第 43 卷 第 6A 期
2016 年 6 月

计 算 机 科 学
Computer Science

Vol. 43 No. 6A
June 2016

大数据聚类算法综述

海 沫
(中央财经大学信息学院 北京 100081)

摘 要 随着数据量的迅速增加,如何对大规模数据进行有效的聚类成为挑战性的研究课题。面向大数据的聚类算法对传统金融行业的股票投资分析、互联网金融行业中的客户细分等金融应用领域具有重要价值。对已有的大数据聚类算法进行了详细划分,并比较了每种聚类算法的优缺点,进一步总结了已有研究存在的问题,最后对未来的研究方向进行了展望。

关键词 大数据,聚类算法,股票投资分析,客户细分
中图法分类号 TP311 文献标识码 A

Survey of Clustering Algorithms for Big Data

HAI Mo
(School of Information, Central University of Finance and Economics, Beijing 100081, China)

Abstract With the rapid increase of data size, it is a challenge to cluster the large scale data. Clustering algorithms for big data are very important for the stock investment analysis in the traditional finance field, customer segmentation in Internet finance field and so on. Firstly, the existing clustering algorithms for big data were divided, and then the advantages and disadvantages of each type were compared. After that, the problems of the existing researches were summarized. Finally, the future research directions were given.

Keywords Big data, Clustering algorithms, Stock investment analysis, Customer segmentation

1 介绍

大数据是指在可容忍的时间内无法用现有的信息技术和软硬件工具对其进行传输、存储、计算和应用等的数据集合。此外,大数据又被引申为解决问题的方法,即通过收集、分析海量数据获得有价值的信息,并通过实验、算法和模型发现规律,形成新的商业模式[1]。随着金融业务和服务的多样化、金融市场整体规模的扩大、互联网金融的飞速发展,金融行业的数据收集能力逐步提高,存储了大量时间连续、动态变化的数据。与其他行业相比,大数据对金融业更具潜在价值,金融业正面临着前所未有的科技挑战。面对激增的海量数据,如何实现分析和洞察,将是金融行业创新和转型的关键。面向大数据的聚类算法对传统金融行业的股票投资分析、互联网金融行业中的客户细分等金融应用领域具有重要应用价值。本文将已有的大数据聚类算法的研究划分成两种:单机聚类算法和多机聚类算法,并对每一类型进行了详细划分,同时比较了每种聚类算法的优缺点。

2 大数据聚类算法

大数据聚类算法可分为两类:单机聚类算法和多机聚类算法,其分类如图 1 所示。

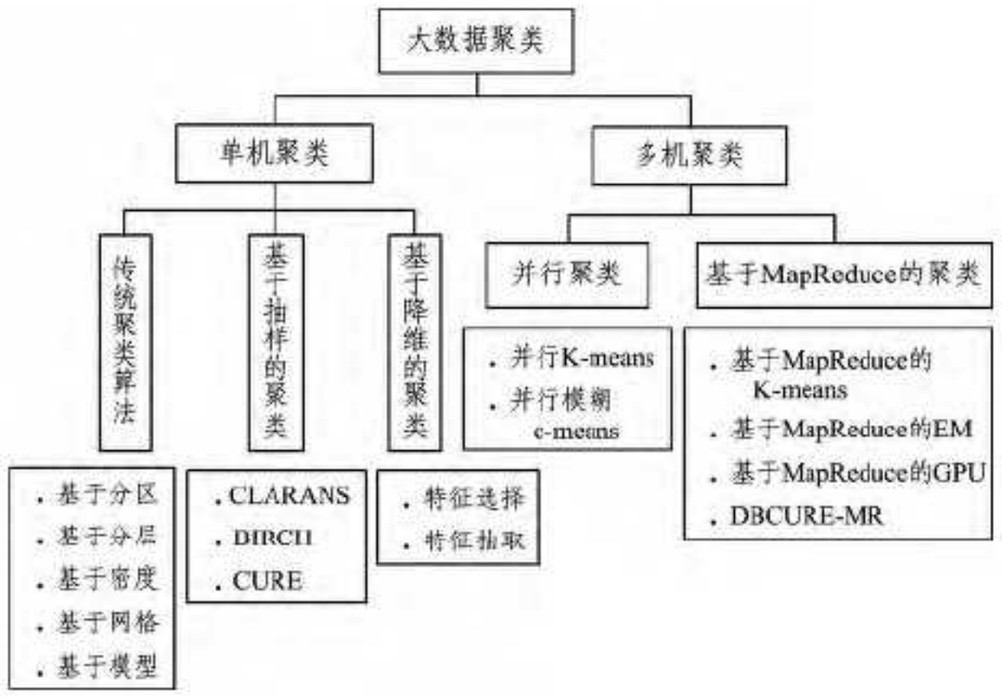

图 1 大数据聚类算法的分类

2.1 单机聚类算法

(1)传统聚类算法

传统的聚类算法可分为:分区聚类、层次聚类、基于密度的聚类、基于模型的聚类和基于网格的聚类[2-12]。

a)分区聚类算法:该类算法基于点的相似性在单个分区中基于距离来划分数据集。其缺点是需要用户预定义一个参数 k,而它通常具有不确定性。代表性的分区算法包括:K-means、K-methods、K-modes、PAM、CLARA、CLARANS 和 FCM。

本文受北京高等学校青年英才计划项目(YETP0988)资助。
海 沫(1978—),女,博士,副教授,CCF 高级会员,主要研究方向为分布式系统、大数据分析和处理,E-mail:haimo-hm@163.com。

b)层次聚类算法:该类算法将数据划分成不同的层次,并提供了可视化。其基于相似性或距离将数据自底向上或自顶向下进行分层划分,其划分结果表示为一种层次分类树。它的主要缺点是:一旦完成了某个划分阶段,就无法撤销。其代表性算法有:BIRCH、CURE、ROCK 和 Chameleon。

c)基于密度的聚类算法:该类算法能够以任意一种方式发现簇。簇定义为由低密度区域分开的密集区域。基于密度的聚类算法不适用于大型的数据集。其代表性算法包括:DBSCAN、OPTICAL DBCLASD 和 DENCLUE,它们常用来过滤噪音。

d)基于模型的聚类算法:该类算法基于多元概率分布规律,可以测量划分的不确定性,其中,每个混合物代表一个不同的簇。该类算法对大数据集的处理很慢。该类算法的代表性算法有 EM、COBWEB、CLASSIT 和 SOM。

e)基于网格的聚类算法:该类算法的过程分为 3 个阶段,首先,将空间划分为矩形方格以获取一个具有相同大小的方格的网格;然后,删除低密度的方格;最后,将相邻的高密度的方格进行结合以构成簇。该类算法最明显的优点在于其复杂度显著减少。其代表性的算法有:CRIDCLUS、STING、OptiGrid、CLICK 和 WaveCluster。

表 1 从处理的数据规模、聚类质量、可扩展性和稳定性 4 个方面比较了以上 5 类传统聚类算法中的代表性算法。

表 1 传统聚类算法的比较

聚类算法	处理的数据规模	聚类质量	可扩展性	稳定性
FCM(分区聚类算法)	大	高	低	低
BRICH(层次聚类算法)	超大	低	高	中等
DENCLUE(基于密度的聚类算法)	超大	低	高	中等
EM(基于模型的聚类算法)	大	高	低	高
OptiGrid(基于网格的聚类算法)	超大	低	高	中等

从表 1 的比较结果可发现不存在 4 个方面表现都好的聚类算法。相对于其他算法,EM 和 FCM 具有更高的聚类质量。BIRCH、DENCLUE 和 OptiGrid 能处理的数据规模更大,但它们的聚类质量更低。聚类算法的主要缺点是它们的不稳定性,即多次运行同一算法会得到不同的结果。

(2)基于抽样的聚类算法

该类算法不同于原来基于整个数据集的聚类算法,它只需要在数据集的一个样本上应用聚类算法,就可以推广到整个数据集。因为它只需要对更小的数据集进行聚类,其所需的聚类时间大幅减少、存储空间大幅度减小。

a)CLARANS

CLARANS[13] 的前身为 CLARA[14]。与围绕中心点的聚类算法 PAM[14] 相比,CLARA 能够处理更大规模的数据,并降低了算法复杂性和时间开销,其运行时间与数据对象的个数呈线性关系。PAM 计算了所有数据对象两两之间的相异矩阵并存储在内存中,需要 $O(n^2)$ 的存储空间,因此 PAM 不能用于 n 较大的情况。为了解决这个问题,CLARA 没有一次性地计算整个相异矩阵。PAM 和 CLARA 可被看作概念上的图形搜索问题,其中的每个节点都是一个可能的聚类方法。PAM 从一个随机选择的节点开始,一直移动到不能找到一个更好的邻居节点。CLARA 只搜索由抽样得到的数据点构成的子图,从而缩小了搜索空间。

CLARANS 算法提高了 CLARA 算法的效率。与 PAM 一样,CLARANS 的目标是通过搜索整个图找到一个局部最优解。不同的是,在每次迭代中,它只检查当前节点邻居节点的样本节点。显而易见,CLARANS 和 CLARA 都是应用抽样技术来缩小搜索空间,其不同点在于抽样方式。CLARA 的抽样在开始阶段就已经完成,它将整个搜索过程限制在一个特定的子图中;而 CLARANS 则是针对搜索过程的每次迭代进行抽样。观察结果表明:用于 CLARANS 的动态抽样方法比用于 CLARA 的方法更高效[15]。

b)BIRCH

如果数据大小大于内存大小,I/O 开销将决定聚类时间。BIRCH[16] 为这个问题提供了一个其他算法都没有提出的解决方案。BIRCH 运用自身数据结构,其被称为聚类特征(Clustering Feature,CF),也叫做聚类特征树(CF Tree)。聚类特征是每个聚类的摘要。对聚类而言,每个数据点并不是同等重要,所以不是所有的数据点都需要放到内存中。

聚类特征可用三元组 $\langle N, LS, SS\rangle$ 表示。其中,N 表示聚类中的数据点的总数,LS 表示聚类中数据点的线性和,SS 表示聚类中数据点的平方和。如果两个聚类合并后的聚类的 CF 是两个原始聚类的 CF 之和,则可得出 CF 满足加法性质的结论。该性质的重要性在于它使得在无需访问原始数据集的情况下合并两个聚类。

BIRCH 算法有两个关键阶段。首先,它扫描数据点并在内存中建立一棵树;然后,应用聚类算法对叶子节点进行聚类。文献[17]中的实验结果表明:在时间开销和空间开销两方面,BIRCH 要优于 CLARANS。在处理异常值方面,BIRCH 也要优于 CLARANS。图 2 为 BIRCH 算法的流程图。

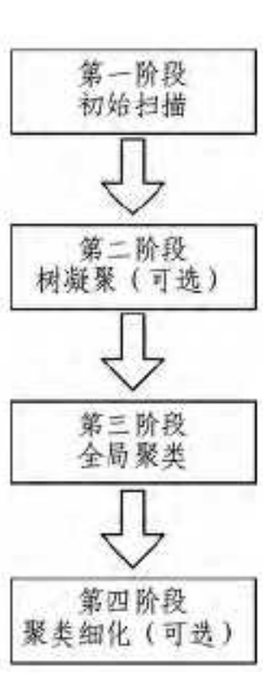

图 2 BIRCH 流程图

c)CURE

前面的算法都用单个数据点表示一个聚类,其适用于球形聚类,但实际应用中聚类可能有各种不同的复杂形状。为了应对这一挑战,CURE[18] 使用一组分散的数据点来代表这个聚类。CURE 实际上是一个层次算法,它把每一个数据点都看作一个单独的聚类,并且一直合并两个已有的聚类,直到最后的聚类个数为 k。每个阶段选择两个需合并的聚类的过程都是建立在计算这两个聚类的代表点的所有可能的节点对的最短距离的基础之上。两个主要的数据结构使 CURE 能够有效地搜索。第一个是堆,用来记录现有的聚类与距离其最短的聚类间的距离;另一个是 k-d 树,用来存储每个聚类的所有代表点。

CURE 也使用抽样技术来加速计算。它选取一个输入数

据集的样本,并在样本上执行之前提到的运算过程。为了清楚阐明需要的样本大小,Chernoff 界限用在原始研究中。如果数据集非常大,即抽取样本后数据集仍然很大,这个过程将很耗时。为了解决这个问题,CURE 使用分区方式对算法进行加速。如果将 n 看作原始数据集,n' 看作样本数据集,CURE 会将 n' 分为 p 个分区,并且每个分区都会进行局部的分层聚类直到达到预设的聚类数目或者两个需合并的聚类之间的距离超过了某个阈值;接着,执行另一个聚类过程,并传递所有 p 个分区的所有局部聚类;最后,所有没被抽出的数据点将被分配到距其最近的聚类中。文献[18]中的实验结果显示:与 BIRCH 相比,CURE 的运行时间更短,并通过一个常数因子缩小代表点和聚类中心的距离以获得在处理异常值方面的健壮性优势。

(3)基于降维的聚类算法

数据大小的测量有两个维度:变量的数量和实例的数量。这两个维度都可能有很大的值,这可能导致在分析数据时出现问题。因而,有必要实现数据处理工具并且在应用聚类算法之前对数据集进行预处理,以获得对数据中蕴含的知识的更好理解。降维能够解决这一问题。降维的目的是基于一个事先定义的标准,选择或提取具有相关特性的最优子集。根据使用的标准,特征子集的选择可以消除无关和冗余的信息。这种选择或提取可以缩小样本空间,使其对该问题更具代表性。对于大规模数据集,降维通常用在聚类算法之前以避免高维度的缺点。

a)特征选择

特征选择根据某种性能标准从原始变量集合中选取变量的最优子集。文献[19]提出了一种基于特征选择面向大数据的聚类算法。首先,应用特征选择算法以减少数据集的大小;然后,在选出的子集中应用并行 k-means 算法。实验结果表明:相对于已有算法,该算法对大数据的聚类精度更高、聚类时间更短。

b)特征提取

特征提取是从转换后的空间——投影空间中选取特征。提取方法使用所有信息来压缩和产生一个更小维度的向量。文献[20]提出了一种基于 PCA 和 LS-SVM 的面向大数据的特征提取方法和聚类算法。实验结果表明:这种基于特征提取的聚类算法可以解决大数据的聚类问题。

2.2 多机聚类

(1)并行聚类

对大数据的处理需要并行计算从而在合理的时间内获取结果。并行聚类对数据进行划分并将其分布在不同的机器上,这使得单台机器聚类速度加快,并且增加了可扩展性。

文献[21]提出的并行 K-means 算法在 IBM SP2 POWER 上实现了 16 个节点的并行。文献[22]在以太网上用 32 台机器进一步实现了 K-means 算法的并行版本,其对大规模数据呈现出接近线性的加速比。文献[23,24]证明了并行 K-means 算法的可扩展性。

(2)基于 MapReduce 的聚类

MapReduce 是一种将任务分布在大量的服务器上分布执行的任务分区机制。Map 阶段将一个任务分解为更小的子任务并将这些子任务分配到不同服务器上执行;Reduce 阶段将子任务执行的结果进行合并。

文献[25]提出了一种加速 K-means 聚类算法的方法。文献[26]对 K-means 聚类算法的加速方法进行了改进。文献[27]提出了一种能给出更好近似的快速 K-means 算法。文献[28]提出了一种改进的 K-means 算法以解决大数据处理过程中的问题。该算法提出了一种新的 MapReduce 模型以消除对迭代的依赖,从而达到更高性能。文献[29]基于 MapReduce 改写了 EM 算法,使得每台计算机的内存只需加载部分数据。这种方法可以减少时间和内存开销。Younghoon 和 Kyuseok 开发了 DBCURE-MR[30]。这是一种基于 Map-Reduce 的并行 DBCURE 算法。传统的基于密度的算法逐个发现每个聚类,而 DBCURE-MR 能够并行发现多个聚类。实验结果表明:DBCURE-MR 能够有效地查找聚类。文献[31]显示了 GPU 和 CUDA 平台的动态并行特征如何给 BIRCH 带来好处。GPU 能够加速 BIRCH,并使其比具有高可扩展性和精确性的 CPU 快 154 倍。

表 2 比较了不同大数据聚类算法的优缺点。

表 2 大数据聚类算法

聚类算法	优点	缺点
数据挖掘聚类算法	·实现简单	·不能处理大量数据
基于抽样的聚类	·时间开销和空间开销小	·抽样的精确性影响聚类的精确性
基于降维的聚类	·减少数据集 ·优化处理开销 ·高效、可扩展	·没有为高维数据集提供有效的解决方案
并行聚类	·高效 ·可扩展性好	·算法不易实现
基于 MapReduce 的聚类	·高可扩展性	·需要消耗更多软硬件资源 ·难以基于 MapReduce 实现每个查询 ·对常用操作(选择/提取)没有提供原语

结束语 本文对已有的面向大数据的聚类算法进行了划分,并对每种类型的优缺点进行了比较。传统聚类算法需要进行抽样或降维才能应用到大数据的聚类中,但会损失聚类的精确性。尽管并行聚类对于大数据的聚类具有效率高、可扩展性好的优点,但这些算法实现的复杂性高,而基于 MapReduce 能够实现高可扩展的面向大数据的聚类算法,但需要消耗更多的软硬件资源。未来需要研究在不消耗更多软硬件资源的情况下简单、高效、可扩展和精确的面向大数据的聚类算法。

参考文献

[1] Manyika J,Chui M,Brown B,et al. Big data:The Next Frontier for Innovation,Competition,and Productivity[R]. McKinsey Global Institute,2011

[2] Fahad A,Alshatri N,Tari Z,et al. A Survey of Clustering Algorithms for Big Data:Taxonomy & Empirical Analysis[J]. IEEE Transactions on Emerging Topics in Computing,2014,2(3):1

[3] Ayed A B,Halima M B,Alimi M. Survey on clustering methods:Towards fuzzy clustering for Big Data[C]// 6th International Conference of Soft Computing and Pattern Recognition (SoCPaR). IEEE,2014:331-336

[4] Sherin A,Uma S,Saranya K,et al. Survey On Big Data Mining Platforms,Algorithms And Challenges[J]. International Journal of Computer Science & Engineering Technology,2014,5(9):854-862

[5] Arora S,Chana I. A survey of clustering techniques for Big Data analysis[C] // 5th International Conference Confluence The Next Generation Information Technology Summit (Confluence). IEEE,2014:59-65

[6] Nagpal P B,Mann P A. Survey of Density Based Clustering Algorithms[J]. International Journal of Computer Science and its Applications,2011,1(1):313-317

[7] Xu R,Wunsch D. Survey of clustering algorithms,Neural Networks[J]. IEEE Transactions,2005,16(3):645-678

[8] Yadav C,Wang S,Kumar M. Algorithm and approaches to handle large Data-A Survey[J]. Eprint Arxiv,2013,361-363(3):1117-1181

[9] Shirkhorshidi A S, Aghabozorgi S, Wah T Y, et al. Big Data Clustering:A Review[M] // Computational Science and Its Applications(ICCSA 2014). Springer International Publishing, 2014:707-720

[10] Wu X,Zhu X,Wu G Q,et al. Data mining with Big Data[J]. IEEE Transactions on Knowledge and Data Engineering,2014,26(1):97-107

[11] Aggarwal C C,Reddy C K. Data Classification:Algorithms and Applications[C] //CRC Press. 2014

[12] Vadgasiya M G, Jagani J M. An enhanced algorithm for improved cluster generation to remove outlier's ratio for large datasets in data mining[P]. Development,2014

[13] Ng R T,Han J. CLARANS:A method for clustering objects for spatial data mining[J]. IEEE Trans. Knowl. Data Eng.,2002,14(5):1003-1016

[14] Kaufman L,Rousseeuw P J. Finding Groups in Data:An Introduction on Cluster Analysis[M]. John Wiley and Sons,1990

[15] Ng R T,Han J. CLARANS:A method for clustering objects for spatial data mining[J]. IEEE Trans. Knowl. Data Eng.,2002,14(5):1003-1016

[16] Zhang T,Ramakrishnan R,Livny M. BIRCH:An efficient data clustering method for very large database[C] // SIGMOD Conference. 1996:103-114

[17] Zhang T,Ramakrishnan R,Livny M. BIRCH:An efficient data clustering method for very large database[C] // SIGMOD Conference. 1996:103-114

[18] Guha S,Rastogi R. CURE:An efficient clustering algorithm for large database[J]. Inf. Syst.,2001,26(1):35-58

[19] Bu F,Chen Z,Zhang Q,et al. Incomplete Big Data Clustering Algorithm Using Feature Selection and Partial Distance[C] // 5th International Conference on Digital Home (ICDH). IEEE,2014:263-266

[20] Kim B J. A Classifier for Big Data[M] //Convergence and Hybrid Information Technology. Springer Berlin Heidelberg,2012:505-512

[21] Dhillon I S,Modha D S. A data-clustering algorithm on distributed memory multiprocessors[M] // Large-Scale Parallel Data Mining. Springer Berlin Heidelberg. 2000:245-260

[22] Stoffel K,Belkoniene A. Parallel k/h-means clustering for large data sets[M] // Euro-Par'99 Parallel Processing. Springer Berlin Heidelberg,1999:1451-1454

[23] Nagesh H S,Goil S,Choudhary A. A scalable parallel subspace clustering algorithm for massive data sets[C] // International Conference on Parallel Processing,2000. IEEE,2000:477-484

[24] Ng M K, H Zhe-xue. A Parallel k-Prototypes Algorithm for Clustering Large Data Sets in Data Mining[J]. Intelligent Data Engineering and Learning,1999:263-290

[25] Davidson I,Satyanarayana A. Speeding up k-means clustering by bootstrap averaging[J]. IEEE Data Mining Workshop on Clustering Large Data Sets,2003,133(12):982-992

[26] Farnstrom F,Lewis J,Elkan C. Scalability for clustering algorithms revisited[J]. ACM SIGKDD Explorations Newsletter,2000,2(1):51-57

[27] Domingos P,Hulten G. A general method for scaling up machine learning algorithms and its application to clustering[C] // ICML. 2001:106-113

[28] Cui X,Zhu P,Yang X,et al. Optimized Big Data K-means clustering using MapReduce[J]. The Journal of Supercomputing,2014,70(3):1249-1259

[29] Zhao Y,Chen Y,Liang Z,et al. Big Data Processing with Probabilistic Latent Semantic Analysis on MapReduce[C] // International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery. 2014:162-166

[30] Younghoon K, Kyuseok S, Min-Soeng K, et al. DBCUREMR:An efficient density-based clustering algorithm for large data using MapReduce[J]. Information Systems,2014,42:15-35

[31] Jianqiang D, Fei W, Bo Y. Accelerating BIRCH for clustering large scale streaming data using CUDA dynamic parallelism [M]. Intelligent Data Engineering and Automated Learning-IDEAL 2013. Springer Berlin Heidelberg,2013:409-416

(上接第 379 页)

[58] Gama J,Rocha R,Medas P. Accurate decision trees for mining high-speed data streams[C] // Proceedings of the ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM,2003:523-528

[59] Bifet A, Holmes G,Kirkby R,et al. Moa:Massive online analysis [J]. The Journal of Machine Learning Research,2010,11:1601-1604

[60] Neumeyer L, Robbins B, Nair A, et al. S4:Distributed stream computing platform[C] // 2010 IEEE International Conference on Data Mining Workshops (ICDMW). IEEE,2010:170-177

[61] Wojnarski M. Debellor:a data mining platform with stream architecture[M] // Transactions on Rough Sets IX. Springer Berlin Heidelberg,2008:405-427

[62] Fan C Y,Chang P C,Lin J J,et al. A hybrid model combining case-based reasoning and fuzzy decision tree for medical data classification[J]. Applied Soft Computing,2011,11(1):632-644

[63] Wu M C,Lin S Y,Lin C H. An effective application of decision tree to stock trading[J]. Expert Systems with Applications,2006,31(2):270-274

[64] Kim J W,Lee B H,Shaw M J,et al. Application of decision-tree induction techniques to personalized advertisements on internet storefronts[J]. International Journal of Electronic Commerce,2001,5(3):45-62

[65] Zeitouni K, Chelghoum N. Spatial decision tree-application to traffic risk analysis[C] // ACS/IEEE International Conference on Computer Systems and Applications. IEEE,2001:203-207