(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 106127244 A
(43)申请公布日 2016.11.16

(21)申请号 201610455944.7

(22)申请日 2016.06.22

(71)申请人 TCL集团股份有限公司
地址 516006 广东省惠州市仲恺高新技术开发区十九号小区

(72)发明人 冯研

(74)专利代理机构 深圳市君胜知识产权代理事务所 44268
代理人 王永文 刘文求

(51)Int.Cl.
G06K 9/62(2006.01)

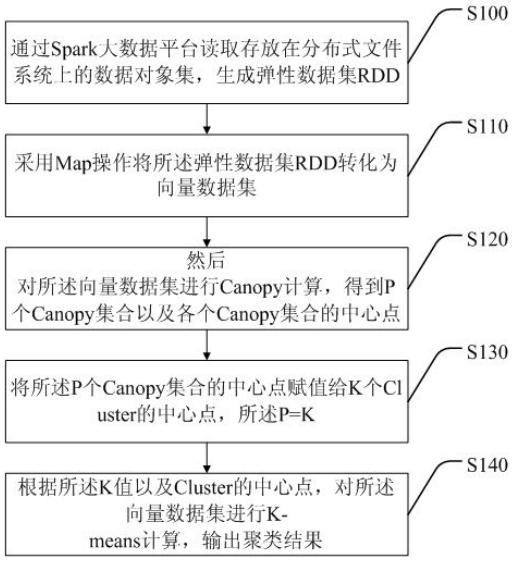
权利要求书2页 说明书7页 附图2页

(54)发明名称

一种并行化K-means改进方法及系统

(57)摘要

本发明公开一种并行化K-means改进方法及系统，其中，所述方法包括步骤：通过Spark大数据平台读取存放在分布式文件系统上的数据对象集，生成弹性数据集RDD；采用Map操作将所述弹性数据集RDD转化为向量数据集；通过Map操作对所述向量数据集进行Canopy计算，得到P个Canopy集合以及各个Canopy集合的中心点；将所述P个Canopy集合的中心点赋值给K个Cluster的中心点；通过Map操作对所述向量数据集进行K-means计算，输出聚类结果。通过本发明方法可解决K-means算法聚类结果稳定性和准确性差的问题；并有效提高K-means算法的执行效率。



CN 106127244 A

1.一种基于Spark平台实现的并行化K-means改进方法,其特征在于,包括步骤:

A、通过Spark大数据平台读取存放在分布式文件系统上的数据对象集,生成弹性数据集RDD;

B、采用Map操作将所述弹性数据集RDD转化为向量数据集;

C、然后对所述向量数据集进行Canopy计算,得到P个Canopy集合以及各个Canopy集合的中心点;

E、将所述P个Canopy集合的中心点赋值给K个Cluster的中心点,所述P=K;

F、根据所述K值以及Cluster的中心点,对所述向量数据集进行K-means计算,输出聚类结果。

2.根据权利要求1所述的基于Spark平台实现的并行化K-means改进方法,其特征在于,所述步骤A之前还包括:

A0、配置Spark参数并初始化Spark环境。

3.根据权利要求1所述的基于Spark平台实现的并行化K-means改进方法,其特征在于,所述步骤C具体包括:

C1、采用Map操作计算所述向量数据集中的数据对象与初始Canopy集合中心点的距离,当所述距离均大于T1时,则所述数据对象生成新的Canopy集合,并产生新的Canopy中心点;

C2、通过合并所述初始Canopy集合以及新生成的Canopy集合生成全局Canopy集合,以及通过合并所述初始Canopy集合中心点以及新生成的Canopy集合中心点生成全局Canopy中心点集合;

C3、采用Map操作计算所述向量数据集中的数据对象与全局Canopy中心点集合中的中心点之间的距离,当所述距离小于T2时,则对所述数据对象进行标记并将所述数据对象划分到与相应中心点对应的Canopy集合中,最终得到P个Canopy集合及其中心点数据。

4.根据权利要求3所述的基于Spark平台实现的并行化K-means改进方法,其特征在于,所述步骤C3还包括:

C31、对划分到所述Canopy集合中的数据对象执行Cache操作,使所述数据对象存放在内存中。

5.根据权利要求1所述的基于Spark平台实现的并行化K-means改进方法,其特征在于,所述步骤F具体包括:

F1、采用Map操作计算所述向量数据集中的数据对象与所述初始Cluster中心点的距离,将所述数据对象划入到与所述数据对象距离最短的Cluster中心点所在的Cluster,生成新的Cluster;

F2、采用reduceByKey操作计算所述新生成的Cluster中数据对象的平均值,作为新的Cluster中心点;

F3、采用Map操作计算新生成的Cluster中心点与所述初始Cluster中心点的平方差,当所述平方差小于等于预定阈值时,输出聚类结果;当所述平方差大于预定阈值时,则返回步骤F1,直至输出聚类结果。

6.一种基于Spark平台实现的并行化K-means改进系统,其特征在于,包括:

读取模块,用于通过Spark大数据平台读取存放在分布式文件系统上的数据对象集,生成弹性数据集RDD;

向量化模块，用于采用Map操作将所述弹性数据集RDD转化为向量数据集；

Canopy计算模块，用于通过Map操作对所述向量数据集进行Canopy计算，得到P个Canopy集合以及各个Canopy集合的中心点；

赋值模块，用于将所述P个Canopy集合的中心点赋值给K个Cluster的中心点，所述P=K；

K-means计算模块，用于根据所述K值以及Cluster的中心点，通过Map操作对所述向量数据集进行K-means计算，输出聚类结果。

7.根据权利要求6所述的基于Spark平台实现的并行化K-means改进系统，其特征在于，所述系统还包括：

初始化模块，配置Spark参数并初始化Spark环境。

8.根据权利要求6所述的基于Spark平台实现的并行化K-means改进系统，其特征在于，所述Canopy计算模块具体包括：

新Canopy集合生成单元，用于采用Map操作计算所述向量数据集中的数据对象与初始Canopy集合中心点的距离，当所述距离均大于T1时，则所述数据对象生成新的Canopy集合，并产生新的Canopy中心点；

合并单元，用于合并所述初始Canopy集合中心点以及新生成的Canopy集合中心点，生成全局Canopy中心点集合；

数据对象划分Canopy单元，用于采用Map操作计算所述向量数据集中的数据对象与全局Canopy中心点的距离，当所述距离小于T2时，则对所述数据对象进行标记并将所述数据对象划分到所述Canopy集合中，最终得到P个Canopy集合及其中心点数据。

9.根据权利要求8所述的基于Spark平台实现的并行化K-means改进系统，其特征在于，所述数据对象划分Canopy单元还包括：

存储子单元，用于对划分到所述Canopy集合中的数据对象执行Cache操作，使所述数据对象存放在内存中。

10.根据权利要求6所述的基于Spark平台实现的并行化K-means改进系统，其特征在于，所述K-means计算模块具体包括：

数据对象划分Cluster单元，用于采用Map操作计算所述向量数据集中的数据对象与所述初始Cluster中心点的距离，将所述数据对象划入到与所述数据对象距离最短的Cluster中心点所在的Cluster，生成新的Cluster；

新Cluster中心点生成单元，用于采用reduceByKey操作计算所述新生成的Cluster中数据对象的平均值，作为新的Cluster中心点；

聚类结果输出单元，用于采用Map操作计算新生成的Cluster中心点与所述初始Cluster中心点的平方差，当所述平方差小于等于预定阈值时，输出聚类结果；当所述平方差大于预定阈值时，返回数据对象划分Cluster单元，直至输出聚类结果。

一种并行化K-means改进方法及系统

技术领域

[0001] 本发明涉及聚类算法领域，尤其涉及一种基于Spark平台实现的并行化K-means改进方法及系统。

背景技术

[0002] 在数据挖掘中，聚类分析占据重要的地位，其是伴随着数据挖掘的发展而发展起来的，其可以将一些无序的数据进行自动的分组，也是探索性数据挖掘的主要任务之一，聚类分析算法根据数据对象的属性去创建各种群组(cluster)，也称是“簇”。目前主流的标准有群组内成员之间距离、数据空间密度、迭代和特定的统计分布。在这些已有的算法中，K-means 算法是最常见也是最简单的算法，能很好的应用到我们的应用中。然而，在大数据环境下，已有传统算法并不能有效的完成大数据的挖掘，所以需要利用并行计算技术来提高聚类分析算法的处理能力和效率。

[0003] 在数据挖掘中，K-means 算法是人们经常接触的划分式聚类分析算法。K-means算法目标是将待分区的数据对象，划分到k 个簇(cluster)中，这些数据对象和其所属簇的中心的距离最近，K-Means算法的结果是对数据空间进行沃罗努瓦分割。传统的K-means算法在实际应用时存在着以下的缺陷：

1、K-means 算法的复杂度由其迭代次数决定的，迭代次数较高时，算法的复杂度较高；

2、在K-means 算法中k 值是预设的，是根据应用使用者的经验提出的，其准确性不高。当k 值过大的时候会影响到算法的效率，当然重要的是对数据划分过细，实际并没有用处，当k 值过小的时候，没有起到聚类算法将数据分类的作用；

3、随着需要处理的数据量的增大，算法在计算的空间和时间代价也会随着之变大，当处理数据时，需要将数据从文件中读取，这样给I/O、CPU 和内存等系统资源造成巨大的压力；

4、由于K-means算法中其分配后的Cluster 的范围多是椭圆型，无法识别其他形状的Cluster。

[0004] 因此，现有技术还有待于改进和发展。

发明内容

[0005] 鉴于上述现有技术的不足，本发明的目的在于提供一种基于Spark平台实现的并行化K-means改进方法及系统，旨在解决现有K-means算法在处理大数据时运算效率低、且聚类结果的稳定性和准确性差的问题。

[0006] 本发明的技术方案如下：

一种基于Spark平台实现的并行化K-means改进方法，其中，包括步骤：

A、通过Spark大数据平台读取存放在分布式文件系统上的数据对象集，生成弹性数据集RDD；

B、采用Map操作将所述弹性数据集RDD转化为向量数据集；

C、然后对所述向量数据集进行Canopy计算，得到P个Canopy集合以及各个Canopy集合的中心点；

E、将所述P个Canopy集合的中心点赋值给K个Cluster的中心点，所述P=K；

F、根据所述K值以及Cluster的中心点，对所述向量数据集进行K-means计算，输出聚类结果。

[0007] 较佳地，所述的基于Spark平台实现的并行化K-means改进方法，其中，所述步骤A之前还包括：

A0、配置Spark参数并初始化Spark环境。

[0008] 较佳地，所述的基于Spark平台实现的并行化K-means改进方法，其中，所述步骤C具体包括：

C1、采用Map操作计算所述向量数据集中的数据对象与初始Canopy集合中心点的距离，当所述距离均大于T1时，则所述数据对象生成新的Canopy集合，并产生新的Canopy中心点；

C2、通过合并所述初始Canopy集合以及新生成的Canopy集合生成全局Canopy集合，以及通过合并所述初始Canopy集合中心点以及新生成的Canopy集合中心点生成全局Canopy中心点集合；

C3、采用Map操作计算所述向量数据集中的数据对象与全局Canopy中心点集合中的中心点之间的距离，当所述距离小于T2时，则对所述数据对象进行标记并将所述数据对象划分到与相应中心点对应的Canopy集合中，最终得到P个Canopy集合及其中心点数据。

[0009] 较佳地，所述的基于Spark平台实现的并行化K-means改进方法，其中，所述步骤C3还包括：

C31、对划分到所述Canopy集合中的数据对象执行Cache操作，使所述数据对象存放在内存中。

[0010] 较佳地，所述的基于Spark平台实现的并行化K-means改进方法，其中，所述步骤F具体包括：

F1、采用Map操作计算所述向量数据集中的数据对象与所述初始Cluster中心点的距离，将所述数据对象划入到与所述数据对象距离最短的Cluster中心点所在的Cluster，生成新的Cluster；

F2、采用reduceByKey操作计算所述新生成的Cluster中数据对象的平均值，作为新的Cluster中心点；

F3、采用Map操作计算新生成的Cluster中心点与所述初始Cluster中心点的平方差，当所述平方差小于等于预定阈值时，输出聚类结果；当所述平方差大于预定阈值时，则返回步骤F1，直至输出聚类结果。

[0011] 一种基于Spark平台实现的并行化K-means改进系统，其中，包括：

读取模块，用于通过Spark大数据平台读取存放在分布式文件系统上的数据对象集，生成弹性数据集RDD；

向量化模块，用于采用Map操作将所述弹性数据集RDD转化为向量数据集；

Canopy计算模块，用于对所述向量数据集进行Canopy计算，得到P个Canopy集合以及各个Canopy集合的中心点；

赋值模块，用于将所述P个Canopy集合的中心点赋值给K个Cluster的中心点，所述P=K；

K-means计算模块，用于根据所述K值以及Cluster的中心点，对所述向量数据集进行K-means计算，输出聚类结果。

[0012] 较佳地，所述的基于Spark平台实现的并行化K-means改进系统，其中，所述系统还包括：

初始化模块，配置Spark参数并初始化Spark环境。

[0013] 较佳地，所述的基于Spark平台实现的并行化K-means改进系统，其中，所述Canopy计算模块具体包括：

新Canopy集合生成单元，用于采用Map操作计算所述向量数据集中的数据对象与初始Canopy集合中心点的距离，当所述距离均大于T1时，则所述数据对象生成新的Canopy集合，并产生新的Canopy中心点；

合并单元，通过合并所述初始Canopy集合以及新生成的Canopy集合生成全局Canopy集合，以及通过合并所述初始Canopy集合中心点以及新生成的Canopy集合中心点生成全局Canopy中心点集合；

数据对象划分Canopy单元，用于采用Map操作计算所述向量数据集中的数据对象与全局Canopy中心点集合中的中心点之间的距离，当所述距离小于T2时，则对所述数据对象进行标记并将所述数据对象划分到与相应中心点对应的Canopy集合中，最终得到P个Canopy集合及其中心点数据。

[0014] 较佳地，所述的基于Spark平台实现的并行化K-means改进系统，其中，所述数据对象划分Canopy单元还包括：

存储子单元，用于对划分到所述Canopy集合中的数据对象执行Cache操作，使所述数据对象存放在内存中。

[0015] 较佳地，所述的基于Spark平台实现的并行化K-means改进系统，其中，所述K-means计算模块具体包括：

数据对象划分Cluster单元，用于采用Map操作计算所述向量数据集中的数据对象与所述初始Cluster中心点的距离，将所述数据对象划入到与所述数据对象距离最短的Cluster中心点所在的Cluster，生成新的Cluster；

新Cluster中心点生成单元，用于采用reduceByKey操作计算所述新生成的Cluster中数据对象的平均值，作为新的Cluster中心点；

聚类结果输出单元，用于采用Map操作计算新生成的Cluster中心点与所述初始Cluster中心点的平方差，当所述平方差小于等于预定阈值时，输出聚类结果；当所述平方差大于预定阈值时，则返回数据对象划分Cluster单元，直至输出聚类结果。

[0016] 有益效果：本发明先采用Canopy算法对大数据进行预聚类处理，粗糙地将数据划分到覆盖的Canopy集合中，使得K-means算法中的初始中心节点不是随机产生，并且K-means算法中的K值选取时可以参照Canopy集合的个数P，从而解决K-means算法聚类结果稳定性和准确性差的问题；并且本发明基于Spark平台将Canopy算法应用到K-means聚类算法前能高效地加速聚类操作，降低K-means算法中的迭代次数，从而有效提高改进的-means算法的执行效率，避免局部最优。

附图说明

[0017] 图1为本发明一种基于Spark平台实现的并行化K-means改进方法较佳实施例的流程图。

[0018] 图2为本发明一种基于Spark平台实现的并行化K-means改进系统较佳实施例的结构框图。

具体实施方式

[0019] 本发明提供一种基于Spark平台实现的并行化K-means改进方法及系统，为使本发明的目的、技术方案及效果更加清楚、明确，以下对本发明进一步详细说明。应当理解，此处所描述的具体实施例仅仅用以解释本发明，并不用于限定本发明。

[0020] 请参阅图1，图1为本发明一种基于Spark平台实现的并行化K-means改进方法较佳实施例的流程图，其包括步骤：

S100、通过Spark大数据平台读取存放在分布式文件系统上的数据对象集，生成弹性数据集RDD；

S110、采用Map操作将所述弹性数据集RDD转化为向量数据集；

S120、然后对所述向量数据集进行Canopy计算，得到P个Canopy集合以及各个Canopy集合的中心点；

S130、将所述P个Canopy集合的中心点赋值给K个Cluster的中心点，所述P=K；

S140、根据所述K值以及Cluster的中心点，对所述向量数据集进行K-means计算，输出聚类结果。

[0021] 首先，在本发明中，所述步骤S100之前还包括步骤S10，配置Spark参数并初始化Spark环境。具体地，本发明采用Spark大数据平台对大数据进行聚类处理，在Spark平台中，HdfsPath为存放数据对象集的地址，master为Spark的集群地址，deltaDist为K-means计算部分的收敛阈值。所述步骤S10在Spark平台上的实现代码为：casc=new SparkContext(master,“SparkCK”)；通过配置Spark参数以及初始化Spark环境便于之后的聚类计算。

[0022] 进一步，在所述步骤S100中，通过Spark大数据平台读取存放在分布式文件系统上的数据对象集，生成弹性数据集RDD，其在Spark平台上的实现代码为：ckLine=casc.textFile(HdfsPath)；在本发明中，所述弹性数据集RDD指的是一个只读的，可分区的分布式数据集，这个数据集的全部或部分可以缓存在内存中，可在多次计算间重用。具体地，所述RDD(弹性数据集)是Spark提供的最重要的抽象的概念，它是一种有容错机制的特殊集合，可以分布在集群的节点上，以函数式编操作集合的方式，进行各种并行操作。

[0023] 进一步，所述RDD在应用时具有多个优点：a、它是分布式的，可以分布在多台机器上，进行计算；b、它是弹性的，计算过程中内存不够时它会和磁盘进行数据交换；c、实质是一种更为通用的迭代并行计算框架，用户可以显示的控制计算的中间结果，然后将其自由运用于之后的计算。

[0024] 传统的并行计算模型无法有效的解决迭代计算(iterative)，而本发明采用Spar解决迭代计算，其主要实现思想就是把所有计算的数据保存在分布式的内存中。

[0025] 在本发明所述步骤S110中，采用Map操作将所述弹性数据集RDD转化为向量数据集，其在Spark平台上的实现代码为：dataObjects=ckLine. map(parseVector(_))。本发明通过将所述弹性数据集RDD转化为向量数据集，为后面的距离计算做准备。

[0026] 进一步，所述步骤S120通过Map操作对所述向量数据集进行Canopy计算，得到P个Canopy集合以及各个Canopy集合的中心点，具体包括：

S121、采用Map操作计算所述向量数据集中的数据对象与初始Canopy集合中心点的距离，当所述距离均大于T1时，则所述数据对象生成新的Canopy集合，并产生新的Canopy中心点，所述T1为Canopy算法中的固定阈值；所述步骤S121在Spark平台上的实现代码为：canopyMapcenter=new HashSet()

```
rawCenters = dataObjects.map{
x =>(x._1, canopy_1(x, canopyMapcenter, t1))}
.filter(y => y._2 != null).collect().toList;
```

S122、通过合并所述初始Canopy集合以及新生成的Canopy集合生成全局Canopy集合，以及通过合并所述初始Canopy集合中心点以及新生成的Canopy集合中心点生成全局Canopy中心点集合；所述步骤S122在Spark平台上的实现代码为：canopyCenters= new HashSet()

```
for (i <- 0 until rawCenters.size) {
canopy_1( rawCenters (i)._2, canopyCenters, t1)};
```

S123、采用Map操作计算所述向量数据集中的数据对象与全局Canopy中心点集合中的中心点之间的距离，当所述距离小于T2时，则对所述数据对象进行标记并将所述数据对象划分到与相应中心点对应的Canopy集合中，最终得到P个Canopy集合及其中心点数据；具体地，所述Canopy集合是指初始Canopy集合以及新生成的Canopy集合中的一个，其条件是与中心点对应，而该中心点是指与数据对象距离小于T2的中心点，所述T2为Canopy算法中的固定阈值，且所述T1大于T2。所述步骤S123在Spark平台上的实现代码为：canopyDos= dataObjects.map{x=>(x._1, canopy_2(x, canopyCenters, t2))}.collect().cache()。

[0027] 进一步，对划分到所述Canopy集合中的数据对象执行Cache操作，使所述数据对象存放在内存中，便于后面K-means聚类计算时可多次重复使用，而不需要每次都在分布式文件系统上读取，从而提高K-means的计算效率。

[0028] 本发明先采用Canopy算法对所述向量数据集中的数据对象进行预聚类计算，即先廉价地和粗糙地将数据对象划分到Canopy集合中，得到P个Canopy集合及其中心点。

[0029] 进一步，在所述步骤S130中，将所述P个Canopy集合的中心点赋值给K个Cluster的中心点，所述P=K，其在Spark平台上的实现代码为：clusterCenters= new Hashset (canopyCenters)。

[0030] 本发明通过上述步骤可使得K-means算法Cluster中的初始中心点不是随机产生的，而是根据初始Cluster的分布属性对中心节点进行优化，并且聚类算法的K值选取时可参考Canopy的个数P，从而有效解决K-means算法中聚类结果稳定性和准确性差的问题。

[0031] 在本发明所述步骤S140中，根据所述K值以及Cluster的中心点，通过Map操作对所述向量数据集进行K-means计算，输出聚类结果，具体包括：

S141、采用Map操作计算所述向量数据集中的数据对象与所述初始Cluster中心点的距离，将所述数据对象划入到与所述数据对象距离最短的Cluster中心点所在的Cluster，生成新的Cluster；所述步骤S141在Spark平台上的实现代码为：while(iterateDist> deltaDist){

```
Step11:mapDos= canopyDos. map {
do => (closestCenter(do, clusterCenters),(do, 1)) };
```

S142、采用reduceByKey操作计算所述新生成的Cluster中数据对象的平均值，作为新的Cluster中心点；所述步骤S142在Spark平台上的实现代码为：

```
newClustercenters = mapDos.reduceByKey {
case ((s1, c1), (s2, c2)) => (s1 + s2, c1 + 2)}.map{
case (index, (s, c)) => (index, s / c)}.collect();
```

S143、采用Map操作计算新生成的Cluster中心点与所述初始Cluster中心点的平方差，当所述平方差小于等于预定阈值时，输出聚类结果；当所述平方差大于预定阈值时，则返回步骤F1，直至输出聚类结果。所述步骤S143在Spark平台上的实现代码为：iterateDist=0.0

```
Step14: for ((index, value)← newClustercenters) {
iterateDist+= canopyCenters.get(index).get.squaredDist(value)
clusterCenters (index) = canopy_2(value, canopyCenters, t2))
}}。
```

[0032] 本发明通过对向量数据集执行Map操作计算数据对象和初始Cluster中心点的距离，将所述数据对象划入到距离最短的中心点所在的Cluster，进一步，对向量数据集做reduceByKey操作计算Cluster中数据对象的平均值，并作为新的Cluster中心点，接着，采用Map操作计算新生成的Cluster中心点与所述初始Cluster中心点的平方差，当所述平方差小于等于预定阈值时，则聚类结果收敛，更新Cluster中心点，并输出聚类结果。

[0033] 基于上述方法，本发明还提供一种基于Spark平台实现的并行化K-means改进系统较佳实施例，如图2所示，其包括：

读取模块100，用于通过Spark大数据平台读取存放在分布式文件系统上的数据对象集，生成弹性数据集RDD；

向量化模块110，用于采用Map操作将所述弹性数据集RDD转化为向量数据集；

Canopy计算模块120，用于对所述向量数据集进行Canopy计算，得到P个Canopy集合以及各个Canopy集合的中心点；

赋值模块130，用于将所述P个Canopy集合的中心点赋值给K个Cluster的中心点，所述P=K；

K-means计算模块140，用于根据所述K值以及Cluster的中心点，对所述向量数据集进行K-means计算，输出聚类结果。

[0034] 较佳地，所述的基于Spark平台实现的并行化K-means改进系统，其中，所述系统还包括：

初始化模块，配置Spark参数并初始化Spark环境。

[0035] 较佳地，所述的基于Spark平台实现的并行化K-means改进系统，其中，所述Canopy计算模块120具体包括：

新Canopy集合生成单元，用于采用Map操作计算所述向量数据集中的数据对象与初始Canopy集合中心点的距离，当所述距离均大于T1时，则所述数据对象生成新的Canopy集合，并产生新的Canopy中心点；

合并单元，通过合并所述初始Canopy集合以及新生成的Canopy集合生成全局Canopy集

合，以及通过合并所述初始Canopy集合中心点以及新生成的Canopy集合中心点生成全局Canopy中心点集合；

数据对象划分Canopy单元，用于采用Map操作计算所述向量数据集中的数据对象与全局Canopy中心点集合中的中心点之间的距离，当所述距离小于T2时，则对所述数据对象进行标记并将所述数据对象划分到与相应中心点对应的Canopy集合中，最终得到P个Canopy集合及其中心点数据。

较佳地，所述的基于Spark平台实现的并行化K-means改进系统，其中，所述数据对象划分Canopy单元还包括：

存储子单元，用于对划分到所述Canopy集合中的数据对象执行Cache操作，使所述数据对象存放在内存中。

[0036] 较佳地，所述的基于Spark平台实现的并行化K-means改进系统，其中，所述K-means计算模块140具体包括：

数据对象划分Cluster单元，用于采用Map操作计算所述向量数据集中的数据对象与所述初始Cluster中心点的距离，将所述数据对象划入到与所述数据对象距离最短的Cluster中心点所在的Cluster，生成新的Cluster；

新Cluster中心点生成单元，用于采用reduceByKey操作计算所述新生成的Cluster中数据对象的平均值，作为新的Cluster中心点；

聚类结果输出单元，用于采用Map操作计算新生成的Cluster中心点与所述初始Cluster中心点的平方差，当所述平方差小于等于预定阈值时，输出聚类结果；当所述平方差大于预定阈值时，则返回数据对象划分Cluster单元，直至输出聚类结果。

[0037] 关于上述模块单元的技术细节在前面的方法中已有详述，故不再赘述。

[0038] 综上所述，本发明先采用Canopy算法对大数据进行预聚类处理，粗糙地将数据划分到覆盖的Canopy集合中，使得K-means算法中的初始中心节点不是随机产生，并且K-means算法中的K值选取时可以参照Canopy集合的个数P，从而解决K-means算法聚类结果稳定性和准确性差的问题；并且本发明基于Spark平台将Canopy算法应用到K-means聚类算法前能高效地加速聚类操作，降低K-means算法中的迭代次数，从而有效提高改进的-means算法的执行效率，避免局部最优。

[0039] 应当理解的是，本发明的应用不限于上述的举例，对本领域普通技术人员来说，可以根据上述说明加以改进或变换，所有这些改进和变换都应属于本发明所附权利要求的保护范围。

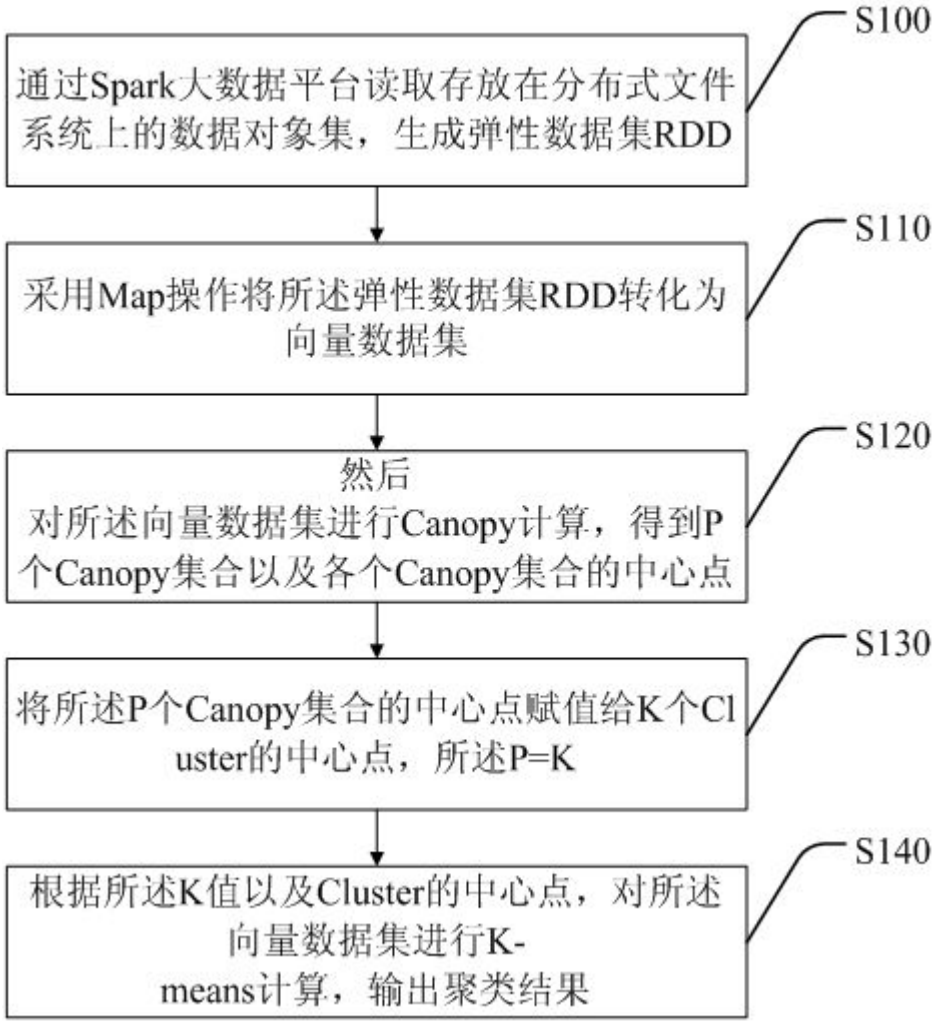

S100
通过Spark大数据平台读取存放在分布式文件系统上的数据对象集，生成弹性数据集RDD
S110
采用Map操作将所述弹性数据集RDD转化为向量数据集
S120
然后
对所述向量数据集进行Canopy计算，得到P个Canopy集合以及各个Canopy集合的中心点
S130
将所述P个Canopy集合的中心点赋值给K个Cluster的中心点，所述P=K
S140
根据所述K值以及Cluster的中心点，对所述向量数据集进行K-means计算，输出聚类结果

图1

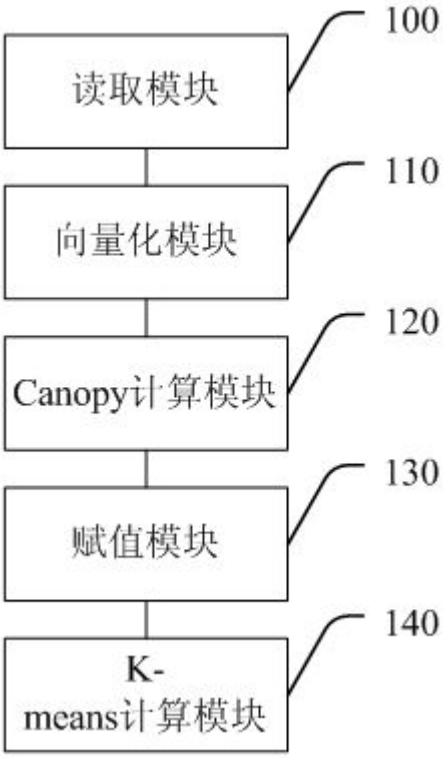


图2