



一种改进的 CLTree 算法

李卓航

(浙江大学信息与电子工程学院, 浙江 杭州 310058)

摘要: 针对聚类算法 CLTree 精度低、算法效率低的问题, 提出了 CLTree-R 算法, 之后将其应用于 UCI 数据集进行聚类分析。基于 Spark 平台的特性对数据进行并行处理, 加快了算法运行效率。实验结果也表明, 使用该算法对官方数据集进行聚类分析时, 可以得到较为合理的顾客划分。

关键词: 聚类; Spark; 数据挖掘; 并行化

中图分类号: TP399

文献标识码: A

doi: 10.11959/j.issn.1000-0801.2016214

An improved CLTree algorithm

LI Zhuohang

College of Information Science & Electronic Engineering, Zhejiang University, Hangzhou 310058, China

Abstract: An improved algorithm called CLTree-R was proposed. It could compensate the shortcoming of CLTree algorithm such as low accurate and inefficiency. Then CLTree-R was applied in clustering analysis for UCI data sets. In order to improve the efficiency, data set was parallel processed on Spark platform. Experimental results show that this algorithm can get reasonable customer classification when making cluster analysis on official data set.

Key words: clustering, Spark, data mining, parallelization

1 引言

聚类算法是数据挖掘十大算法之一^[1], 聚类定义为将物理或抽象对象的集合分成由类似对象组成的多个类的过程。聚类需要达成的目标是类间的差别尽量大, 而类内的差别尽量小, 通常被用于探索性分析。数据挖掘的精髓在于从海量价值密度低的数据中发现高价值的结论, 聚类可以应用于数据分析、图像分割及文件恢复等领域。

本文提出了一种改进的决策树归纳聚类 CLTree 算法^[2], 原算法的基本思想是把聚类问题转化为分类问题, 在进行决策树生长时采取信息增益的标准生成树的分支, 即 Quinlan J R^[3]提出的著名 ID3 算法中的度量标准, 而之后

的 C4.5 算法论证了采用信息增益比率这一度量标准比信息增益的效果好^[4], 本文使用改进的算法构造完 CLTree 之后, 再利用预剪枝策略实现聚类分析。最后基于 Spark 平台实现并行化处理, 提高了算法效率, 可以解决 GB 级以上数据的处理问题。

2 理论基础

2.1 CLTree 算法

首先, CLTree 算法是一种基于网格划分的典型聚类算法, 网格划分有由底向上和自顶向下两种, CLTree 算法采用了自顶向下的划分方法, 其优点在于无需指定划分参数、适用于高维数据、对噪音不敏感, 其划分过程如下所示。

步骤1 将数据空间分成 m 个区域。

步骤2 对每个区域进行划分。

步骤3 如满足划分停止规则转步骤2,否则转步骤4。

步骤4 停止划分。

CLTree 算法划分的标准是信息增益,根据这个划分标准,依照参考文献[4]的步骤建立决策树,其核心思想是提供构建决策树对数值型数据实现聚类分析,而决策树算法没有已知的类标签,不能直接进行聚类分析。可以通过将数据空间中的类别看成被低密度区域分割开的高密度区域,这样所有数据都具有 A 类标签,此时假设数据空间中存在另一种 B 类标签,把空间中的数据区域与空白区域加以分类,可以解决聚类问题。

2.2 C4.5 算法

C4.5 算法实质上是对 ID3 算法的一种扩展,另外 C4.5 算法还可以处理连续型数据,而 ID3 算法只能处理离散型数据,其计算式如下:

$$Gain(S,A)=Info(S)-\sum_{i=1}^m \frac{|S_i|}{|S|} \times Info(S_i) \quad (1)$$

其中, S 为样本集, A 为离散属性。 $Info(S)$ 是信息熵,是决策树进行正确判断时需要的信息量。设 S 中有 m 个类,则:

$$Info(S)=-\sum_{j=1}^m p_j \times \lg(p_j) \quad (2)$$

其中, p_j 为 S 中含有类 j 的概率。

C4.5 的选择准则为信息增益比率,其计算式如下:

$$GainRatio(S,A)=\frac{Gain(S,A)}{SplitInfo(S,A)} \quad (3)$$

划分信息计算式为:

$$SplitInfo(S,A)=-\sum_{i=1}^c P(v_i) \lg P(v_i) \quad (4)$$

其中, c 为划分的总数。 $SplitInfo(S,A)$ 是以属性 A 作为划分依据时, S 的广度与划分的均匀性。

C4.5 算法比 ID3 算法的信息增益大,可以解决多值属性的信息量倾斜问题。另外, C4.5 采用预剪枝策略控制决策树无止境增长,避免得到层数很小的无意义分类^[5]。

2.3 Spark 平台

Spark 平台是一个新生的分布式云计算平台。文件系统、数据库、数据处理系统、机器学习是 Spark 平台的组成部分。Spark 共有 4 层架构,即应用层、数据处理层、数据管理层、资源管理层。顶层负责把数据分组传递给 Spark 计

算平台,得到想要的处理结果。数据处理层对数据进行加工,是一种以内存为基础但不全依赖内存的计算,将计算结果回传给上层,即应用层。数据管理层的功能是共享平台内的信息。资源管理层的功能与 YARN 或者 Mesos 类似,可以为集群提供信息共享的管道。

Spark 生态系统指的是广义 Spark, 该 Spark 计算平台也含有 4 层架构且架构形式与 Hadoop 类似,如图 1 所示。

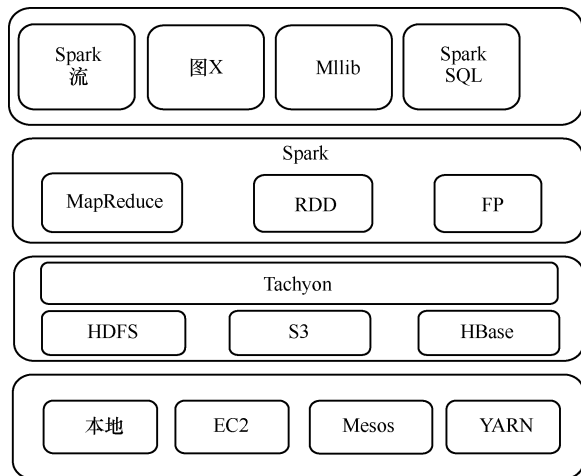


图 1 Spark 架构

2.4 聚类评价指标

芮氏指标(RI)是评价聚类效果的手段之一,其值越大,说明聚类的效果越好,其计算式为:

$$RI=\frac{TP+TN}{TP+FP+FN+TN} \quad (5)$$

其中, TP 为同一类的群体被划分到同一簇中, TN 为不同类的群体被划分到不同簇中, FP 为不同类的群体被划分到同一簇中, TN 为同一类的群体被划分到不同簇中。

3 CLTree-R 算法

3.1 CLTree 算法缺点及其改进

虽然 CLTree 能够很好地处理高维数据,但是还是存在些许不足:首先,CLTree 算法采用 ID3 经典算法里的划分标准来构建决策树,所以在把聚类分开时会偏向属性值较多的变量,因此分簇的精度会降低^[6]。其次,在划分过程中需要对数据集进行多次扫描,算法效率降低。

针对以上不足,提出两点改进: C4.5 算法中的度量标准要比原算法采用的度量标准好,所以把 CLTree 算法中的度量标准信息增益替换为信息增益比率,在 Spark 里通过新建 Entropy_Ratio 单例对象并混入 Impurity 特质实现



替换,提出 CLTree-R(CLTree-ratio)算法。利用可以并行处理大数据集的 Spark 平台解决算法效率低的问题。

3.2 CLTree-R 算法的 Spark 实现

在 Spark 平台中,RDD 是高度抽象的数据集合,它有 3 个固有点,分别是分区、函数、依赖。分区是为了并行,函数用来计算,而依赖是利用 DAG 图处理每个 RDD 先后关系的前提。

Spark 里的分区方式有 3 种:HashPartitioner、RangePartitioner、自定义。本文采用 RangePartitioner,因为采用 RangePartitioner 实现分区,能够尽量保证每个分区中数据量的均匀,而且分区之间是有序的,即每个分区中的元素都比另一个分区内的元素小或者大,但是分区内的元素不能保证顺序,简单说就是将一定范围内的数映射到某一个分区内。先进行分区,然后进行局部聚类,最后根据局部聚类好的数据再次进行聚类,最后进行规约操作。

主要实现代码如下:

Procedure CLTree-RTest(appName, master, jar, file, Ratio, maxDepth)

输入 应用程序名 A ,主节点 M ,程序 jar 包 J ,源数据 F ,信息增益比率计算方法 R ,树的最大深度 D 。

输出 trainErr。

Begin

New SparkConf scf

scf.appName <- A

scf.master <- M

New SparkContext sc

sc.jarLocation <- J

Load File F

FD <- F .split("").map(_toDouble)

Label <- Train(FD , F , D)

Predict <- Label.predict

trainErr <- FindNotEqual(Label,Predict) / FD

sc.stop

End

End CLTree-RTest

数据集,包括 9 个特征,共计 1 710 671 条交易记录。

采用 CentOS 操作系统,AMD Athlon 64×2 Dual Core Processor4000+的 CPU (主频 2.10 GHz,内存 2 GB)。Spark 平台配置:操作系统为 CentOS 6.5(64 bit),一个主节点,两个从节点。

4.2 实验结果与分析

本文将 CLTree-R 算法应用于葡萄牙出租车服务轨迹数据集,对乘客的相关信息和司机的表现进行聚类分析。基于 Spark 大数据处理平台,表 1 给出了 CLTree 算法改进前后的芮氏指标。表 2 描述了数据集的相关属性及其取值情况,共得到 20 个服务轨迹的聚类,表 3 给出了 3 个聚类结果的描述。

表 1 CLTree 算法改进前后的芮氏指标

算法	RI
CLTree	0.53
CLTree-R	0.67

表 2 数据集的相关属性及其取值情况

参数	取值
trip_ID	string
call_type	{A,B,C}
origin_call	如果 call_type='A',确定乘客;否则,为空值
origin_stand	如果 call_type='B',确定旅途的起始点,否则为空值
taxi_ID	integer
timestamp	integer
daytype	{D,E,F}
missing_data	{false,true}
polyline	{longitude, latitude}

表 3 聚类实验结果

类型	C3	C42	C21
trip_ID	T18	T104	T304
call_type	C	B	A
origin_call	NA	NA	59 708
origin_stand	NA	33	NA
taxi_ID	-	-	-
timestamp	-	-	-
daytype	{D}	{D,E}	{D,F}
missing_data	false	false	false
polyline	北部	中部	南部

从表 1 可以看出,用信息增益比率替换信息增益后,改进算法 CLTree-R 较原算法 CLTree 的聚类效果有所提升。

trip_ID 表示对于每个旅途来说都有一个唯一的 ID。

4 实验分析

4.1 实验数据与实验环境

实验数据选自 UCI^[7]数据库,选取 Taxi Service Trajectory

call_type 有 3 种:A 表示旅途服务是从中央大厅派出的;B 表示旅途服务是在一个具体街道面向出租车司机的;C 表示其他情况,比如随机地点随机叫车。origin_call 表示使用服务的每个机主号码;origin_stand 表示唯一的出租车招呼站;taxi_ID 表示每个旅途的出租车 ID;timestamp 表示旅途开始的时间戳;daytype 表示旅途日期类型:D 表示正常日期,如工作日、周末,E 表示节假日,F 表示节假日之前。missing_data 为布尔类型,有两种取值,false 表示 GPS 跟踪数据流完成,true 表示位置丢失。polyline 表示位置信息,用经纬度描述。

从表 3 可以看出,C3 主要是在非节假日随机叫车的乘客,这类乘客是出租车收入的主要群体,此类顾客理应得到日常服务。C42 主要是在节假日叫车且会要求出租车到达指定地点的乘客,说明这类乘客经济实力不错或者经济压力不是很大,针对此类乘客可以试着发展成为长期熟客以达到双赢。C21 主要是一些在节假日来临前叫车的乘客,且目的地大都是葡萄牙南部。由此得出,本文提出的改进算法 CLTree-R 可以合理划分不同类型的乘客。

5 结束语

本文提出了一种改进之后的 CLTree-R 算法,实验分析和测试表明,该方法可以合理划分不同种类的乘客,进而让出租车公司更好地服务于大众。基于 Spark 平台,采用了较好的分区策略可以让算法更快地运行。将以上特性应用于 Taxi Service Trajectory 数据集,对高层将有极大的帮助。

参考文献:

- [1] 韩家炜.数据挖掘:概念与技术[M].北京:机械工业出版社,2012.
HAN J W. Data mining: concepts and techniques [M]. Beijing: China Machine Press, 2012.
- [2] DUNHAM M H. Datamining introductory and advanced topics[M]. New York: ACM Press, 2002: 23-60.
- [3] QUINLAN J R. Machine learning [M]. Berlin: Springer, 1986: 81-106.
- [4] QUINLAN J R. C4.5:program for machine learning [M]. New York: ACM Press, 1993.
- [5] 薛薇,陈欢歌. SPSS Modeler 数据挖掘方法及应用[M]. 北京:电子工业出版社,2014.
XUE W, CHEN H G. Data mining method and its application of SPSS Modeler [M]. Beijing: Publishing House of Electronics Industry, 2014.
- [6] 伍育红. 聚类算法综述[J]. 计算机科学, 2015, 42(6A): 491-499.
WU Y H. General overview on clustering algorithms [J]. Computer Science, 2015, 42(6A): 491-499.
- [7] BLAKE C. UCI repository of machine learning database [J]. Neural Information Processing Systems, 1998.

[作者简介]



李卓航(1994-),男,浙江大学信息与电子工程学院本科在读。