

引用格式: 潘淼鑫, 林甲祥, 陈崇成, 等. 基于C-SOM和Spark的并行空间离群挖掘方法及应用[J]. 地球信息科学学报, 2019, 21(1): 128-136. [Pan M X, Lin J X, Chen C C, et al. Parallel spatial outliers mining based on C-SOM and Spark[J]. Journal of Geo-information Science, 2019, 21(1): 128-136. ] DOI:10.12082/dqxxkx.2019.180221

# 基于C-SOM和Spark的并行空间离群挖掘方法及应用

潘淼鑫<sup>1,2,3</sup>, 林甲祥<sup>4</sup>, 陈崇成<sup>1\*</sup>, 叶晓燕<sup>1</sup>

1. 福州大学福建省空间信息工程研究中心空间数据挖掘与信息共享教育部重点实验室, 福州 350108; 2. 福建师范大学数学与信息学院, 福州 350117; 3. 福建省公共服务大数据挖掘与应用工程技术研究中心, 福州 350117; 4. 福建农林大学计算机与信息学院, 福州 350002

## Parallel Spatial Outliers Mining based on C-SOM and Spark

PAN Miaoxin<sup>1,2,3</sup>, LIN Jiexiang<sup>4</sup>, CHEN Chongcheng<sup>1\*</sup>, YE Xiaoyan<sup>1</sup>

1. Key Lab of Spatial Data Mining and Information Sharing of Ministry of Education, Spatial Information Research Center of Fujian, Fuzhou University, Fuzhou 350108, China; 2. College of Mathematics and Informatics, Fujian Normal University, Fuzhou 350117, China; 3. Fujian Provincial Engineering Technology Research Center for Public Service Big Data Mining and Application, Fuzhou 350117, China; 4. College of Computer and Information Sciences, Fujian Agriculture and Forestry University, Fuzhou 350002, China

**Abstract:** Spatial outlier mining can find the spatial objects whose non-spatial attribute values are significantly different from the values of their neighborhood. Faced with the explosion of spatial data and problems such as single machine performance bottleneck and difficult expansion, the traditional centralized processing mode has gradually failed to meet the needs of applications. In this paper, we propose a parallel spatial outlier mining algorithm and its prototype system which are based on Constrained Spatial Outlier Mining (C-SOM) and make full use of the advantages of a parallel computing framework Spark's fast memory computing and scalability. The parallel algorithm uses C-SOM algorithm as the core algorithm, executes the C-SOM algorithm on a Spark cluster composed of multiple nodes for a global dataset and many local datasets concurrently to get the global outliers and the local outliers. Datasets are divided into multiple regional datasets according to the administrative division. A region dataset is considered as a local dataset and the global dataset contains all of the selected local datasets to be mined. The lightweight prototype system implements the parallel algorithm based on Spark and adopts Browser/Server architecture to provide users with a visualized operation interface which is concise and practical. Users can select the region datasets and set the parameters of C-SOM algorithm on interfaces. The prototype system will execute the parallel algorithm on a Spark cluster and finally list both the global and local outliers which have the top largest outlier factor values so that users can make further analysis. At last, we use the soil geochemical investigation data from Fujian eastern coastal zone area in China and a series of artificial

收稿日期: 2018-05-03; 修回日期: 2018-07-03.

基金项目: 福建省重点科技计划项目(2015H0015); 福建省教育厅基金(JAT160125); 福建省社科青年项目(FJ2017C084)。

[ **Foundation items:** Key Science and Technology Plan Projects of Fujian Province, No.2015H0015; Fujian Provincial Education Department Foundation, No.JAT160125; Social Science Youth Projects of Fujian Province, No.FJ2017C084. ]

作者简介: 潘淼鑫(1987-), 女, 博士生, 主要从事大数据挖掘与云计算研究。E-mail: pan\_miaoxin@qq.com

\*通讯作者: 陈崇成(1968-), 男, 博士, 教授, 主要从事空间数据挖掘与知识网络、地学可视化与虚拟地理环境研究。

E-mail: chencc@fzu.edu.cn

datasets to carry out experiments. The results of the soil geochemical datasets experiments validate the rationality and effectiveness of the parallel algorithm and its prototype system. The results of the artificial datasets experiments show that, compared to single machine implementation, our parallel system can support analysis for much more datasets and its efficiency is much higher when the number of datasets is big enough. This study confirms the local instability characteristics of spatial outliers and demonstrates the rationality, and effectiveness of the parallel algorithm and its prototype system to detect global and local spatial outliers simultaneously.

**Key words:** C-SOM; Spark; parallel computing; spatial outlier; data mining

**\*Corresponding author:** CHEN Chongcheng, E-mail: chencc@fzu.edu.cn

**摘要:** 空间离群挖掘可以发现空间数据集中非空间属性值与邻域中其他空间对象明显不同的空间对象。随着空间数据量的快速增加,传统集中式处理模式面临单机性能瓶颈、难以扩展等问题,已逐渐不能满足应用需要。因此,本文根据 Spark 并行计算框架,充分利用 Spark 快速内存计算和扩展性的优势,提出了一种基于考虑约束条件的空间离群挖掘算法(C-SOM)和 Spark 的并行空间离群挖掘算法和原型系统。该并行算法以 C-SOM 为核心,并行地在多个计算节点对全局数据集和各局部数据集执行 C-SOM 算法,得到全局离群和局部离群。轻量级的原型系统基于 Spark 实现了该并行算法,采用 Browser/Server 架构,提供给用户可视化的操作界面,简洁实用。最后,通过福建省东南沿海土壤化学元素调查数据和人工合成数据的离群分析,验证了该并行算法和原型系统的合理性、有效性和高效性。

**关键词:** C-SOM; Spark; 并行计算; 空间离群; 数据挖掘

## 1 引言

空间离群是指那些非空间属性值与其空间邻域中其他对象的非空间属性值显著不同的空间对象,尽管它们相对整个样本来说可能并没有显著不同<sup>[1-2]</sup>。例如,基于非空间属性房龄,在一个正在发展的都市区中,一座被老房子包围的新房子是一个空间离群。空间离群挖掘可以发现数据集中预料外的、隐含的知识,被广泛应用于许多地理信息系统和空间数据库中,应用领域包括生态环境、交通管理、公共安全、公共健康、气候、基于位置的服务等<sup>[3]</sup>。

现有的空间离群挖掘算法大致可以分为 2 类,基于图的方法和定量测试方法<sup>[4]</sup>。基于图的方法在空间数据可视化的基础上,将空间离群突出显示出来,比较有代表性的算法有 variogram clouds<sup>[5]</sup>、pocket plots<sup>[6]</sup>、Moran scatterplot<sup>[1,4,7]</sup> 以及 Shekhar 等提出的针对图结构数据集的空间离群检测方法<sup>[8-9]</sup>。定量测试方法通过精确的测试来识别空间离群,又可以分为面向一维属性的方法和面向高维属性的方法<sup>[10]</sup>。面向一维属性的典型算法包括 Scatterplot<sup>[11]</sup>、Z 值、迭代 R 值、迭代 Z 值、迭代比率、中值、加权 Z 值<sup>[12]</sup>等。面向高维属性的方法沿用了许多面向一维属性方法的思想,主要算法有基于不同标度变量相异度的方法、基于马氏距离的多属性方法、基于相关系数的多属性方法、基于密度的方法等。由于传统的基于密度的离群挖掘算法已经相对成熟,局

部离群检测的效率和准确性也较高,因而被广泛使用于空间离群挖掘实践中。以上这些方法没有考虑实际生活中客观存在的约束条件(如河流、桥梁等)对空间离群挖掘结果的影响,空间数据挖掘过程中若不考虑客观存在的一些约束条件,挖掘结果往往会发生错误或者不合理的情况。文献[10]提出了考虑约束条件的空间离群挖掘算法(C-SOM),构建考虑约束条件的 Delaunay 三角网来表达空间邻近关系,采用密度思想定义空间对象的离群因子,通过离群因子“序值图”寻找合理离群数目,取得了较好的效果。

传统集中式数据挖掘面临内存限制、处理速度慢、硬盘容量不足等问题,随着人类收集的空间数据量的快速增加,并行和分布式计算已经成为大数据处理、分析过程中不可或缺的关键技术<sup>[13-14]</sup>,如网格计算<sup>[15]</sup>、云计算<sup>[16]</sup>、Hadoop<sup>[17]</sup>、MapReduce<sup>[18-19]</sup>、Spark<sup>[20-21]</sup>等等,通过将任务分解为可并发执行的多个子问题并在互联的多台节点上同时运行,突破了计算能力、存储能力等限制。学术界和工业界提出了许多分布式和并行数据挖掘算法。在离群挖掘的分布式和并行算法方面,文献[22]提出了一种高效的分布式离群点检测算法,文献[23]、[24]基于 MapReduce 实现离群数据的并行挖掘。针对空间离群挖掘的分布式和并行算法尚不多见,一些相关的典型研究工作有:文献[25]基于 Hadoop 来存储和检测时空数据,设计了分布式算法并发地进行离群

挖掘;文献[26]基于 Spark 实现了一种分布式条件下的空间离群点挖掘算法。但这些算法没有对局部数据集进行离群挖掘,由于空间离群具有局部不稳定特征,局部离群挖掘在许多实际应用中更有意义。文献[3]、[27]提出了基于网格的并行与分布式空间离群挖掘算法,同步地检测全局离群和局部离群。在众多的并行计算框架中,Spark 以其快速、高容错、高可扩展和易用的特点得到了广泛的应用。本文提出了一种基于 Spark 的并行空间离群挖掘方法,对全局数据集和各局部数据集,采用考虑约束条件的空间离群挖掘算法 C-SOM,进行并行离群挖掘。

## 2 Spark 并行计算框架

Spark 是加州大学伯克利分校的 AMP 实验室开发一种开源的基于内存的快速通用可扩展的数据分析引擎,既拥有 Hadoop MapReduce 所具有的优点,又做到了 MapReduce 所没能做到的。MapReduce 编程繁琐,速度慢。Spark 则易于编程,而且快捷灵活,它将中间结果保存在内存中,而不像 MapReduce 需要频繁读写 HDFS。与 MapReduce 相比,Spark 在迭代计算方面,速度比 MapReduce 快 1~2 个数量级<sup>[28]</sup>。这使得 Spark 可以更好地处理迭代计算较多的机器学习、图形处理等任务。各种大数据公司,如 Cloudera、MapR 等,都已表示要以 Spark 取代 MapReduce。

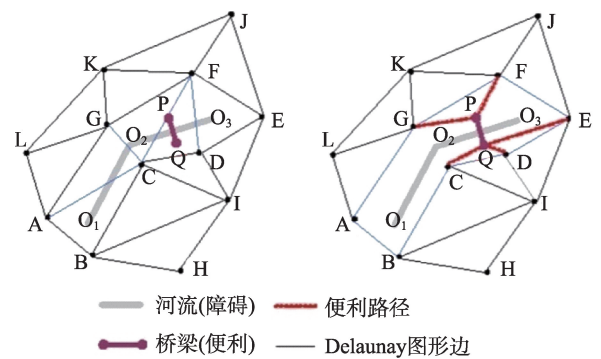
RDD(弹性分布式数据集)是 Spark 的编程基础,整个 Spark 生态系统中都是基于对 RDD 的操作完成的。RDD 是只读的分布式的数据集合,这个数据集被划分为多个分区并分散地存储在集群中的多个节点之上。RDD 的获得只能通过两种途径:①在内存集合中或外部存储系统,通过程序创建 RDD;②通过其他 RDD 的某种转换操作得到。RDD 上的操作可以分为二大类,即 Transformation(转换)和 Action(动作)。Transformation 是指从一种 RDD 转换为另一种 RDD,它是一种延迟操作,需要通过 Action 来触发。Action 操作是将 RDD 输出的操作,它会触发之前所有的 Transformation,并向 Spark 集群提交作业,同时将数据输出到 Spark 系统中。

## 3 C-SOM 算法

考虑约束条件的空间离群挖掘(C-SOM)将约束条件嵌入到空间数据挖掘过程中,使得对象之间的相似度受到邻近关系和约束条件的双重限制,因

而与传统的数据挖掘不同<sup>[10]</sup>。C-SOM 算法的主要过程是:首先,以四方边缘结构 QuadEdge 为核心数据结构,根据空间属性值构建考虑约束条件的 Delaunay 三角网来表达空间邻近关系,通过在 Delaunay 三角网中计算对象的  $k$  阶邻近来确定对象的空间邻域;然后,将属性约束嵌入到空间对象之间的属性距离计算之中,采用欧氏距离对空间对象之间的相似性度量进行计算,按照基于密度的思想定义一个衡量空间对象离群率的离群因子 OF (Outlier Factor),并通过对象和邻域的比较确定各个空间对象的离群因子值;最后,将离群因子最大的若干个空间对象作为候选离群,并对候选离群进行分析与确认,获得最终的空间离群挖掘结果。C-SOM 算法的具体设计和实现见文献[10]。

普通 Delaunay 三角网和考虑河流、桥梁约束的 Delaunay 三角图如图 1(a)和(b)所示。图 1(b)中  $O_1O_2$  和  $O_2O_3$  是河流表示障碍线段,边  $PQ$  为桥梁表示便利线段,边  $PF, PG, QC, QD, QE$  为便利路径,其余图形边为普通 Delaunay 边。约束 Delaunay 三角图的具体构建过程见文献[29]。



(a) 普通Delaunay三角网 (b) 考虑障碍和便利条件的Delaunay图

图1 普通Delaunay三角网和约束Delaunay图

Fig. 1 Common Delaunay triangulation and constrained Delaunay graph

为使约束 Delaunay 图的空间邻近关系定义与普通 Delaunay 三角网的空间邻近关系定义保持一致,对约束 Delaunay 图的 Delaunay 距离  $d'_r(A, B)$  定义如式(1)所示。

$$d'_r(A, B) = \begin{cases} d_r(A, B) = 1 & (\text{当 } A, B \in V_s \text{ 时}) \\ \frac{1}{2} & (\text{当 } A \in V_s \cap B \in V_{Facility} \text{ 或 } A \in V_{Facility} \cap B \in V_s \text{ 时}) \\ 0 & (\text{当 } A, B \in V_{Facility} \text{ 时}) \end{cases} \quad (1)$$

式中:  $V_s$  表示空间对象顶点集合;  $V_{Facility}$  表示空间便



利顶点集合。若 $A$ 与 $B$ 都是 $V_s$ 中的点且之间存在Delaunay图边,则与普通Delaunay三角网一样, $A$ 与 $B$ 的距离为1;若 $A$ 与 $B$ 其中一个点在 $V_s$ 中而另一个点在 $V_{Facility}$ 中,则 $A$ 与 $B$ 的距离为0.5;若 $A$ 与 $B$ 都是 $V_{Facility}$ 中的点,则 $A$ 与 $B$ 的距离为0。

对于给定的空间目标 $A'$ 和 $B'$ ,若 $A'$ 沿着Delaunay图边移动到 $B'$ 所经过的最少Delaunay边的距离(按Delaunay距离 $d'_T(A,B)$ 进行计算)之和为 $k$ ,则称这2个目标之间的Delaunay距离为 $k$ ,记为 $d'_T(A',B')=k$ 。所有与给定目标 $P$ 的Delaunay距离为 $k$ 的目标集合,称为目标 $P$ 的 $k$ 阶邻近,记为 $Neighbor_k(P)$ 。以图1中的顶点 $C$ 为例,其1阶邻近为 $Neighbor_1(C)=\{G, F, E, D, I, B\}$ 。这里讨论的目标 $A$ 和 $B$ 是 $V_s$ 中的两个对象,对于 $V_{Facility}$ 中的目标,通常不进行目标的 $k$ 阶邻近计算,涉及到便利顶点的Delaunay距离计算必须遵循 $d'_T(A,B)$ 的定义。

属性约束用于确定参与空间离群挖掘的专题属性及相应的权重。若 $A=\{A_1, A_2, \dots, A_m\}$ 为空间对象的 $m$  ( $m \geq 1$ )个专题属性,各属性的权重为 $\omega=\{\omega_1, \omega_2, \dots, \omega_m\}$ ,空间对象 $P_i$ 的邻域为 $NN(P_i)$ (邻居的个数为相应地表示为 $\|NN(P_i)\|$ ),则对象相异度和离群因子分别定义如式(2)–式(4)。对象离群因子的值越大,表示与邻域对象的差别越大,就可能是离群点。

定义1 空间对象 $P_i$ 与 $P_j$ 的相异度 $diff(P_i, P_j)$ :空间对象 $P_i$ 和 $P_j$ 在专题属性值上的差异,值等于空间对象 $P_i$ 和 $P_j$ 在专题属性值上的欧氏距离。

$$diff(P_i, P_j) = \sqrt{\sum_{k=1}^m \omega_k [A_k(P_i) - A_k(P_j)]^2} \quad (2)$$

定义2 空间对象 $P_i$ 与邻域 $NN(P_i)$ 相异度 $diff$

( $P_i$ ):空间对象 $P_i$ 与其邻域 $NN(P_i)$ 在专题属性值上的差异,值等于空间对象 $P_i$ 与其所有邻居对象的相异度平均值。

$$diff(P_i) = \frac{\sum_{O \in NN(P_i)} diff(P_i, O)}{\|NN(P_i)\|} \quad (3)$$

定义3 空间对象 $P_i$ 的离群因子 $OF(P_i)$ :空间对象 $P_i$ 的相异度与其邻居对象的相异度的比值平均:

$$OF(P_i) = \frac{\sum_{O \in NN(P_i)} \frac{diff(P_i)}{diff(O)}}{\|NN(P_i)\|} \quad (4)$$

## 4 基于C-SOM和Spark的并行离群挖掘算法

### 4.1 算法设计与实现

由于空间数据集的地理分布特征,以及空间离群的局部区域性特征,空间离群往往需要从局部和全局2个角度进行刻画,挖掘各局部数据集的空间离群点和全局数据集的空间离群点。因此并行空间离群挖掘中,既需要对局部数据集进行空间离群的挖掘,也需要从全局数据集的角度对局部不稳定特征进行挖掘。

以C-SOM算法为核心的基于Spark的并行空间离群挖掘的基本过程(图2)。(1)存储系统中保存着以行政区划为划分依据的各地区数据集,如地区1数据集表示福州市空间数据集,地区2数据集表示泉州市空间数据集,将待挖掘地区的空间数据集读入内存,一个地区数据集作为一个局部数据集,同时将这些地区数据集在内存中按地区合并,即一个地区的数据集后跟着另一个地区的数据集,最后

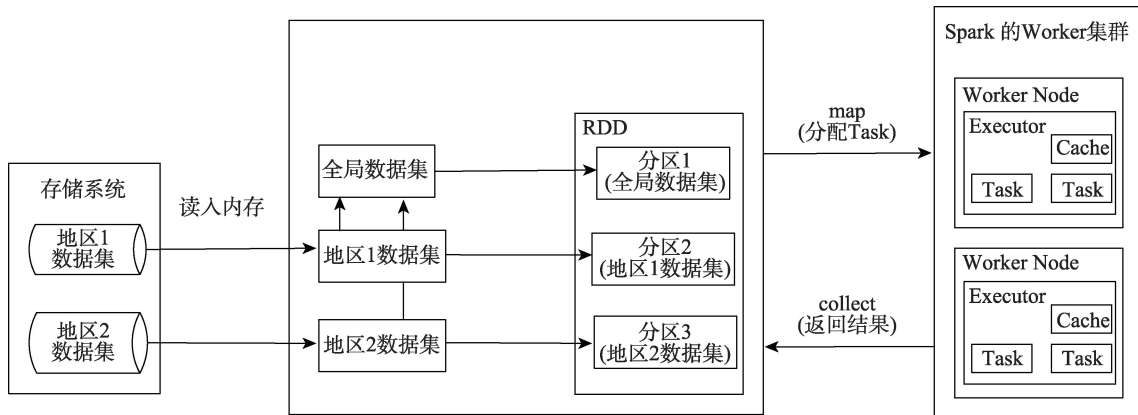


图2 基于Spark的并行空间离群挖掘的基本过程

Fig. 2 The general process of parallel spatial outlier mining based on Spark

生成包含所有局部数据集的全局数据集;②将全局数据集和各局部数据集都转成RDD,以便Spark进行处理,每个地区数据集作为一个分区,全局数据集单独作为一个分区;③Spark将各个分区的离群挖掘任务分配给Worker集群,图2中一个Task表示一个分区的挖掘任务,集群里的各个Worker Node(WN)并发地调用C-SOM算法执行分配到的Task;④各个WN再把每个Task的计算结果(离群点)返回给Spark的集群管理节点(Master)。

基于C-SOM和Spark的并行空间离群挖掘算法伪代码如下所示:

Algorithm 基于C-SOM和Spark的并行空间离群挖掘算法

**Input:**

待挖掘的多个数据集名称列表names,约束条件  
constraintMap,邻域阶数k,离群点个数outlierNum

**Output:**

每个地区和全局的数据集的离群点result,格式为  
Map<数据集名称,离群点集合>

```
1、 List<<地区名,该地区数据集>> data = read(names); //
将待挖掘数据集读入内存,得到各局部数据集,地区名作
为数据集名称
2、 total = 合并data中各个数据集; //total是全局数据集
3、 data.add(<“全局”,total>); //“全局”表示数据集名称
4、 初始化Spark;
5、 spatial = parallelize(data); //将data转成RDD并分区
6、 distributeResult = spatial.map(分区->C-SOM(分区数
据,constraintMap,k,outlierNum));
7、 result = distributeResult.collect(); //收集各个分区的计
算结果
return result.toMap();
```

算法首先根据用户提供的数据集名称列表names将数据读入内存,得到按行政区划分的各局部数据集列表data,再将它们合并得到全局数据集total。然后,全局数据集和各局部数据集通过Spark转成RDD,全局数据集作为一个分区,各局部数据集分别作为一个分区。接着,对各个分区的数据执行C-SOM算法:根据约束条件constraintMap得到分区数据的约束Delaunay三角图,根据k值和约束Delaunay三角图得到每个空间对象的邻域,基于此计算每个空间对象的离群因子,进行倒序排序后,得到离群因子最大的outlierNum个空间对象作为该数据集的离群点。最后将各个分区的离群点集中收集到Spark的Master,每个地区得到outlierNum个离群点,整个地区也得到outlierNum个离群点,全局离群点在局部地区离群挖掘结果中基本也是离群因子较大的离群点,将这些离群点都返回给用户

户,以供用户进行进一步分析。

## 4.2 算法的性能与效率分析

不妨假设,待挖掘的每个局部数据集含有m条空间数据,局部数据集的个数为n, $T(mn)$ 为全局数据集的计算时间, $T(m)$ 表示一个局部数据集的计算时间,则集中式挖掘的总时间 $T_1=T(mn)+n\times T(m)$ 。对于基于Spark的并行挖掘系统,首先要初始化Spark服务,然后将各个分区的数据和计算任务分配给对应的WN,各个WN结束某个计算任务后将计算结果传输给Spark的Master。假设Spark的初始化与调度时间为GS,数据的网络传输时间为Comm,离群点的计算时间为Comp,GS由Spark自身框架决定。Comm包括全局数据集、局部数据集和各数据集计算结果的网络传输,主要由数据规模 $m\times n$ 决定。假设WN的个数为d,即d台机器可同时执行挖掘任务,则Comp大致为 $T(mn)+(n/d)\times T(m)$ 。并行算法离群挖掘的总时间如下:

$$T_{\text{Spark}} = GS + Comm + Comp \\ = GS + Comm + T(mn) + (n/d) \times T(m) \quad (5)$$

要使 $T_{\text{Spark}} < T_1$ ,即 $GS + Comm + T(mn) + (n/d) \times T(m) < T(mn) + n \times T(m)$ ,则要求 $GS + Comm < (1 - 1/d) \times n \times T(m)$ ,从理论上来说,若Spark的初始化与调度时间以及网络通信时间之和小于并行挖掘带来的效率提高,则并行算法将花费更少的执行时间。此外,分布式的架构也突破了内存的限制,通过扩展WN,可以支持对更多的数据进行挖掘。

## 5 基于C-SOM和Spark的并行离群挖掘原型系统

为了帮助数据分析人员快速分析数据,构建Spark上的并行离群挖掘,本文设计了一个轻量级的原型系统。系统的架构图如图3所示,该系统基于Browser/Server(B/S)架构,B/S架构使客户端无需安装应用程序,打开浏览器,即可使用并行数据挖掘服务。系统的服务端分为表现层、业务逻辑层、数据层和Spark层。表现层为JSP页面,接收用户请求,并提交给业务逻辑层的Servlet。Servlet先从数据层读取待挖掘的空间数据集,然后将挖掘任务提交给Spark,由Spark负责对各数据集进行并行离群挖掘。Spark包括Master节点和多个Worker Node节点。Master作为Spark的集群管理节点,接

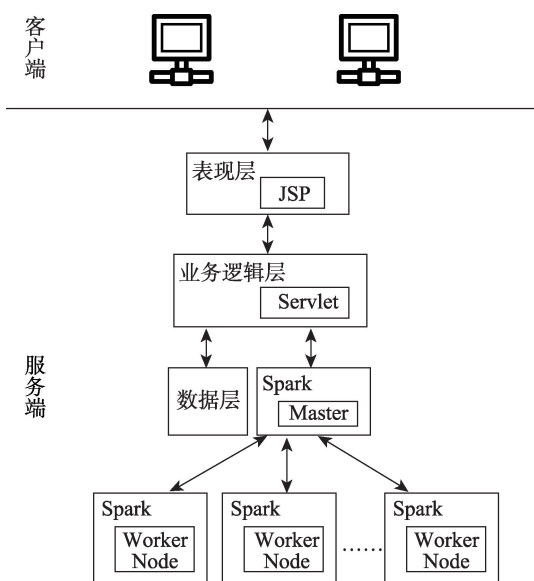
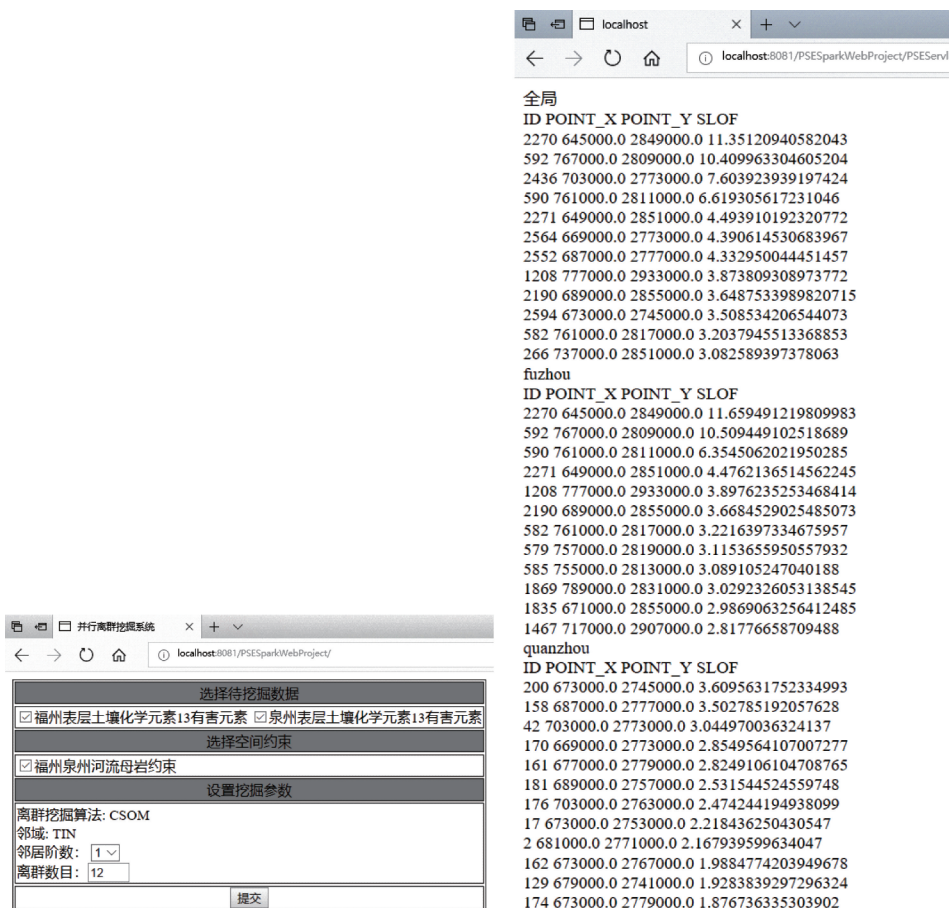


图3 基于Spark的并行空间离群挖掘原型系统架构  
Fig. 3 The architecture diagram of parallel spatial outlier mining prototype system based on Spark

收Servlet提交的挖掘任务并进行任务分解,得到多个子任务后分发给Worker Node执行,然后收集Worker Node的计算结果,再将结果返回给Servlet,Servlet再把结果通过表现层展现给用户。

系统的典型界面如图4所示。图4(a)中,用户在Web界面上选择要挖掘的数据集并设置考虑约束条件的空间离群挖掘算法参数,点击“提交”按钮即可开始执行并行空间离群挖掘。图4(b)显示了全局数据集和局部数据集的部分离群挖掘结果,其中“全局”下面的是整个地区数据集的离群挖掘结果,“fuzhou”下面的是福州地区数据集的离群挖掘结果,每个数据集返回了离群因子值最大的12个离群点,并按离群因子值的倒序排列,每一行代表一个离群点,包含了离群点的对象ID、横坐标、纵坐标和离群因子值。用户通过分析这些候选离群点及其对应属性值,结合实际情况,得到最终的离群点。



(a) 系统的数据集选择和参数设置界面

(b) 系统离群挖掘结果界面

图4 原型系统中数据选择、算法参数设置和结果界面

Fig. 4 Interfaces of data selecting, parameters setting and results in the prototype system



## 6 土壤数据离群挖掘实例分析

地球表层土壤的化学元素与人类的生存和发展息息相关。随着中国福建沿海地区工农业的快速发展,土壤污染问题越来越严重,对人类的健康构成严重威胁。土壤调查和恢复活动主要是通过采集原位样品,寻找元素浓度异常的位置来分析地球化学元素,特别是镉、汞、砷、Cu、铅、锌、铬、镍等重金属元素以及其他一些对人体健康有害的元素,它们的含量分布反映了人为二次污染的真实情况。最终检测出的土壤异常点将为以后对环境污染现状的分析提供科学依据。

### 6.1 实验区概况

本实例的土壤数据主要来自中国地质调查局开展的福建省沿海经济带生态地球化学调查项目,严格按照2 km×2 km的间距采样。实验区选取福建的福州市和泉州市这两个经济发达、人口稠密、高度城市化的地区,福州地区主要包含福州的五区八县,泉州地区涵盖惠安和石狮。土壤数据集共有61个属性,包括采样点的横坐标、纵坐标及样本标志(样品ID、分析编号、地市、县区、样品原号),以及54项土壤化学元素和指标。本实例将横坐标和纵坐标作为土壤数据的空间属性来确定采样点的空间邻近关系,专题属性选取13项有害元素(砷,银,铍,镉,铬,铜,汞,镍,铅,锑,硒,铊,锌)作为非空间属性进行异常分析,以探索人类活动对土壤二次污染的影响。

由于福建地处多山地区,大范围内成土母质对福建地区土壤的形成影响明显,因此本文考虑成土母质作为可能影响土壤化学元素值和指标的一个约束条件,对成土母质类型按“岩类”进行划分,采用折线段表示岩类分界线。除了成土母质类型外,河流和海域对区域的分割也会影响土壤分布。由于福州和泉州内陆水域面积较小,其对土壤分布的不连续影响较小。因此,本文只将较宽的河流和海域作为影响土壤化学元素含量的一个约束条件,采用折线段对这些约束条件进行建模与表达。

### 6.2 实验流程和结果分析

#### 6.2.1 土壤数据并行离群挖掘

以市级行政区划为划分依据,将实验区数据集划分为福州和泉州2个局部数据集,并由基于C-SOM和Spark的轻量级原型系统进行并行空间离群挖掘。在原型系统的Web界面中选择待挖掘的数据集“福州表层土壤化学元素13有害元素”和“泉州表层土壤化学元素13有害元素”,选择空间约束“福州泉州河流母岩约束”,采用1阶邻近为地理对象的空间邻域,并返回各地区离群因子最大的12个土壤采样异常点,如图4(a)所示。点击“提交”按钮后,即可显示整个实验区和2个地区(福州、泉州)离群因子最大的12个土壤采样异常点,如图4(b)所示,整理后的候选土壤数据离群如表1所示。

表1中,福州地区中对象ID为2270,592,590,2271,1208,2190,582和泉州地区中对象ID为200,158,42,170(对应于整个实验区中对象ID为2594,

表1 实验区并行离群挖掘结果

Tab.1 Results of the parallel outlier mining for the experimental area

序号	整个实验区		福州地区		泉州地区	
	对象ID(横坐标,纵坐标)	离群因子	对象ID(横坐标,纵坐标)	离群因子	对象ID(横坐标,纵坐标)	离群因子
1	2270(645 000, 2 849 000)	11.351	2270(645 000, 2 849 000)	11.659	200(673 000, 2 745 000)	3.610
2	592(767 000, 2 809 000)	10.410	592(767 000, 2 809 000)	10.509	158(687 000, 2 777 000)	3.503
3	2436(703 000, 2 773 000)	7.604	590(761 000, 2 811 000)	6.355	42(703 000, 2 773 000)	3.045
4	590(761 000, 2 811 000)	6.619	2271(649 000, 2 851 000)	4.476	170(669 000, 2 773 000)	2.855
5	2271(649 000, 2 851 000)	4.494	1208(777 000, 2 933 000)	3.898	161(677 000, 2 779 000)	2.825
6	2564(669 000, 2 773 000)	4.391	2190(689 000, 2 855 000)	3.668	181(689 000, 2 757 000)	2.532
7	2552(687 000, 2 777 000)	4.333	582(761 000, 2 817 000)	3.222	176(703 000, 2 763 000)	2.474
8	1208(777 000, 2 933 000)	3.874	579(757 000, 2 819 000)	3.115	17(673 000, 2 753 000)	2.218
9	2190(689 000, 2 855 000)	3.649	585(755 000, 2 813 000)	3.089	2(681 000, 2 771 000)	2.168
10	2594(673 000, 2 745 000)	3.509	1869(789 000, 2 831 000)	3.029	162(673 000, 2 767 000)	1.988
11	582(761 000, 2 817 000)	3.204	1835(671 000, 2 855 000)	2.987	129(679 000, 2 741 000)	1.928
12	266(737 000, 2 851 000)	3.083	1467(717 000, 2 907 000)	2.818	174(673 000, 2 779 000)	1.877

2552, 2436, 2564的采样点)在整个实验区空间离群挖掘时也被检测为离群,全局离群点中只有对象ID为266的采样点没有出现在各局部离群中。可以看出,全局离群点在局部离群挖掘结果中基本也是离群因子较大的离群点,但并不是各局部离群点的简单加成,二者在离群点位置和顺序上存在少量差异。这在一定程度上证实了空间离群的局部不稳定特征和将上述土壤采样点作为候选离群点的合理性,同时也验证了基于C-SOM和Spark的原型系统能够有效地进行并行空间离群挖掘。

#### 6.2.2 并行算法性能与效率的验证分析

本实验对单机上执行集中式离群挖掘和多机上执行并行离群挖掘的性能和效率进行对比分析。实验环境是在1台惠普台式机(Intel Core i7-6700 四核 3.40 GHz CPU, 16 GB 内存)上搭建4台配置相同的虚拟机,每个虚拟机都是单核单线程,2 GB 内存。单机实验在其中1台虚拟机上进行。并行实验在4台虚拟机组成的Spark集群中进行,其中1台作为Master,其余3台作为Worker Node。随机生成 $n$ 个局部数据集,第 $i$ 个局部数据集位于起点为 $(i \times 10\ 000, i \times 10\ 000)$ ,长、宽都为10 000的方形区域内。每个局部数据集包含3000条空间数据,每条空间数据包含横坐标、纵坐标以及13个非空间属性。单机集中式离群挖掘和基于Spark集群的并行离群挖掘的时间效率如图5所示。

由图5可知:①当局部数据集个数 $n$ 为90时,单机系统就出现了内存溢出,而并行系统直到150个局部数据集仍然运作正常,这说明并行系统能够支持更大的数据量的离群挖掘;②当 $n$ 比较小时,并行系统花费的时间多于单机系统,但随着 $n$ 的增加,并行系统时间开销增加的幅度小于单机系统,当 $n$ 达到80时,并行系统的时间开销已经小于单机系统,

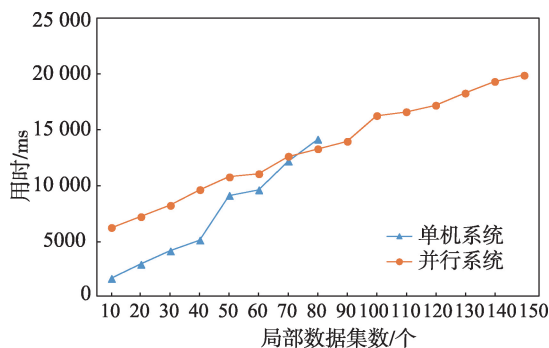


图5 单机实现和并行系统的离群挖掘效率对比

Fig. 5 Comparison of outlier mining efficiency between single machine implementation and parallel system

这主要是由于基于Spark的并行系统存在一定的初始化与调度时间以及网络通信时间,当 $n$ 比较小时,并行挖掘带来的效率提高比不上这些额外增加的时间开销,但随着 $n$ 的增加,这些额外的时间开销增长缓慢,而并行挖掘带来的效率提高越来越明显。可以看出,当 $n$ 足够大时,并行系统的离群挖掘效率优于单机系统。

## 7 结论

本文设计了基于考虑约束条件的空间离群挖掘算法C-SOM和并行计算框架Spark的并行空间离群挖掘算法及其原型系统,充分利用了Spark的快速内存计算和扩展性的优势。并行算法以C-SOM为核心,在多个计算节点并发地对分配到的数据集执行C-SOM算法以挖掘其离群点,再将各计算节点得到的离群点汇聚起来一并提供给用户进行分析。轻量级的原型系统基于Spark实现了该并行算法,并采用B/S架构,提供简单实用的可视化界面以方便用户进行数据分析。

通过福建省沿海的土壤化学调查数据13项有害元素含量的异常检测与分析,验证了基于C-SOM和Spark的并行空间离群挖掘算法及其原型系统的正确性和有效性。通过单机系统和并行系统进行人工合成数据空间离群挖掘的对比测试,证明并行系统能够有效地提高挖掘效率,当数据集个数足够大时,并行系统的时间开销小于单机系统。此外,由于Spark框架的扩展性,并行系统突破了单机内存的限制,能够支持对更多的数据进行挖掘。

#### 参考文献(References):

- [1] Shekhar S, Lu C T, Zhang P. A unified approach to detecting spatial outliers[J]. *GeoInformatica*, 2003,7(2):139-166.
- [2] Singh A K, Lalitha S. A novel spatial outlier detection technique[J]. *Communications in Statistics: Theory and Methods*, 2018,47(1):247-257.
- [3] Chen C C, Lin J X, Wu X Z, et al. Parallel and distributed spatial outlier mining in grid: Algorithm, design and application[J]. *Journal of Grid Computing*, 2015,13(2):139-157.
- [4] Lu C T, Chen D, Kou Y. Algorithms for spatial outlier detection[C]. Melbourne: Proceeding of 3<sup>rd</sup> IEEE International Conference on Data Mining, 2003.
- [5] Haslett J, Brandley R, Craig P, et al. Dynamic graphics for exploring spatial data with application to location global and local anomalies[J]. *The American Statistician*,



- 1991,45(3):234-242.
- [6] Pannatier Y. Variowin: Software for spatial data analysis in 2D[J]. *Statistics & Computing*, 1996,11(7):531-534.
- [7] Anselin L. Local indicators of spatial association: LISA[J]. *Geographical Analysis*, 1995,27(2):93-115.
- [8] Shekhar S, Lu C T, Zhang P. Detecting graph-based spatial outliers[J]. *Intelligent Data Analysis*, 2002,6(5):451-468.
- [9] Shekhar S, Lu C T, Zhang P. Detecting graph-based spatial outliers: Algorithms and applications (a summary of results) [C]. San Francisco: Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2001.
- [10] 林甲祥. 考虑约束条件的分布式空间离群挖掘及其应用研究[D]. 福州: 福州大学, 2010. [ Lin J X. Research on distributed spatial outlier mining in the presence of constraints and its applications[D]. Fuzhou: Fuzhou University, 2010. ]
- [11] Anselin L. Exploratory spatial data analysis and geographic information systems[J]. *New Tools for Spatial Analysis*, 1994,17:45-54.
- [12] Kou Y, Lu C T, Chen B. Spatial weighted outlier detection [C]. Philadelphia: Proceedings of the 6<sup>th</sup> SIAM International Conference on Data Mining, 2006.
- [13] Tsai C F, Lin W C, Ke S W. Big data mining with parallel computing: A comparison of distributed and MapReduce methodologies[J]. *Journal of Systems and Software*, 2016, 122:83-92.
- [14] Gan W S, Lin J C W, Chao H C, et al. Data mining in distributed environment: A survey[J]. *WIREs Data Mining and Knowledge Discovery*, 2017,7(6):e1216.
- [15] Luo P, Lu K, Shi Z Z, et al. Distributed data mining in grid computing environments[J]. *Future Generation Computer Systems*, 2007,23(1):84-91.
- [16] Gkatzikis L, Koutsopoulos I. Migrate or not? Exploiting dynamic task migration in mobile cloud computing systems[J]. *IEEE Wireless Communication*, 2013,20(7):24-32.
- [17] Apache. Hadoop[EB/OL]. <http://hadoop.apache.org/>, 2018-03-26.
- [18] Dean J, Ghemawat S. Mapreduce: Simplified data processing on large clusters[J]. *Communications of the ACM*, 2008,51(1):107-113.
- [19] 邬群勇, 苏克云, 邹智杰. 基于 MapReduce 的海量公交乘客 OD 并行推算方法[J]. *地球信息科学学报*, 2018,20(5): 647-655. [ Wu Q Y, Su K Y, Zou Z J. A mapreduce-based method for parallel calculation of bus passenger origin and destination from massive transit data[J]. *Journal of Geo-information Science*, 2018,20(5):647-655. ]
- [20] Apache. Spark[EB/OL]. <http://spark.apache.org/>, 2018-04-05.
- [21] 景维鹏, 霍帅起. 基于自定义 RDD 的海量遥感图像并行镶嵌方法[J]. *地球信息科学学报*, 2017,19(10):1346-1354. [ Jing W P, Huo S Q. A model of parallel mosaicking for massive remote sensing images based on self-defined RDD[J]. *Journal of Geo-information Science*, 2017,19(10):1346-1354. ]
- [22] 王习特, 申德荣, 白梅, 等. BOD: 一种高效的分布式离群点检测算法[J]. *计算机学报*, 2016,39(1):36-51. [ Wang X T, Shen D R, Bai M, et al. BOD: An efficient algorithm for distributed outlier detection[J]. *Chinese Journal of Computers*, 2016,39(1):36-51. ]
- [23] 张继福, 李永红, 秦啸, 等. 基于 MapReduce 与相关子空间的局部离群数据挖掘算法[J]. *软件学报*, 2015,26(5): 1079-1095. [ Zhang J F, Li Y H, Qin X, et al. Related-subspace-based local outlier detection algorithm using mapreduce[J]. *Journal of Software*, 2015,26(5):1079-1095. ]
- [24] 任燕. 基于 MapReduce 与距离的离群数据并行挖掘算法[J]. *计算机系统应用*, 2018,27(2):151-156. [ Ren Y. Parallel mining of distance-based outliers using mapreduce[J]. *Computer Systems & Applications*, 2018,27(2):151-156. ]
- [25] Yu D, Ping L, Li W. Spatio-temporal outlier detection based on cloud computing[J]. *Journal of Computational Information Systems*, 2014,10(13):5481-5488.
- [26] 张卫平, 刘纪平, 仇阿根, 等. 一种分布式计算的空间离群点挖掘算法[J]. *测绘科学*, 2017,42(8):85-90. [ Zhang W P, Liu J P, Chou A G, et al. A spatial outlier mining algorithm based on distributed computing[J]. *Science of Surveying and Mapping*, 2017,42(8):85-90. ]
- [27] 姚明经, 林甲祥, 陈崇成, 等. 网格环境下分布式空间离群挖掘体系的设计与应用[J]. *地球信息科学学报*, 2011,13(3):383-390. [ Yao M J, Lin J X, Chen C C, et al. Service and application of grid based distributed spatial outliers mining[J]. *Journal of Geo-information Science*, 2011, 13(3):383-390. ]
- [28] Zaharia M, Chowdhury M, Das T, et al. Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing[C]. Proceedings of the 9<sup>th</sup> USENIX Conference on Networked Systems Design and Implementation. San Jose, USA, 2012.
- [29] Lin J X, Chen C C, Wu J W. CD-graph: Planar graph representation for spatial adjacency and neighbourhood relation with constraints[J]. *International Journal of Geographical Information Science*, 2013,27(10):1902-1923.