

基于分形维数的聚类融合算法

吴晓璇^{1,2}, 倪志伟^{1,2}, 倪丽萍^{1,2}

(1. 合肥工业大学 管理学院商务智能研究所, 合肥 230009; 2. 合肥工业大学 过程优化与智能决策教育部重点实验室, 合肥 230009)

摘要: 对基于分形维数的聚类融合算法进行了研究。首先介绍分形维数聚类算法, 产生聚类成员; 然后利用投票法进行聚类融合; 最后简单介绍了云计算环境下分布式聚类融合思想。基于分形维数的聚类融合算法比单一分形维数聚类算法得到的聚类结果更好, 具有更好的鲁棒性。在分形维数聚类算法中, 结合网格聚类与单一分形聚类的优点, 提出了基于网格和分形维数的聚类算法, 它可以发现任意形状且距离非邻近的聚类, 适合于海量、高维数据。

关键词: 计算机系统结构; 网格; 分形维数; 投票策略; 聚类融合

中图分类号: TP311.13 **文献标志码:**A **文章编号:**1671-5497(2012)Sup. 1-0364-04

Clustering ensembles algorithm based on fractal dimension

WU Xiao-xuan^{1,2}, NI Zhi-wei^{1,2}, NI Li-ping^{1,2}

(1. School of Management and Business Intelligence Institute, Hefei University of Technology, Hefei 230009, China;
2. Key Laboratory of Process Optimization and Intelligent Decision-making, Ministry of Education, Hefei University of Technology, Hefei 230009, China)

Abstract: The fractal dimension based clustering ensembles algorithm was studied. It introduced clustering algorithm based on fractal dimension at first in order to create partitions for clustering ensembles, then using voting strategy to get ensembles result. Finally, an idea on distributed clustering ensembles under cloud computing environment was briefly discussed. Fractal dimension based clustering ensembles algorithm can offer better solutions in terms of robustness, novelty and stability than the single clustering algorithm based on fractal dimension. Combining the approaches based on grid and fractal, the clustering algorithm called grid and fractal dimension based clustering algorithm (GFDC) was presented to create partitions for clustering ensembles instead of clustering algorithm based on fractal dimension. GFDC is able to capture arbitrary shapes and non-neighboring clustering and can be applied to the massive and high-dimension dataset.

Key words: computer systems organization; grid; fractal dimension; voting strategy; clustering ensembles

聚类技术是数据挖掘的一个基本功能, 它作为最常用的探索性数据分析技术, 与分类预测方

法有明显不同之处, 它属于无监督的学习方法。聚类分析通过将数据样本划分成不同的类别, 使

收稿日期: 2012-03-20.

基金项目: “863”国家高技术研究发展计划项目(2011AA040501); 国家自然科学基金项目(71271071); 中央高校本科科研业务费专项资金资助项目(2011HGBZ1310).

作者简介: 吴晓璇(1984-), 女, 博士研究生, 研究方向: 分形数据挖掘, E-mail: kexinyufan@163.com

得同一个类内的数据对象的相似性尽可能大,不同类中的数据对象的差异性尽可能的小,从而发现和确定数据的结构分布,进一步描述数据和发现数据库中隐含的有用信息。

聚类融合技术^[1-3]是结合已有的聚类算法产生出(不同算法或者同一个算法不同的初始化和参数产生的样本数据)聚类成员,然后对聚类成员再次“聚类”达到最大化已有聚类结果共享信息的目的,从而摆脱了对原先数据分布敏感的缺陷,得到比单一算法更为优越的结果。聚类融合技术能够很好地提高聚类算法的鲁棒性和稳定性,并且能够实行并行计算。

本文结合聚类融合的概念^[4-6],提出了一种基于分形维数的聚类融合算法,它较之单一分形聚类算法,可以得到更好的聚类效果好。

1 算法描述

基于分形维数的聚类融合算法^[7],首先利用基于网格和分形维数的聚类算法^[8],通过选取不同的参数多次使用此算法产生聚类成员;再利用投票算法进行融合。因此从上述聚类融合的方法中得知,基于分形维数的融合聚类算法首先要解决两个问题:

- (1)如何产生 N 个具有差异性的聚类结果;
- (2)如何采用投票的方式对样本进行归类,得到最终的聚类结果。

1.1 聚类成员产生

在本文中用于产生聚类成员是利用基于网格和分形维数的聚类算法(Grid and fractal dimension based clustering, GFDC)^[8],分为初始聚类阶段和扩展聚类阶段。在初始聚类阶段,通过一遍扫描数据集来创建初始的底层网格结构,利用网格和密度的聚类技术在整个数据集中寻找部分点来生成初始类;在扩展阶段,利用初始阶段创建的网格结构,结合分形理论,将其他数据点以网格为单位添加到初始类中去,最终形成在整个数据集上的聚类结果。

对数据样本集进行 N 次随机抽样,产生 N 个数据样本集,对每个数据样本集利用 GFDC 进行聚类,共进行 N 次,则必然会产生 N 个具有差异性的聚类结果。在该算法之前,需要首先指定聚类的数量 k,给定样本数据集 $X = (X_1, X_2, \dots, X_N)$,最后形成 k 个聚类(C_1, \dots, C_k),具体算法步骤如下。

1.1.1 初始聚类

步骤 1 扫描数据集,构建 m 层网格结构。计算点集 X 中的每个点所在的底层网格坐标 $Grid_i = (g_{m_1}, g_{m_2}, \dots, g_{m_t})$,并统计每个底层网格所包含的数据点数 S_i 。

步骤 2 计算底层网格密度 $density(Grid_i)$,以网格密度最大的网格 $Grid_h$ 为中心,扫描其邻近的网格 $Grid_i$,若 $|density(Grid_h) - density(Grid_i)| < \tau$,则聚为一类,其中 τ 为密度差别阈值。

步骤 3 上一步聚得的类记为 C_i ,在剩余网格中选取单元网格密度最大的单元 $density(Grid_h)$,若 $|density(C_i) - density(Grid_h)| < \tau$,则将 $Grid_h$ 并入类 C_i ;然后重新搜寻剩余网格中密度最大者,否则将其与邻近网格的密度相比较,若差值小于密度差别阈值 τ ,则合并为一新类。

步骤 4 重复步骤 3,直至初始类别数达到用户设定的初始聚类数 k。

1.1.2 扩展聚类

步骤 1 分别计算在初始聚类阶段得到的 k 个初始类的分形维数 $f_i, i=1, 2, \dots, k$ 。

步骤 2 对于底层网格中没有被划分的网格逐个分别加入到各个初始类中,此时再计算各类的分形维数 $f'_i, i=1, 2, \dots, k$;令 $\Delta f_i = f'_i - f_i, i=1, 2, \dots, k$ 。

步骤 3 选取出 Δf_i 最小的类,对设定的阈值参数 δ ,若 $\min(\Delta f_i) \leq \delta, i=1, 2, \dots, k$,则将该网格中的数据点加入到这个类中;否则,新建一类,把该网格中数据点加入到新类。

步骤 4 重复步骤 2 和步骤 3,直至所有的网格都被归类。

步骤 5 步骤 4 得到聚类数 L 和各类的分形维数 $F_i, i=1, 2, \dots, L$ 。此时若 $L \geq 5k$,则令 $\Delta = |F_i - F_j|, i, j=1, 2, \dots, L$,对于给定的阈值参数 ϵ ,当 $\Delta \leq \epsilon$ 时,那么将 F_i 和 F_j 合并为一类,以合并后得到的结果作为最终的聚类结果。

1.2 融合过程——投票法则

在解决了如上两个问题之后,所有聚类结果的类标签就一致了。下面结合聚类融合的过程,给出基于投票策略^[5]的聚类融合算法的详细描述。

设定一个矩阵 $Matrix[N][K]$,N 为样本数据的个数,K 为类的个数,用来存放每一个样本属于某个类的次数,最后扫描矩阵 $Matrix[N][K]$,

记录每一个样本属于某个类的最大值,把样本归于次数最大的列所标识的类,得到最终的聚类结果。而矩阵 $Matrix[N][K]$ 产生的算法详细介绍如下:

设定一个矩阵 $MergeMtr[X][Y]$, X 为属于某个类的总样本数,Y 为样本的维数。用来存放隶属于某个类的所有样本 $Pattern[N][Y]$, N 为样本的总数,Y 为样本的维数,用于存放原始的样本,扫描并计算样本矩阵 $Pattern[N][Y]$ 中的每一个样本 $Pattern[i][Y]$ 分别在每一个类样本矩阵 $MergeMtr[X][Y]$ 中出现的次数。填充 $Matrix[N][K]$ 。

2 基于 MapReduce 的分布式分形维数聚类融合算法

在很多情况下,相关信息的获取和存储都是在地理上分散的环境下进行的,主要由于组织和操作上的限制造成的。然而,传统的机器学习算法中很多均是针对单机单处理器设计的,聚类算法也要求所有的数据都要集中在一起才能进行下去,不适用于分布式环境,使得对分布式数据挖掘的需求正在逐渐增长。接下来将介绍如何在分布式计算环境中改进算法完成相同的处理任务。

MapReduce 是近年来兴起的用于大规模数据集分布式的计算模型,帮助实现了大量的算法来处理海量原始数据。利用该模型,用户可以自定义其中的 Map 及 Reduce 函数来实现并行算法。数据文件首先被划分成小块,并以 $(key_{in}, value_{in})$ 的数据结构形式作为 Map 函数的输入,按式(1)同时输出一系列新的 $(key_{out}, value_{intermediate})$ 。MapReduce 框架收集所有 Map 的输出,并按照 key_{out} 值分组。最后,每一个 key_{out} 对应的组 $(key_{out}, list(value_{intermediate}))$ 按式(2)输入同一个 Reduce 函数。于是所有的 Reduce 过程并行生成一系列的 $value_{out}$ 。

$$Map(key_{in}, value_{in}) \rightarrow list(key_{out}, value_{intermediate}) \quad (1)$$

$$Reduce(key_{out}, list(value_{intermediate})) \rightarrow list(value_{out}) \quad (2)$$

MapReduce 作为并行处理海量数据集的编程模型,可使得程序在集群上并行运行^[9]。在基于网格和分形维数的聚类算法的基础上,利用

MapReduce 模型设计了分布式分形维数聚类算法。该算法是一个迭代的过程,迭代的每一步都对应一个 Job 的执行。每个 Job 涉及到 Map, Reduce 过程。Map 函数用来负责对所有的数据对应的网格分配到聚类中,为方便此过程,聚类代表先放在分布式缓存(Distributed Cache)中,之后读到 Map 中全局变量里。该算法包含三个 MapReduce 过程,如图 1 所示。第一个过程,Mapper1 把数据随机划分成 K 份,Reducer1 并行地在各个划分上进行 GFDC 算法,产生候选聚类代表;在第二个过程中只有一个 Mapper(图 1 中,记为 Mapper2),负责对候选聚类代表进行分形维数聚类算法,从而得到高质量的聚类;在第三阶段对应 Mapper3、Reducer3,用第二阶段形成的聚类代表为所有的数据对应的网格分配到聚类中,完成聚类。

当数据量非常大时,Reducer1 产生的聚类代表就会很多,此时需要进行层次采样,重复执行第一个过程。即在 Reducer1 产生的聚类代表之上再次执行 Mapper1、Reducer1 对应的过程,直至聚类代表的数量可以被 Mapper2 处理。紧接再执行 Mapper2 及其后续处理。基于分形维数聚类算法的 MapReduce 过程步骤如下:

(1) 在 Mapper1 中,需要使用一个随机数产生器,用来对数据进行随机划分 K 份。Reducer1 中,对数据集的一个划分利用 GFDC 算法生成聚类代表,并输出到 HDFS。

(2) 把各个划分上的聚类代表进行汇合后,并初始化整个聚类代表集合的分形维数利用分形维数聚类得到最终的聚类集合。因此这个 Job 中只用一个 Map 任务来得到最终的聚类成员,并输出到 HDFS 上。

(3) 在得到所有数据上的最终聚类成员后,在 Mapper3 中将这些聚类代表存放在分布式缓存(Distributed Cache)中,用来为所有的数据对应的网格分配到聚类中。再利用设计共识函数、投票等算法实现聚类融合,从而既实现了分布式聚类又得到比单一聚类算法更高效稳定的融合聚类结果。

在实现了基于 MapReduce 的分布式分形维数聚类融合算法后,可以把结果放到 Hadoop 平台的集群中,通过大量的数据进行试验,来验证算法的有效性及对大规模数据的适用性。

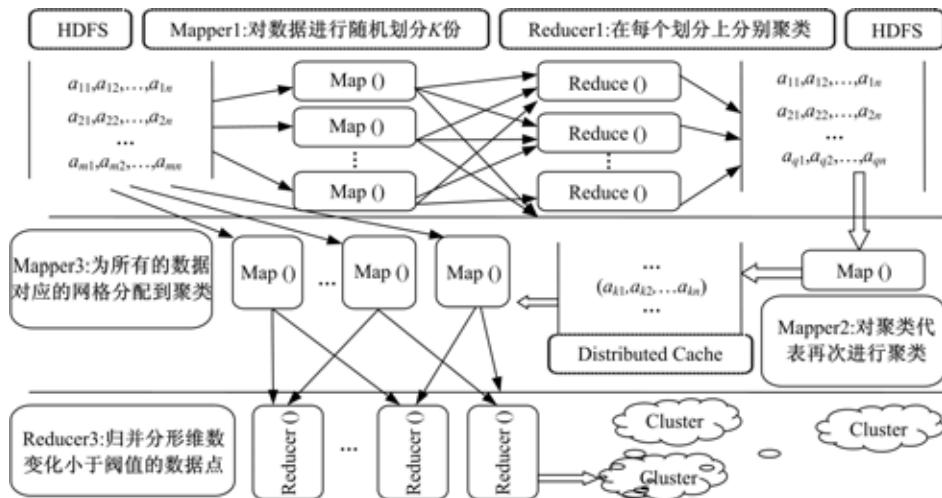


图 1 基于 MapReduce 的分布式分形维数聚类算法

Fig. 1 Based on the MapReduce distributed fractal dimension clustering algorithm

3 结束语

对基于分形维数的聚类融合算法进行了研究, 修改了传统的基于投票的聚类融合算法。在聚类成员产生阶段, 不是使用经典的 k-means 算法(该方法最大的优点是算法的复杂度低, 运行方便等; 但是对于边界难以区分的数据、分布非球形的数据以及在处理高维数据时, 该方法的效果并不理想), 而是采用基于网格和分形维数的聚类算法, 此算法在处理高维数据时非常有效。

参考文献:

- [1] 阳琳赟, 王文渊. 聚类融合方法综述[J]. 计算机应用研究, 2005(12): 8-10.
Yang Lin-yun, Wang Wen-yuan. Clustering ensemble approaches: overview[J]. Computer Application research, 2005(12): 8-10.
- [2] 蒋盛益. 基于投票机制的融合聚类算法[J]. 小型微型计算机系统, 2007(2): 306-309.
Jiang Sheng-yi. Custer fusion algorithm based on majority voting mechanism[J]. Journal of Chinese Computer Systems, 2007(2): 306-309.
- [3] 邹远强, 李国徽, 赵梓屹. 基于遗传和蚁群算法融合的聚类新方法[J]. 科学技术与工程, 2006(23): 4700-4705.
Zou Yuan-qiang, Li Guo-hui, Zhao Zi-yi. New clustering algorithm based on combination of genetic algo-

rithm and ant colony algorithm [J]. Science Technology and Engineering, 2006(23): 4700-4705.

- [4] Strehl A, Ghosh J. Cluster ensembles: a knowledge reuse framework for combining multiple partitions [J]. Journal of Machine Learning Research, 2003, 3 (3): 583-617.
- [5] Fred A L. Finding consistent clusters in data partitions [C] // Proceeding of the 2nd International Workshop on Multiple Classifier Systems, Volume 2096 of Lecture Notes in Computer Science. [s. l.]: Springer, 2001: 309-318.
- [6] Topchy A, Jain A K, Punch W. A mixture model for clustering ensembles [C] // Proceedings of the 4th SIAM International Conference on Data Mining, 2004: 379-390.
- [7] 倪志伟, 倪丽萍, 刘慧婷, 等. 动态数据挖掘[M]. 北京: 科技出版社, 2010: 107-110.
- [8] 梁敏君, 倪志伟, 倪丽萍, 等. 基于网格和分形维数的聚类算法[J]. 计算机应用, 2009, 29(3): 830-832.
Liang Min-jun, Ni Zhi-wei, Ni Li-ping, et al. Clustering algorithm based on Grid and Fractal dimension [J]. Journal of Computer Applications, 2009, 29(3): 830-832.
- [9] 杜晨阳. 分布式聚类算法研究与应用[D]. 杭州: 浙江大学计算机科学学院, 2011.
Du Chen-ying. Research and application of distributed clustering[D]. Hangzhou: College of Computer science, Zhejiang University, 2011.