基于 Spark 的 Canopy-FCM 在气象中的应用

勾志竟[1],宫志宏[2],徐 梅[1],刘布春[3]
(1. 天津市气象信息中心,天津 300074;
2. 天津市气候中心,天津 300074;
3. 中国农业科学院 农业环境与可持续发展研究所,北京 100081)

摘 要:随着气象事业现代化水平的不断提高,气象部门积累了海量的气象数据,如何从海量的气象数据中挖掘出有用的知识,是提高气象服务能力的关键所在。针对传统聚类算法无法有效处理海量数据的问题,提出了一种基于 Spark 框架的 Canopy-FCM(Canopy-fuzzy C-means)并行化聚类算法。该算法将 Canopy 算法与 FCM 算法相结合,避免了 FCM 算法对初始聚类中心敏感的问题,并结合 Spark 分布式框架内存计算的优势,大大降低了海量气象数据的处理时间。通过采用天津市 208 个区域自动气象站 4~10 月逐月降水观测数据,评估了天津市不同区域的降水情况。实验结果表明,提出的方法不仅可以快速有效地从气象数据中挖掘出有用的信息,同时与基于 Hadoop 框架下的算法相比,有更高的运行速率和加速比,也为相关部门有效地做出水旱灾害监测预警与风险防范决策提供了一种全新的思路和方法。

关键词:FCM;Canopy;Spark;气象;数据挖掘
中图分类号:TP39 **文献标识码**:A **文章编号**:1673-629X(2020)08-0169-05
doi:10.3969/j.issn.1673-629X.2020.08.029

Application of Canopy-FCM Algorithm Based on Spark in Meteorology

GOU Zhi-jing[1],GONG Zhi-hong[2],XU Mei[1],LIU Bu-chun[3]
(1. Tianjin Meteorological Information Center,Tianjin 300074,China;
2. Tianjin Climate Center,Tianjin 300074,China;
3. Institute of Environment and Sustainable Development in Agriculture,Chinese Academy of Agricultural Sciences,Beijing 100081,China)

Abstract:As the continuous improvement of the modernization level of meteorological service,a huge amount of meteorological data was accumulated in meteorological department. How to dig out useful knowledge from massive meteorological data is the key to improve the meteorological service ability. In view of the issues that traditional clustering algorithm cannot effectively processing massive data,the parallel Canopy-FCM (Canopy-fuzzy c-means) clustering algorithm based on the Spark framework is proposed. The algorithm combines Canopy algorithm with FCM algorithm,which can avoid the sensitivity of FCM algorithm to the initial clustering center. Combined with the advantages of memory calculation of Spark distributed framework,the processing time of massive meteorological data has been greatly reduced. Then,the precipitation situation of different regions has been evaluated through using the monthly precipitation observation data from April to October of 208 regional automatic weather stations in Tianjin. The experiment shows that the proposed method cannot only dig out useful information from meteorological data quickly and effectively,but also has higher running speed and acceleration ratio compared with the algorithm based on Hadoop framework. It also provides a new idea for related departments to make effectively decision on monitoring and early warning of flood and drought disasters and risk prevention.

Key words:FCM;Canopy;Spark;meteorology;data mining

0 引 言

随着科技的进步,气象部门获取数据的途径也越来越多,收集并产生的气象数据呈指数级增长。如何将数据挖掘技术应用到气象预报预测和气象灾害预测等方面[1-3],从海量的气象数据中挖掘有价值的信息,成为气象行业研究的重点。传统的数据处理方法已经不能很好地处理海量数据,挖掘数据内部规律时更为乏力,而数据挖掘算法与分布式处理框架[4]的出现为

收稿日期:2019-09-27 **修回日期**:2020-01-20
基金项目:国家重点研发项目(2017YFC1502800)
作者简介:勾志竟(1989-),男,硕士,工程师,研究方向为大数据、气象数据挖掘;宫志宏,通讯作者,硕士,高工,研究方向为农业气象信息化。

挖掘海量气象数据提供了一种新的思路。

陈正威[5]在 Hadoop 平台上运用预处理有向无环图和支持向量机(PDAG-SVM)算法对降水量做出预测,该方法在预测精度和预测效率上都取得了令人满意的结果;王昊等[6]提出了一种 Hadoop 平台下基于离散贝叶斯网络的数据挖掘改进算法,预测精度明显高于目前短期气候预测中采用的朴素贝叶斯算法;张晨阳等[7]提出基于 Hadoop 的计算等价类的数据约简算法与朴素贝叶斯分类算法,该并行数据挖掘方案可以有效处理海量气象数据,并具有良好的扩展性;Lv Zhenhua 等[8]提出了并行 K-means 算法,并用于遥感图像的分类;李莉等[9]基于 Spark 平台提出并行 K-means 算法对气候区进行划分,对气象领域研究有重要现实意义。

从目前的相关研究可以看出,学者们不断对海量数据挖掘方法进行研究和优化,而聚类算法作为数据挖掘的重要方法,将其与分布式处理框架相结合[10-12]处理海量数据成为数据挖掘领域越来越活跃的研究方向。文中提出了一种 Canopy-FCM 算法,可以有效避免模糊 C-均值聚类算法对初始聚类中心敏感的问题,同时针对海量气象数据,采用 Spark 内存计算分布式框架快速有效地从气象数据中挖掘出有用的信息,大大的提高了运行效率。

1 模糊 C 均值算法(FCM)

模糊 C 均值(fuzzy C-means,FCM)算法[13]是 1974 年由 Dunn 提出并由 Bezdek 推广的,它是基于模糊集合论,把聚类问题转化为非线性规划问题,并通过迭代求解。

令 $X=\{X_1,X_2,\cdots,X_n\}$ 为待分类样本,FCM 将其分为 c 个模糊组,使得目标函数值最小,目标函数如下所示:

$$J=\sum_{i=1}^{c}\sum_{j=1}^{n}u_{ij}^{m}\parallel x_j-C_i\parallel^2 \tag{1}$$

$$\sum_{i=1}^{c}u_{ij}=1,j=1,2,\cdots,n \tag{2}$$

其中,u_{ij} 是样本 j 属于类 i 的隶属度,C_i 为第 i 类的中心,$m\in[1,\infty]$ 为模糊因子。

通过式(2),采用拉格朗日乘数法构造以下目标函数:

$$\bar{J}=J+\sum_{j=1}^{n}\lambda_j\left(\sum_{i=1}^{c}u_{ij}-1\right) \tag{3}$$

对所有参数求导,得到使式(3)达到最小值的必要条件为:

$$C_i=\frac{\sum_{j=1}^{n}(u_{ij}^{m}x_j)}{\sum_{i=1}^{n}u_{ij}^{m}} \tag{4}$$

$$u_{ij}=\left[\sum_{k=1}^{c}\left(\frac{\parallel x_j-C_i\parallel}{\parallel x_j-C_k\parallel}\right)^{\frac{2}{m-1}}\right]^{-1} \tag{5}$$

由式(4)和式(5)可以知道,给定初始样本集合 X,以及分类数目 c 和模糊因子 m,FCM 算法按照以下步骤不断迭代就可以计算出隶属度矩阵 $\boldsymbol{U}$ 和聚类中心 C。

(1)用随机数生成器生成初始隶属度矩阵 $\boldsymbol{U}$,且满足约束条件式(2)。

(2)用式(4)更新聚类中心。

(3)用式(5)更新隶属度矩阵 $\boldsymbol{U}$。

(4)计算式(1)的目标函数值,如果小于阈值 ε,则算法停止,否则重复步骤(2)和(3)。

2 Canopy-FCM 算法设计

2.1 Canopy 算法

FCM 算法采用随机生成聚类中心的方法,但无法保证为每个分类找到较好的中心,而聚类中心直接影响算法的运行效率。针对初始中心敏感,容易陷入局部最优的问题,文中采用 Canopy 算法[14]初始化聚类中心。Canopy 算法可以很快得到最优的分类数,其具体步骤如下:

(1)给定样本 $X_1,X_2,\cdots,X_n$,设定初始阈值 T_1,T_2,$T_1>T_2$。

(2)在样本中随机挑选样本 x,计算 x 到其他样本点的距离 d。

(3)把 $d<T_1$ 的样本点划分为一个 Canopy,同时把 $d<T_2$ 的样本点从数据集移除。

(4)重复步骤 2、3,直到数据集为空。

2.2 Canopy-FCM 算法框架

Canopy-FCM 算法基本步骤如下:

Step1:利用 Canopy 算法生成初始聚类中心。

Step2:初始化隶属度矩阵 $\boldsymbol{U}$。

Step3:更新聚类中心 C。

Step4:更新隶属度矩阵 $\boldsymbol{U}$。

Step5:是否满足终止条件,若满足,则算法停止;否则,重复 Step3 和 Step4。

3 基于 Spark 的并行 Canopy-FCM 模型

3.1 Spark 计算模型

Spark 是基于内存计算的分布式计算框架,起源于加利福尼亚大学伯克利分校的实验室研究项目[15],其低延迟、低系统开销、容错性高、分布式数据结构以及强大的函数式编程接口可以很好应对迭代式计算应用的高性能需求,在大规模数据处理任务中有广泛的应用。

Spark 在分布式环境下采用主从结构模型,包括

Driver 和 Worker 节点,程序运行之前将数据存储在 Hadoop Distributed File System(HDFS)中,接着 Driver 会运行应用中的方法创建 SparkContext 以及 RDD,DAGScheduler 对象将每个 job 分成多个 Stage,并为每个 stage 创建 TaskSet,TaskScheduler 将 task 提交给 executor 执行,executor 调用 Taskrunner 封装 task,并行线程池中取一个线程执行 task。其架构如图 1 所示。

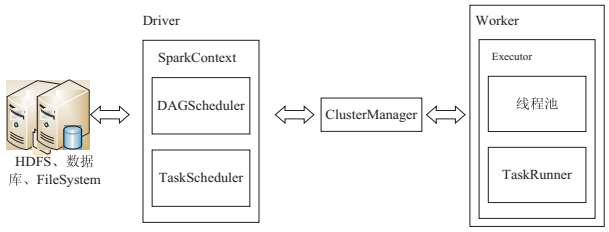

图 1　Spark 架构

3.2　Canopy-FCM 算法的并行化

基于 Spark 的 Canopy-FCM 算法流程如图 2 所示。

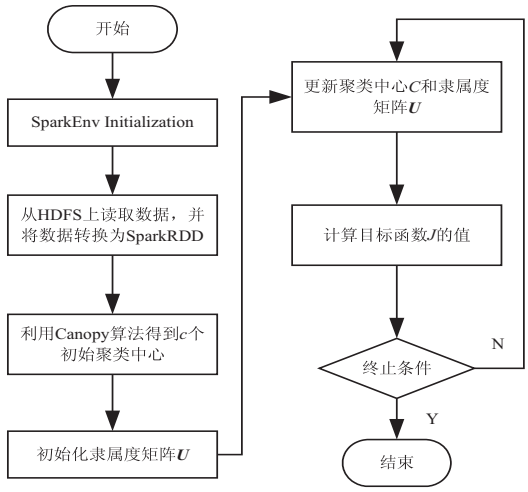

图 2　基于 Spark 的 Canopy-FCM 算法流程

(1)配置好 Spark 运行环境并初始化各参数。通过 hadoop fs -put 命令将数据上传到 HDFS 上,调用 SparkContext 的 sc. textFile( ) 方法将数据转换为 Spark-RDD,通过 map 操作转换为向量缓存到内存中。

(2)在各个子节点通过 map 操作计算数据集中每个点到 Canopy 中心点的欧氏距离,进而得到局部的 Canopy 中心点,然后通过 reduce 操作得到全局的 Canopy 中心点,将其作为 FCM 的初始聚类中心,并广播给各个子节点。

(3)在各个子节点通过 map 操作计算每个数据点到各中心的欧氏距离和隶属度,然后通过 reduceByKey( )和 collectAsMap( )方法得到各数据点到每个分类的距离之和与隶属度之和,对隶属度和聚类中心进行更新。

(4)计算目标函数的值,判断结果是否收敛,如果收敛则算法结束,通过 Combine 操作合并中间结果,并通过 Reduce 操作得到全局聚类中心,否则重复步骤(3)。

Canopy-FCM 算法并行化[16]的伪代码如下:

Input: $X = \{X_1, X_2, \cdots, X_n\}$, T_1 , T_2 , m , K

Output: $C = \{C_1, C_2, \cdots, C_c\}$

Initialization( );

GetInitialPoint(X , C_n);

while$_{1 \leqslant n \leqslant C}^{\max}(|C_n - C_n^{'}|^2) > \varepsilon$ and $k < K$ {

$l \leftarrow$ data. mapPartitions{ point $s \Rightarrow$

for $x_i \leftarrow$ point s {

$\mathrm{Sum}U^{'} += \sum_{j=1}^{n} u_{ij}^{m} x_j$

$\mathrm{Sum}UX^{'} += \sum_{i=1}^{n} u_{ij}^{m}$

}. reduce(merge)

$C^{'} \leftarrow C$; $C \leftarrow$ null

for $j = 1$ to C {

$(\mathrm{sum}UX, \mathrm{sum}U) \leftarrow l(j)$

$C += \mathrm{sum}UX / \mathrm{sum}U$

}

}

4　实例分析

4.1　实验环境与数据集

实验采用 Spark 分布式集群,集群搭建在服务器虚拟化平台上,选取 1 台机器作为主节点,其他 7 台机器作为工作节点。虚拟机各项配置及集群的配置信息分别如表 1、表 2 所示,实验数据采用天津经过质控后的 208 个区域自动气象站4 ~ 10月夏半年逐月降水观测数据。

表 1　虚拟机配置信息

名称	配置
CPU	Intel(R) Xeon(R) CPU E7- 4807
内存	32 G
硬盘	500 G
操作系统	CentOS 6.5
JDK	Jdk1.7.0_79
Hadoop	Hadoop-2.6.0
Hive	Hive-1.2.1
Scala	Scala-2.10.6
Spark	Spark-1.6.0

由表 2 可以看出 Spark 分布式集群在运行时需要一系列的后台程序,主要有:

Master-负责资源的调度(决定在哪些 Worker 上执行 executor)和监控 Worker。

Worker-负责执行任务的进程(executor),并将当前机器的信息通过心跳汇报给 Master。

NameNode–负责管理文件系统的 Namespace。

DataNode–负责管理各个存储节点。

SecondaryNameNode–NameNode 的热备,负责周期性地合并 Namespace image 和 Edit log。

表 2 集群信息配置

主机名	IP	进程
Master	10. xxx. xxx. 23	Master、Worker、NameNode、DataNode、JobHistoryServer、NodeManager
Slave1	10. xxx. xxx. 24	Worker、DataNode、NodeManager、ResourceManager、WebAppProxyServer
Slave2	10. xxx. xxx. 25	Worker、SecondaryNameNode、DataNode、NodeManager
Slave3	10. xxx. xxx. 26	Worker、DataNode、NodeManager
Slave4	10. xxx. xxx. 27	Worker、DataNode、NodeManager
Slave5	10. xxx. xxx. 28	Worker、DataNode、NodeManager
Slave6	10. xxx. xxx. 29	Worker、DataNode、NodeManager
Slave7	10. xxx. xxx. 30	Worker、DataNode、NodeManager

4.2 实验结果及分析

实验结果如图 3 所示,由图 3 可以看出天津 208 个区域自动气象站降水分布可分为 4 个区域,1 区主要集中在中部和北部区域,共有 96 个站;2 区集中在东部区域,共有 29 个站;3 区集中在东南部,共有 31 个站;4 区主要集中在西南部,共有 52 个站。

图 3 天津降水区划图

图 4 是实验得到的天津市 4 个分区降水量年平均分布图,由图 4 可以看出,4 个分区的降水主要集中在 6 ~ 9 月,7 月降水量最为显著,其次是 8 月、6 月、9 月,这一趋势与中国气象局气象数据中心发布的天津气候类型图(1981–2010)一致。4 个分区的具体分析如下:

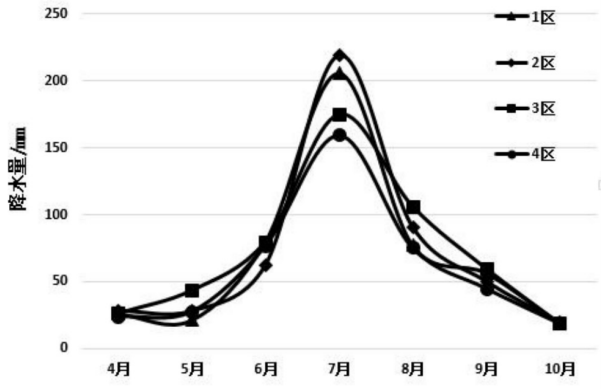

图 4 天津市 4 个分区降水量年平均分布

1 区主要位于天津中部和北部区域,该区域土壤以盐化潮土和粘质土为主,5 月年平均降水量远低于其他分区,4 ~ 10 月总年平均降水量 485. 7 mm。

2 区主要位于天津的东部区域,属于海积、冲积平原区,地势北高南低,4 月、7 月、10 月年平均降水量高于其他三个分区,6 月年平均降水量远低于其他分区,4 ~ 10 月总年平均降水量 498. 1 mm。

3 区主要位于天津市的东南沿海地区,地势低平,以海积低平原为主,土层受海潮影响盐渍化比较严重,5 月、8 月、9 月年平均降水量远高于其他三个分区,4 ~ 10 月总年平均降水量 508. 1 mm。

4 区主要位于天津的西南部,该区域以洼地冲积平原和滨海平原为主,地形平坦但多洼地,地势南高北低,西高东低,4 月、7 月、8 月、9 月及 10 月年平均降水量均低于其他分区,4 ~ 10 月总年平均降水量 425. 2 mm。

为了对比文中设计的 Spark 平台和 Hadoop 平台的集群性能,分别在 Hadoop 环境下和 Spark 环境下由单节点到 8 节点执行相同大小的区域自动站降水数据文件,得到两种环境下的加速比,如图 5 所示。

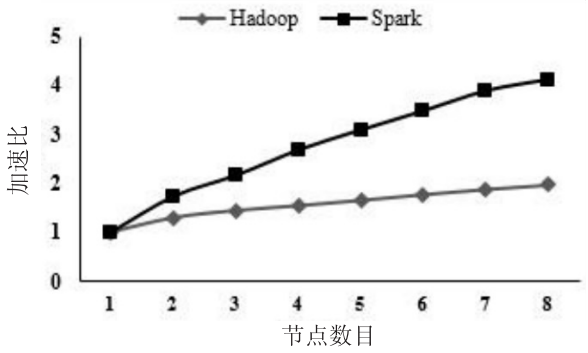

图 5　Hadoop 平台和 Spark 平台的加速比

由图5 不难看出,当节点数目为单节点时,Hadoop 平台和 Spark 平台的性能都处于最差。随着 DataNode 节点数量的增加,Spark 平台和 Hadoop 平台算法的运行时间都有不同程度缩短,而 Spark 平台的加速比要优于 Hadoop 平台,说明文中提出的算法在 Spark 平台下能有效地提高算法的性能,及时准确地挖掘出海量气象数据的有价值信息。

5　结束语

针对模糊 C-均值聚类算法对初始聚类中心敏感及因迭代计算次数增加导致内存不足的问题,设计了一种基于 Spark 框架的 Canopy-FCM 并行化聚类算法。该算法结合 Canopy 算法与模糊 C-均值聚类算法,避免了 FCM 算法对初始化敏感的问题,并结合 Spark 分布式框架内存计算的优势,大大降低了海量气象数据的处理时间。通过采用天津市 208 个区域自动气象站 4 ~ 10 月逐月降水观测数据,评估了天津市不同区域的降水情况。实验结果表明,提出的方法不仅可以快速有效地从气象数据中挖掘出有用的信息,同时还有良好的扩展性,能够为相关部门做好抗旱救灾、防灾救灾工作提供一种全新的思路和方法。但方法仅针对降水区进行了划分,未来可以结合温度、湿度、干燥度等因素做进一步的气候区划研究。

参考文献:

[1] 金　龙,吴建生,林开平,等. 基于遗传算法的神经网络短期气候预测模型[J]. 高原气象,2005,24(6):981-987.

[2] 张乐坚,程明虎,田付友. 人工神经网络及支持向量机在降雨量预报中的应用[J]. 高原气象,2010,29(4):982-991.

[3] 王　军,费　凯,程　勇. 基于改进的 Adaboost-BP 模型在降水中的预测[J]. 计算机应用,2017,37(9):2689-2693.

[4] 刘骥超,叶　钒,谢寒生. 云计算环境下气象大数据的应用研究[J]. 计算机技术与发展,2019,29(5):168-171.

[5] 陈正威. Hadoop 下基于 DAG-SVM 算法的降水量预测研究[D]. 南京:南京信息工程大学,2016.

[6] 王　昊,师　卫,李　欢. Hadoop 下基于贝叶斯网络的气象数据挖掘研究[J]. 电子器件,2016,39(4):841-846.

[7] 张晨阳,马志强,刘利民,等. Hadoop 下基于粗糙集与贝叶斯的气象数据挖掘研究[J]. 计算机应用与软件,2015,32(4):72-76.

[8] LV Zhenhua, HU Yingjie, ZHONG Haidong, et al. Parallel K-Meansclustering of remote sensing images based on MapReduce[C]//Web information systems and mining. Sanya: Springer,2010:162-170.

[9] 李　莉,王小刚. 基于 Spark 的并行 K-means 气象数据挖掘研究[J]. 信息技术,2017(9):26-30.

[10] 李淋淋,倪建成,曹　博,等. 基于 Spark 框架的并行聚类算法[J]. 计算机技术与发展,2017,27(5):97-101.

[11] 吴云龙,李玲娟. 基于 Spark 的模糊聚类算法实现及其应用[J]. 计算机技术与发展,2019,29(1):130-134.

[12] 徐鹏程,王　诚. K-Means 算法改进及基于 Spark 计算模型的实现[J]. 南京邮电大学学报:自然科学版,2017,37(4):113-118.

[13] GUSTAFSON D E, KESSEL W C. Fuzzy clustering with a fuzzy covariance matrix[C]//IEEE conference on decision and control including the 17th symposium on adaptive processes. San Diego: IEEE,1978:761-766.

[14] MCCALLUM A, NIGAM K, UNGAR L H. Efficient clustering of high—dimensional data sets with application to reference matching[C]//Proceedings of the sixth ACM SIGKDD international conference on knowledge discovery and data mining. Boston: ACM,2000:l69-178.

[15] 萨初日拉,周国亮,时　磊,等. Spark 环境下并行立方体计算方法[J]. 计算机应用,2016,36(2):348-352.

[16] 梁　鹏. 基于 Spark 的模糊 c 均值聚类算法研究[D]. 哈尔滨:哈尔滨工业大学,2014.