DOI:10.13203/j.whugis20180473 文章编号:1671-8860(2020)02-0289-07

基于Spark计算框架的路网核密度估计并行算法

郭宇达[1] 朱欣焰[1,2,3] 吴 维[1,2] 佘 冰[4]

1 武汉大学测绘遥感信息工程国家重点实验室,湖北 武汉,430079
2 地球空间信息技术协同创新中心,湖北 武汉,430079
3 武汉大学空天信息安全与可信计算教育部重点实验室,湖北 武汉,430079
4 密西根大学社会研究院,密西根 安娜堡,48106

摘 要:路网核密度估计是路网约束下针对事件点的聚类分析方法,常用于研究交通事故、城市犯罪、车辆轨迹等事件的空间分布模式。传统单机串行的路网核密度估计算法在小数据量条件下的运行效率较高,但随着数据量的增加,算法性能显著下降,无法满足实际应用需求。针对路网核密度估计中的道路网分割和核密度计算,设计并实现了基于Spark计算框架的高效并行算法。以交通事故为例,通过4组实验进行对比分析。结果表明,基于Spark计算框架的路网核密度估计并行算法具有较高的运算效率,并具备良好的可拓展性。

关键词:路网约束;核密度分析;Spark并行计算;空间聚类;事故分析

中图分类号:P208 **文献标志码**:A

为了减少城市交通事故,缓解交通拥堵状况,提高居民出行安全性,在地理信息系统中经常使用事件点的空间分布模式进行相关研究[1-2],通过识别空间中的聚集现象来反映事件发生的热点区域,从而针对性地作出分析和预警。核密度估计方法能够有效分析事件点的空间分布模式[3]。传统的平面空间核密度估计以二维平面为基础,对事件的空间分布模式进行研究,但将其应用到明显沿道路网分布的事件时,会生成错误的聚集模式[4]。因此,在研究与路网密切相关的空间分布模式时,有必要采用路网约束条件下的核密度估计方法[5-7]。相较于传统的平面核密度估计,路网约束条件下的核密度估计需要引入路网拓扑关系[8],计算量显著增加,单机算法的性能难以令人满意[9]。

随着现代化建设的快速发展,车辆使用越来越普遍,城市路网越来越密集,可供决策支持的数据呈井喷式增长。以往的单机算法受到单机内存容量和计算能力的限制,在性能与稳定性方面难以满足要求。因此,越来越多的研究者尝试将并行计算技术应用于大数据空间分析领域,如李坚等[10]基于并行多核处理器架构处理上十亿点云数据;Wang等[11]利用Hadoop的MapReduce框架,提高了大数据条件下的空间叠置分析性能;Yoo等[12]基于MapReduce框架实现了空间自相关分析并行算法。本文基于大数据计算框架Apache Spark,设计并实现了路网的核密度估计并行算法,该算法在大数据量的条件下仍然能够保持较高的性能。

1 路网约束条件下的核密度估计并行算法

1.1 路网约束条件下的核密度估计

平面核密度估计方法以空间上某点为圆心,计算半径为r的圆形范围内事件发生的概率密度。计算公式如下:

$$\lambda(s)=\sum_{i=1}^{n}\frac{1}{\pi r^2}k\left(\frac{d_{is}}{r}\right) \tag{1}$$

式中,$\lambda(s)$为位置s处的密度;d_{is}为事件点i和事件

收稿日期:2019-03-15

项目资助:国家重点研发计划(2018YFB0505500,2018YFB0505503);国家自然科学基金(41830645);航天科技联合基金(4201420100041);深圳海事局项目(108-611710661);中央高校基本科研业务费专项资金。

第一作者:郭宇达,硕士生,主要研究方向为空间大数据分析。fakeguoyd@gmail.com

通讯作者:朱欣焰,博士,教授。xinyanzhu@whu.edu.cn

点s之间的欧氏距离;$k(\cdot)$为核函数(本文中取高斯核函数);r为带宽;n为与s相距r范围内的事件点总数。

路网约束条件下的核密度估计方法是将道路网分解为等长的线性单元(linear pixel,Lixel),以线性单元为最小研究单位,统计距离该线性单元最近的事件点总数,使用最短路网距离代替平面核密度估计方法中的欧氏距离进行计算[7],算法基本步骤如下:

1)将原有的道路网转化为基于路段的线性参考系统,相邻两条道路交叉点之间的线段称为路段,如图1中的AF。

2)将所有路段按照预设单位长度分为等长的线性单元,如图1中的l_1至l_7,不足一个单位长度的也记为一个线性单元。

3)针对每个线性单元,遍历事件点集合,将每个事件点分配给距离其最近的线性单元,统计每个线性单元上的事件点总数。

4)对于每个线性单元,将其带宽范围内的其余线性单元视为邻居,通过如下公式计算其核密度:

$$\lambda(s)=\sum_{i=1}^{n}c_i\frac{1}{r}k(\frac{d_{is}}{r}) \tag{2}$$

式中,d_{is}为线性单元i和线性单元s之间的最短路网距离;c_i为线性单元i上的事件点总数。

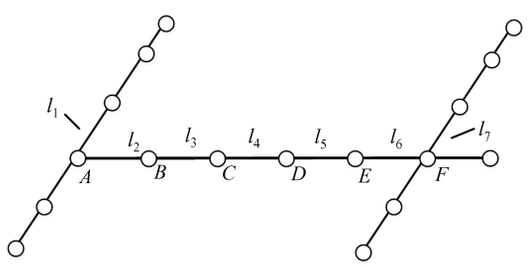

图1 路段示意图

Fig.1 Illustration of Road Segment

考虑到每条道路有不同的道路行政等级,如主干道、次干道、县道、乡道等,不同等级道路上的事件造成的影响程度不同,因此将式(2)重新定义为:

$$\lambda(s)=\sum_{i=1}^{n}w_ic_i\frac{1}{r}k(\frac{d_{is}}{r}) \tag{3}$$

式中,w_i为线性单元i的权重系数。在本文中,权重系数由线性单元所在道路的行政等级确定,主干道权重系数为1.6,次干道权重系数为1.4,县道权重系数为1.2,乡道和其他未明确定义行政等级的道路权重系数为1,权重系数可由用户根据实际应用需求进行自定义。

在基于路网的空间分析中,传统的二维核密度估计方法会产生估计偏差,从而导致错误的聚类结果[9]。因此本文在式(3)的基础上使用Okabe等[9]提出的无偏核函数——等分裂核函数进行密度估计。等分裂核函数如下:

$$K_y(x)=\begin{cases}\dfrac{k(d(x,y))}{(n_1-1)(n_2-1)\cdots(n_s-1)}, & 0\leqslant d(x,y)\leqslant r\\ 0, & d(x,y)>r\end{cases} \tag{4}$$

式中,$K_y(x)$为线性单元l_y与线性单元l_x之间的等分裂无偏核函数;$d(x,y)$为线性单元l_y与线性单元l_x之间的最短路网距离。定义节点的出度为该节点邻接的线性单元数量,如图1中节点A的出度为3,节点F的出度为4,若l_y与l_x之间仅仅通过唯一一个节点相连,则该节点的出度为2,如图1中的节点B、C、D、E。令n_i为线性单元l_y到线性单元l_x之间第i个节点的出度,如图1中,l_1与l_7的最短路网距离之间共有A、B、C、D、E、F共6个节点,n_1为节点A的出度(即3),n_2、n_3、n_4、n_5分别为B、C、D、E的出度(即2),n_6为节点F的出度(即4)。

1.2 Spark并行计算模型

Apache Spark(以下简称Spark)是专为大规模数据处理设计的快速通用的内存计算引擎。Spark中的所有操作在内存充足的情况下均在内存中执行,效率是Hadoop的10倍至100倍[13]。

弹性分布式数据集(resilient distributed dataset,RDD)是Spark中的基本数据抽象,具有自动容错、位置感知调度及可伸缩性等特点。Partition是数据集的基本组成单位,对于单个RDD,每个Partition会被当作一个计算任务进行处理,同时Partition的大小决定了并行计算的粒度。Spark提供了上层应用接口,接受多种数据格式作为输入,内部自动将程序并行化,并负责向集群提交应用和进行一系列的分区、转换、执行等操作,将结果以RDD的方式输出。

1.3 算法设计

基于Spark计算框架的路网核密度估计并行算法的主要思想为:各个excutor节点加载一部分计算数据,通过Spark的broadcast机制,从driver节点将所有excutor节点都使用到的公用数据进行广播,各个excutor节点并行完成各自计算任务,充分利用多核中央处理器(central processing unit,CPU)的计算性能,最后输出计算结果。本

文算法将道路网在几何上抽象为单线进行处理，算法整体流程如图2所示。

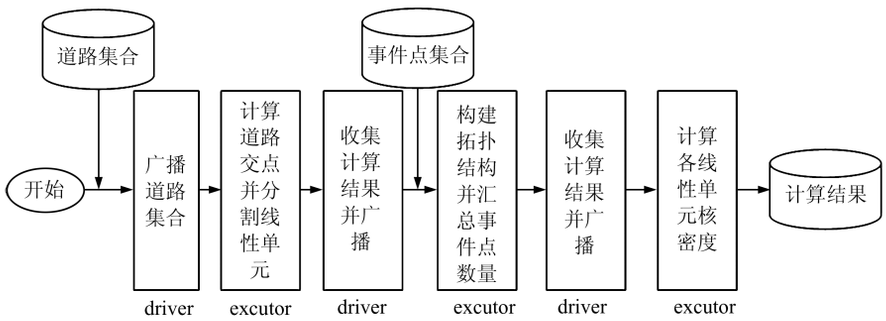

图2　路网核密度估计并行算法流程图

Fig.2　Flowchart of the Parallel Algorithm for Road Network Kernel Density Estimation

其中,单条道路的数据结构LineInfo如下:

LineInfo{

int id;//道路编号

double weight;//该道路的权重系数

int partitionNum;//分区编号

LineString line;//道路(包含坐标信息)

List<Tuple2<Integer, Coordinate>> crossPoints;//记录与其他道路的交点

List<LineString> segments;//有序线性单元集合

Map<Integer, List<Tuple2<Integer, Integer>>> intersections;//路网索引结构

}

算法具体步骤为:

1)将整个路网集合记为S,对S中的每条道路给定唯一编号,记为$L_i(i=1,2\cdots n)$,其中唯一编号对应LineInfo中的id。

2)将步骤1)中的路网集合S进行广播,各个excutor节点并行计算任意两条道路之间的交点,并记录相交道路的编号以及交点的坐标信息,该信息记录在LineInfo的crossPoints中。由于一条道路可能与其他多条道路相交,因此采用List存储,二元组Tuple的键为道路编号,值为交点坐标。

3)获取每条道路的起始点、拐点坐标,构建坐标序列,将交点坐标插入到构建好的坐标序列中。

4)以起止点、交点为分割点,将每条道路分为多条路段,再将各路段按照预设的线性单元长度分割为多个线性单元,不足一个线性单元长度的按一个线性单元处理,将分割完成的线性单元按顺序存入segments中。

5)根据步骤2)中记录的交点坐标信息及步骤4)中的线性单元分段信息构建路网拓扑结构,用于标识道路L_i的第k段线性单元l_{ik}与另一条道路L_j的第t段线性单元l_{jt}之间的邻接关系;在构建路网结构时,对segments中的各个线性单元的首尾节点进行编号,第n个线性单元的首节点编号为$n-1$,尾节点编号为n;针对每个线性单元,结合crossPoints信息和节点编号信息,即可完成路网拓扑结构intersections的构建;intersections中Map的键为当前LineInfo的节点编号,值为与该节点相交的其他LineInfo集合,二元组Tuple的键为LineInfo的编号,值为segments的节点编号。

6)计算每个事件点与每条道路的距离,选择与事件点距离最短的道路L_i,再计算该事件点与L_i上每个线性单元的距离,与该事件点距离最短的线性单元的计数值加1,如果最短距离超过阈值范围,则此事件点为无效点,不予考虑。

7)将步骤6)中的结果进行广播,在每个excutor节点中通过广度优先遍历方式并依据式(3)、式(4)计算每条道路L_i上各段线性单元的核密度值。其中,广度优先遍历的层数通过带宽与线性单元长度的比值来确定。

2　实验结果及算法分析

2.1　实验数据及实验环境

本文以武汉市2006—2017年的机动车简易事故数据和武汉市路网数据为实验对象,设计了4组实验,分析武汉市机动车简易事故在路网约束条件下的空间分布模式及其影响因素。其中,机动车简易事故数据包含532 414条记录,每条记录包含事故发生的经纬度;武汉市路网数据包含1 995条道路信息,总长度为5 283 km。实验采用4台相同的物理机搭建Spark集群,物理机的内存为32 GB,CPU为4核2.50 GHz,硬盘存储

为1 TB；软件配置为CentOS7.2，Hadoop2.7.3，Spark2.2.0。

2.2 实验结果与分析

2.2.1 线性单元长度和带宽长度对计算时长的影响

本实验采用1+4（1个driver节点，4个excutor节点）Spark集群进行核密度值的计算，其中driver节点配置1个CPU核，2 GB内存，每个excutor节点配置3个CPU核，6 GB内存。第1组实验固定线性单元长度为50 m，带宽长度为50～1 000 m；第2组实验固定线性单元长度为20 m，带宽长度为20～400 m。执行时间分别如图3、图4所示。

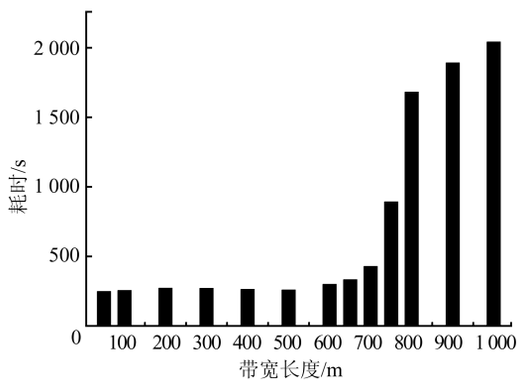

图3 线性单元长度为50 m时的不同带宽长度计算耗时

Fig.3 Calculation Time of Different Bandwidth Lengths When the Lixel Length is 50 m

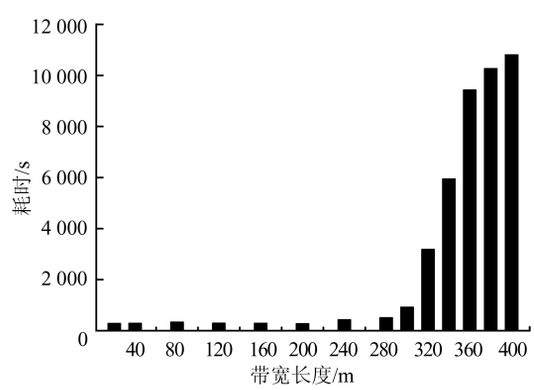

图4 线性单元长度为20 m时的不同带宽长度计算耗时

Fig.4 Calculation Time of Different Bandwidth Lengths When the Lixel Length is 20 m

对于线性单元为50 m时，全武汉市路网共切分为108 864个线性单元；当线性单元为20 m时，全武汉市路网共切分为267 250个线性单元。定义ratio=带宽/线性单元，表示计算单个线性单元的核密度时需要广度优先遍历的邻居层数。

从图3、图4可以明显看出，在不同线性单元长度条件下，计算核密度耗时的增长趋势基本一致，且线性单元长度越小，整体耗时越久。当ratio小于15时（带宽长度分别对应图3中的750 m、图4中的300 m），耗时较少，当ratio大于等于15以后，耗时迅速增加。这是因为当ratio较小时，计算时间大多用于分配事件点到最近的线性单元，而用于计算核密度值的耗时较少；但随着ratio每增加1，对应的每个线性单元在广度优先遍历时需要多遍历一层，遍历的邻居数量越多，计算耗时也越久。在对计算效率有较高要求的场景中，建议将ratio控制在15以内，当ratio值设置过大时，计算耗时过久，难以满足实际应用需求。

2.2.2 带宽长度对空间分布模式的影响

固定线性单元长度为50 m，带宽长度分别为50 m、200 m、500 m、1 500 m，局部路网核密度值可视化效果如图5所示。可以看出，随着带宽的增加，核密度值的起伏越来越小，整体呈现越来越平滑的趋势。这是由于随着带宽的增加，各个线性单元的邻居总数以及邻居覆盖范围越来越大，将更大范围内的事件点计算在内，导致相邻的线性单元之间差异逐渐缩小，因此整体呈现越来越平滑的趋势。

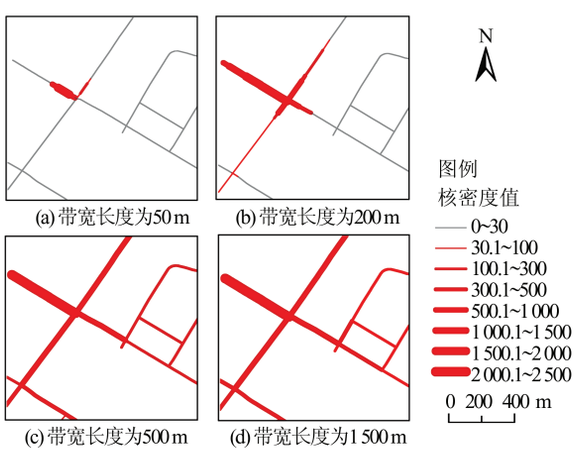

图5 线性单元长度为50 m时不同带宽长度对应的核密度局部图

Fig.5 Kernel Density Local Maps of Different Bandwidth Lengths When the Lixel Length is 50 m

图6展示了全局尺度上带宽长度的增加对于空间分布模式的影响。对于较小的带宽长度，其适合发现事件的局部热点区域，能够精准地定位到真正的事件多发路段；而对于较大的带宽长度，其适合发现更大空间尺度上的热点区域，事件点数量较大的线性单元会辐射到更远的区域，同时也能够揭示事件的高发路段影响区域。这是因为对于较小的带宽长度，计算核密度时只考虑了距离相近的线性单元，线性单元之间的相互影响仅局限在小范围内，因此适合发现局部热点

区域;而对于较大的带宽长度,各个线性单元的邻居范围更大,将多个局部的热点区域合并为更大尺度上的热点区域,适合于大尺度分析。

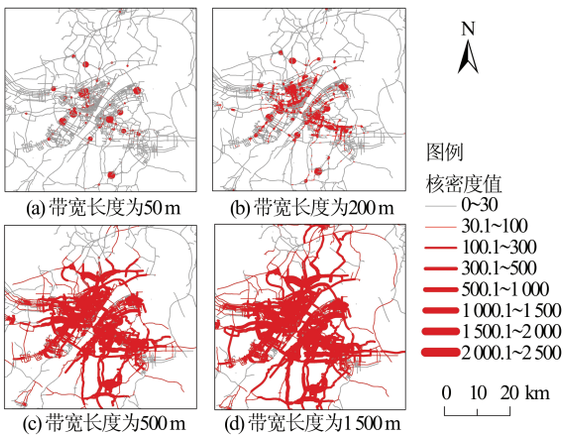

图6　线性单元长度为50 m时不同带宽长度对应的核密度全局图

Fig.6　Kernel Density Global Maps of Different Bandwidth Lengths When the Lixel Length is 50 m

2.2.3　线性单元长度对空间分布模式的影响

固定带宽长度为100 m,线性单元长度分别为10 m、20 m、50 m、100 m,局部路网核密度值可视化效果如图7所示。可以看出,在相同的带宽长度条件下,随着线性单元长度的增加,线性单元核密度值的局部细节逐渐丢失。这是因为当线性单元长度较小时,路网被分割为更多的线性单元,这些线性单元的核密度值彼此存在差异,使得局部细节表现丰富;而当线性单元长度较大时,等同于将若干个原本较小的线性单元作为一个整体进行计算,核密度值将在更大的尺度上呈现均一性,从而不可避免地丢失了局部细节。

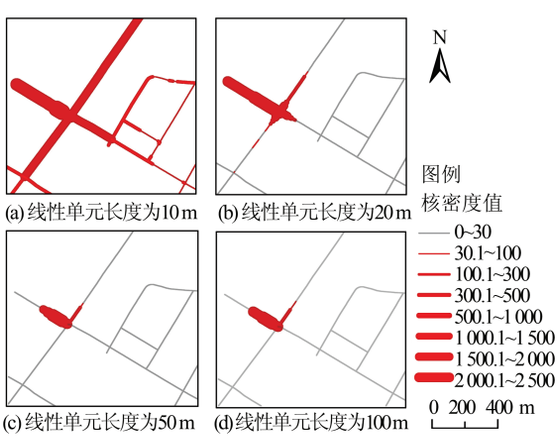

图7　带宽长度为100 m时不同线性单元长度对应的核密度局部细节图

Fig.7　Kernel Density Local Detail Maps of Different Lixel Lengths When the Bandwidth Length is 100 m

2.3　算法分析

并行算法的性能可以通过加速比和并行效率来衡量[14]。本文设计了4组实验对比分析不同CPU核数和不同计算规模条件下的加速比和并行效率。其中,CPU核数配置分别为1核1节点(1核)、2核1节点(2核)、2核2节点(4核)、2核3节点(6核)、2核4节点(8核)。固定线性单元长度为50 m,实验1至实验4的带宽长度分别为200 m(E_1)、400 m(E_2)、600 m(E_3)、800 m(E_4)。

2.3.1　加速比分析

加速比是指通过增加计算单元(即CPU核)所获得的性能提升,定义如下:

$$S(n,p)=\frac{T^*(n)}{T(n,p)} \tag{5}$$

式中,$T(n,p)$为当计算规模为n、CPU核数为p时的并行算法计算耗时;$T^*(n)$为在相同计算规模条件下的串行算法计算耗时;$S(n,p)$为加速比,取值范围为$[1,p]$。加速比越大,表明并行计算的相对耗时越少,算法性能越高。经过多次实验,算法的平均加速比曲线如图8所示。随着CPU核数的增加,算法性能整体上呈现提升的趋势,但随着CPU核数的增多,算法性能的提升速率有所下降。这是由于Spark并行算法的时间不仅由参与运算的CPU核数以及计算规模决定,而且与集群间的通讯消耗有关;随着参与运算节点的增多,Spark需要在多个节点之间进行任务调度以及数据的分发与收集,节点越多,通讯消耗越大。

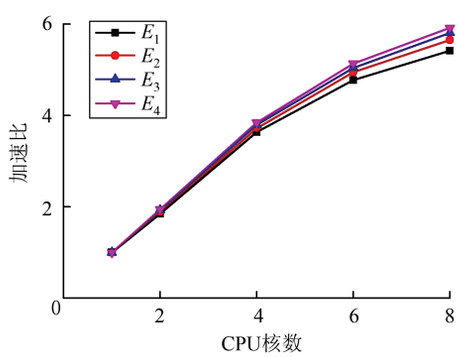

图8　不同CPU核数的加速比曲线

Fig.8　Acceleration Ratio Curves for Different CPU Cores

2.3.2　可拓展性分析

当集群中的计算节点不断增多时,节点间的通讯消耗也在不断增加,可以通过计算并行效率来衡量集群的实际利用率,以评价并行算法的可拓展性。并行效率定义如下:

$$E(n,p)=\frac{S(n,p)}{p}=\frac{T^{*}(n)}{p\cdot T(n,p)} \tag{6}$$

式中，$E(n,p)$为并行效率，取值范围为[0,1]。并行效率越高，表明通讯消耗越少，实际用于计算的时间越长，集群实际利用率越高，可以通过简单地增加计算节点来获得较大的性能提升，即算法的可拓展性越强。算法的并行效率曲线如图9所示。从横轴方向看，随着计算节点的增多，由于集群节点间通讯消耗的存在，并行效率总体呈现下降的趋势。从纵轴方向看，随着计算规模的增大，并行效率有所提高，这是由于计算规模增大，算法运行总时间更长，通讯消耗的时间相对较少所导致。总体而言，基于Spark计算框架的路网核密度估计并行算法在2核4节点的配置条件下仍能保持67.5%以上的并行效率，在2核3节点的配置条件下，并行效率可达79.5%以上，算法具备良好的可拓展性。

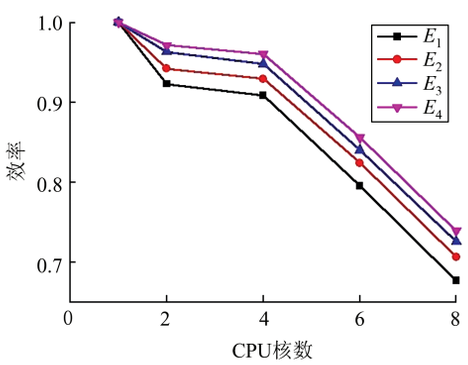

图9 不同CPU核数的效率曲线

Fig.9 Efficiency Curves for Different CPU Cores

3 结 语

本文提出了基于Spark计算框架的路网核密度估计并行算法，对原有的单机核密度估计算法进行了大幅优化，并且在Spark计算框架的基础上完成了算法的并行化，极大地提高了计算效率，能够在大数据量的条件下保持较好的性能，算法拥有良好的加速比和可拓展性，为大数据的空间分析提供了一种可行方案，在交通事故分析、缓解交通拥堵状况等方面有一定的应用价值。但本文针对计算任务的负载均衡只考虑了路网长度，并没有考虑路网复杂度的影响，因此在最终计算结果时仍然存在节点等待的情况，在一定程度上影响了算法性能。后续将进一步研究如何定量分析路网复杂度对算法性能的影响，并针对计算任务进行均衡分配，以进一步提升算法性能。

参 考 文 献

[1] Anderson T K. Kernel Density Estimation and *K*-Means Clustering to Profile Road Accident Hotspots [J]. *Accident Analysis & Prevention*, 2009, 41(3): 359-364

[2] Fu Zisheng, Li Qiuping, Liu Lin, et al. Identification of Urban Network Congested Segments Using GPS Trajectories Double-Clustering Method [J]. *Geomatics and Information Science of Wuhan University*, 2017, 42(9): 1 264-1 270(付子圣，李秋萍，柳林，等. 利用GPS轨迹二次聚类方法进行道路拥堵精细化识别[J]. 武汉大学学报·信息科学版，2017，42(9)：1 264-1 270)

[3] Yu Wenhao, Ai Tinghua, Yang Min, et al. Detecting "Hot Spots" of Facility POIs Based on Kernel Density Estimation and Spatial Autocorrelation Technique [J]. *Geomatics and Information Science of Wuhan University*, 2016, 41(2): 221-227(禹文豪，艾廷华，杨敏，等. 利用核密度与空间自相关进行城市设施兴趣点分布热点探测[J]. 武汉大学学报·信息科学版，2016，41(2)：221-227)

[4] Lu Y, Chen X. On the False Alarm of Planar *K*-Function When Analyzing Urban Crime Distributed Along Streets[J]. *Social Science Research*, 2007, 36(2): 611-632

[5] She Bing, Zhu Xinyan, Su Kehua, et al. Test Methods for Space-Time Interaction of Events Under Road Network Constraits[J]. *Geomatics and Information Science of Wuhan University*, 2015, 40(3): 353-356(佘冰，朱欣焰，苏科华，等. 道路网约束下的事件时空交互检验方法研究[J]. 武汉大学学报·信息科学版，2015，40(3)：353-356)

[6] She Bing, Zhu Xinyan, Ye Xinyue, et al. Weighted Network Voronoi Diagrams for Local Spatial Analysis[J]. *Computers Environment & Urban Systems*, 2015, 52: 70-80

[7] Xie Z, Yan J. Kernel Density Estimation of Traffic Accidents in a Network Space[J]. *Computers, Environment and Urban Systems*, 2008, 32(5): 396-406

[8] Borruso G. Network Density Estimation: A GIS Approach for Analysing Point Patterns in a Network Space[J]. *Transactions in GIS*, 2008, 12(3): 377-402

[9] Okabe A, Satoh T, Sugihara K. A Kernel Density Estimation Method for Networks, Its Computational Method and a GIS-Based Tool [J]. *International Journal of Geographical Information Science*, 2009, 23(1): 7-32

[10] Li Jian, Li Deren, Shao Zhenfeng. A Streaming Data Delaunay Triangulation Algorithm Based on Parallel

Computing[J]. *Geomatics and Information Science of Wuhan University*, 2013, 38(7): 794-798(李坚, 李德仁, 邵振峰. 一种并行计算的流数据Delaunay构网算法[J]. 武汉大学学报·信息科学版, 2013, 38(7): 794-798)

[11] Wang Y, Liu Z, Liao H, et al. Improving the Performance of GIS Polygon Overlay Computation with MapReduce for Spatial Big Data Processing[J]. *Cluster Computing*, 2015, 18(2): 507-516

[12] Yoo J S, Boulware D, Kimmey D. A Parallel Spatial Co-location Mining Algorithm Based on MapReduce[C]. 2014 IEEE International Congress on Big Data, Anchorage, America, 2014

[13] Zaharia M, Chowdhury M, Das T, et al. Resilient Distributed Datasets: A Fault-tolerant Abstraction for In-memory Cluster Computing[C]. The 9th USENIX Conference on Networked Systems Design and Implementation, San Jose, America, 2012

[14] Moreland K, Oldfield R. Formal Metrics for Large-Scale Parallel Performance[C]. The 30th International Conference on High Performance Computing, Frankfurt, Germany, 2015

Parallel Algorithm for Road Network Kernel Density Estimation Based on Spark Computing Framework

GUO Yuda[1]　*ZHU Xinyan*[1,2,3]　*GUO Wei*[1,2]　*SHE Bing*[4]

1　State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan 430079, China

2　Collaborative Innovation Center of Geospatial Technology, Wuhan 430079, China

3　Key Laboratory of Aerospace Information Security and Trusted Computing, Ministry of Education, Wuhan University, Wuhan 430079, China

4　Institute for Social Research, University of Michigan, Ann Arbor 48106, USA

Abstract: Road network kernel density estimation is a cluster analysis method for event points under road network constraints. It is often used to study spatial distribution patterns of traffic accidents, urban crimes, vehicle trajectories and other events. The traditional serial algorithm of road network kernel density estimation has higher efficiency under the condition of small data volume, but with the increase of data volume, the performance of the algorithm is significantly reduced, which can not meet the actual application requirements. In this paper, an efficient parallel algorithm based on Spark computing framework is designed and implemented for road network segmentation and kernel density calculation in road network kernel density estimation method. Taking traffic accidents as an example, four groups of experiments are used for comparative analysis. The results show that the parallel algorithm of road network kernel density estimation based on Spark computing framework has high computational efficiency and good extensibility.

Key words: road network constraints; kernel density analysis; Spark parallel computing; spatial clustering; accident analysis

First author: GUO Yuda, postgraduate, specializes in spatial big data analysis. E-mail:fakeguoyd@gmail.com

Corresponding author: ZHU Xinyan, PhD, professor. E-mail: xinyanzhu@whu.edu.cn

Foundation support: The National Key Research and Development Program of China (2018YFB0505500, 2018YFB0505503); the National Natural Science Foundation of China (41830645); Joint Fund for Aerospace Science and Technology (4201420100041); Project for Shenzhen Marine Bureau (108-611710661); the Fundamental Research Funds for the Central Universities.

引文格式:GUO Yuda, ZHU Xinyan, GUO Wei, et al.Parallel Algorithm for Road Network Kernel Density Estimation Based on Spark Computing Framework[J].Geomatics and Information Science of Wuhan University, 2020, 45(2):289-295.DOI:10.13203/j.whugis20180473(郭宇达, 朱欣焰, 吴维, 等. 基于Spark计算框架的路网核密度估计并行算法[J]. 武汉大学学报·信息科学版, 2020, 45(2):289-295.DOI:10.13203/j.whugis20180473)