for large-scale datasets [J]. Knowledge-Based Systems, 2017, 118: 31-44.

[53] Chen XQ. A new clustering algorithm based on near neighbor influence [J]. Expert Systems with Applications, 2014, 42(21): 7746-7758.

[54] Chen XQ. Clustering based on a near neighbor graph and a grid cell graph [J]. Journal of Intelligent Information Systems, 2013, 40(3): 529-554.