ISSN 1009-3044
Computer Knowledge and Technology 电脑知识与技术
Vol.11, No.4, February 2015
E-mail: xsjl@dnzs.net.cn
http://www.dnzs.net.cn
Tel:+86-551-65690963 65690964

基于Spark框架的聚类算法研究

陈虹君

(电子科技大学成都学院,四川 成都 611731)

摘要:大数据的挖掘是当今的研究热点,也有着巨大的商业价值。新型框架Spark部署在Hadoop平台上,它的机器学习算法几乎可以完全替代传统的Mahout MapReduce的编程模式,但由于Spark的内存模型特点,执行速度快。该文研究了Spark中的机器学习中的聚类算法KMeans,先分析了算法思想,再通过实验分析其应用的方法,然后通过实验结果分析其应用场景和不足。

关键词:大数据;Hadoop;Spark;机器学习;聚类;KMeans

中图分类号:TP311 **文献标识码**:A **文章编号**:1009-3044(2015)04-0056-02

Research on Clustering Algorithm Based on Spark Framework

CHEN Hong-jun

(ChengDu College of University Of Electronic Science And Technology of China, Chengdu 611731, China)

Abstract: Mining big data is current research hotspot, also have a huge commercial value.A new framework of Spark is deployed on the Hadoop platform, in which machine learning algorithms can be almost completely replace the traditional Mahout MapReduce programming mode. But the characteristics of Spark memory model, efficiency of execution is high. This paper studies the KMeans clustering algorithm in Spark machine learning。 The first analyze the idea of the algorithm, and then through the experimental analyze method and its application, and then through results of experimental analyze its application scenarios and lacks.

Key words: big data; Hadoop; Spark; machine learnin; clustering; KMeans

大数据的挖掘是当今的研究热点,也具有很大的商业价值。传统方式在大数据Hadoop平台上利用Mahout以MapReduce的编程方式做数据挖掘,但是有一定的局限,比如效率较低。Spark框架称为快数据,是基于内存的编程模型,它可以把中间的迭代过程不放在磁盘中,直接数据不落地在内存中执行,极大地提高了它的执行速度。Spark是大数据挖掘的新型利器。

1 Spark框架上的机器学习算法

在Spark框架中机器学习几乎可以完全替代Hadoop平台上传统的Mahout,并且具有更高的效率。Spark的机器学习有分类和回归:线性模型、支持向量机、逻辑回归、线性回归;决策树,朴素贝叶斯;协同过滤:交替最小二乘法;聚类:KMeans聚类;降维:奇异值分解,主成分分析;特征提取和变换等。其中KMeans聚类算法是常用的算法,该文将对该算法做详细研究。

2 Spark中KMeans聚类算法分析

聚类算法KMeans:接受输入量K;然后将N个数据对象划分为K个聚类以便使得所获得的聚类满足:同一聚类中的对象相似度较高;而不同聚类中的对象相似度较小。聚类相似度是利用各聚类中对象的均值所获得一个“中心对象”,也称为引力中心,来进行计算的。若有N个数据点需要分为K个cluster,k-means要做的就是最小化。其算法数学表达式(1):

$$J = \sum_{n=1}^{N}\sum_{k=1}^{K} r_{nk}\|x_n - \mu_k\|^2 \tag{1}$$

其中x_n表示原始数据的集合$\{x_1, x_2, x_3, x_4, \cdots, x_n\}$中的元素,K表示给定分类组数K($k \leqslant n$)值的条件下,将原始数据分成K类。$\mu_k$表示分类的平均值。$r_{nk}$表示数据点n被归类到族(cluster)k的时候为1,否则为0。

KMeans在Spark仍然是上面的算法思想。KMeans是org.apache.spark.mllib.clustering包中提供的类,即是spark官方为clustering(聚类)所提供的类,当然也包含了许多方法。

KMeansModel是数据类型,换句话说就是KMeans的一种特定的数据计算模型。train()方法

train(RDD<Vector> data, int k, int maxIterations, int runs,String initializationMode) (2)

第一个是需要计算的数据,第二个是K值,第三个是最大迭代次数,第四个是对整个KMeans模型的计算次数,第五个是初始化模型的方式。该方法返回一个已经(按给定参数)规划好的KMeans模型。这里要说明一下的是train()方法有多种方法重构,除了

收稿日期:2014-12-06
作者简介:陈虹君(1979-),女,四川成都人,副教授,硕士,研究方向为大数据应用。

参数不同,Spark对其他聚类方法如ALS类也有专属的train()方法。这个方法是Junit的断言测试用,可以很好的看出结果是否有问题。

assertEquals([String message],expected,actual) (3)

参数说明:message是个可选的消息,假如提供,将会在发生错误时报告这个消息;expected是期望值,通常都是用户指定的内容;actual是被测试的代码返回的实际值。若expected和actual不相等则会判错,否则算成功。

3 实验分析

为了应用聚类算法,通过三组数据,得到按expectedCenter设置的聚类状态,我们编写了相应程序,以下是程序的片段(4):

sc.addJar("/home/haduser/JavaKMeansSuite.jar"); (4)

```
//分配jar包给worker节点,注意路径
JavaRDD<Vector> data = sc.parallelize(points, 2);
KMeansModel model = KMeans.train(data.rdd(), 1, 1, 1, KMeans.K_MEANS_PARALLEL());
assertEquals(1, model.clusterCenters().length);
assertEquals(expectedCenter, model.clusterCenters()[0]);
```

实验数据输入:

```
List<Vector> points = Lists.newArrayList(
  Vectors.dense(1.0, 2.0, 6.0),
  Vectors.dense(1.0, 3.0, 0.0),
  Vectors.dense(1.0, 4.0, 6.0)
);

Vector expectedCenter = Vectors.dense(1.0, 3.0, 4.0);
```

图1 实验数据输入

输出:

```
[1.0,3.0,4.0],[1.0,3.0,4.0]
```

图2 实验数据输出

分析:可以看出本来是三组数据中各摆放一些数据,但通过KMeans算法,我们成功得到了按expectedCenter(理想的聚类形态)设置的结果集,即将分别位于三处不同地方的数据按照某种设置集中在了一起。

4 UI显示

由于源代码中有三个@test,所以运行的程序也有三个:

ID	Name	Cores	Memory per Node	Submitted Time	User	State	Duration
app-20141016224254-0005	JavaKMeans	0	512.0 MB	2014/10/16 22:42:54	haduser	FINISHED	9 s
app-20141016224244-0004	JavaKMeans	0	512.0 MB	2014/10/16 22:42:44	haduser	FINISHED	10 s
app-20141016224230-0003	JavaKMeans	0	512.0 MB	2014/10/16 22:42:30	haduser	FINISHED	13 s

图3 UI显示

尝试为输入数据添加了一行vectors.dense(1.0, 6.0, 5.0),然后将expectedCenter设置为(1.0, 3.0, 5.0, 4.0)。然后再次运行时assertEquals不出意料地判错了,而且错误很明显聚合结果中的元素只有3个,于是我将输入数据再次修改为:

```
Vectors.dense(1.0, 2.0, 6.0, 7.0),
Vectors.dense(1.0, 3.0, 0.0, 2.0),
Vectors.dense(1.0, 4.0, 6.0, 2.0),
Vectors.dense(1.0, 6.0, 5.0, 7.0)
```

图4 重新输入实验数据

但是assertEuqals再次判错,得到结果为:

```
but was:<[1.0,3.75,4.25,4.5]>
```

图5 输出结果

不过就算法由1变成100次结果还是不变的。这也表明KMeans算法的其中一个缺点:需要根据初始聚类中心来确定一个初始划分,一旦初始值选择的不好,可能无法得到有效的聚类结果

5 总结

聚类KMeans算法是聚类分析中的常用算法,它是数据划分或者分组处理的重要方式,目前在电子商务、生物科学、图像处理、Web文档分类等领域都得到了有效的应用。比如利用聚类KMeans将客户细分为是哪种用户类型,以便于推荐合适的产品。目前,对于KMeans还有很到要优化的地方,比如如何自主式聚类,尽量避免由于初始值选择,而影响到聚类的效果。 (下转第60页)

如使用控制流图,假如7个函数平均存在3条基本路径,则其路径总数达到:$R_{total}=3+3^3+3^4+3^3=138$ 条,相差2个数量级。此外,数学计算中使用公式比较多,所以程序实现时一个函数实现一个基本公式,对于这些基本公式函数使用黑盒测试比较容易,而且充分。

3 嵌入武控软件数学模型测试试验

为了测试数据有可比较性,把测试人员分成能力相当A、B 2个小组,每个组2个人。A组与B组同时对两个测试项目进行测试:某型武控系统目标数据滤波软件与某型武控系统杀伤区解算软件(两个软件都使用标准C编程实现,使用Tonado开发环境软件实现编译,运行在Vxworks操作系统下;第一个软件共3567行代码,32个函数,86个变量;第二软件共5726行代码,46个函数,152个变量)。

A组使用传统人工测试方法与自动测试工具软件(Logiscope、TestBed和PolySpace)完成以上两项测试工作,B小组使用以上描述的函数调用选择法配合自动化测试工具软件(Logiscope与McCabe)及人工测试完成以上两项测试工作。测试结果数据统计如表1。

从表1中可分析出,B组测试使用的时间更少,但发现的缺陷数却更多,在软件规模上增加后,A小组甚至于在语句上不能达100%的覆盖。所以,由以上试验数据表明此针对数学模型的测试相对于传统的测试方法更加有效与充分。

4 结束语

本文简要介绍了嵌入武控软件数学模型测试的难点与重点,针对数学模型的自身相关特点,提出了适合于嵌入武控软件数学模型测试的方法,详细分析此方法的实现过程,并且通过试验分析比较传统测试方法与函数调用选择法。

参考文献:

[1] Shan Jinhui, Jiang Ying, Sun Ping. Research Progress in Software Testing[J].Acta Scientiarum Naturalium Univ-ersitatis Pekinensis, 2005,41(1):134-145.

[2] Harrold M J,Rothermel G.A system for research on and development of program analysis-basedtools[R].OSU-CISRC-3/97-TR17, The Ohio State University, 1997.

[3] 于屏岗.面向故障的软件自动测试技术[D].北京:装甲兵工程学院,2005.

[4] Hemant D P, William P L, Barbara G R. Interprocedural Def-Use associations for C Systems with Single LevelPointers[J]. IEEE Transactions on Software Engineering,1994,20(5): 385-403.

[5] 丁振国. C/C++语言故障模型研究[D].北京:装甲兵工程学院,2005.

(上接第57页)

参考文献:

[1] 编程指南[EB/OL].http://spark.apache.org/docs/latest/programming-guide.html,Spark,2013.

[2] 机器学习库[EB/OL].http://blog.csdn.net/johnny_lee/article/details/25656343,2013.

[3] 最近的spark文档[EB/OL].http://spark.apache.org/docs/latest/,2014.

[4] 聚类算法的研究与应用[EB/OL].http://www.docin.com/p-599574449.html,2014.