大数据是近几年广受关注的一个概念, 现有的基于大数据挖掘城市出租车乘客的出行特征的研究基本基于 Hadoop 这个大数据处理工具, 但是 Hadoop MapReduce 在编写复杂的程序时, 往往要编写多个 MapReduce 任务, 这些任务提交到集群上之后都是独立运行的, 产生的数据不能保存在内存上, 会出现内存和磁盘之间数据不断的转移, 导致程序运行效率低下, 运行时间长, 磁盘和内存占用过大等缺点, 而 Spark 则完美解决这个问题, Spark 不仅继承 Hadoop MapReduce 的优点, 而且在此基础上进行扩充, 相对于 Hadoop MapReduce, Spark 更适合应用如机器学习、图计算、图操作、流式处理等操作, 它把数据加载到内存, 并且可以保存在内存, 通过内存进行计算, 减少磁盘 I/O, 减少数据的移动, 大幅度提高程序的性能, 正因为如此, 经检验, Spark 的性能超出 Hadoop 100 倍以上[2]. Spark 这个大数据处理工具以其对海量数据进行处理的优越性能也越来越受到人们的青睐. 目前, 利用 Spark 平台处理海量出租车数据的研究相对较少, 参考资料也相对较匮乏[3].

随着交通发展和车载 GPS 设备的迅速普及, 每天出租车通过车载 GPS 数据都在源源不断的产生数据[4]. 以西安市为例, 全市约有 1 万辆出租车, GPS 数据每隔 30s 产生一次, 一周的数据量超过 2 亿条. 如何从如此大的数据量中挖掘出有用的交通信息, 是交通大数据分析面临的一个重要问题. 本文基于 Spark 平台提出了 3 个算法用来挖掘出租车乘客出行特征信息. 实验首先对数据进行预处理, 包括过滤异常和错误数据; 而后, 使用计算每小时运行车辆数算法、提取出租车行驶距离算法、计算每小时实载率算法分别挖掘出租车出行距离、使用时间和计算每小时实载率出租车使用需求信息. 实验结果表明实验提出 3 个基于 Spark 平台的挖掘算法能均能在 90 分钟内完成对一周数据的统计分析.

2　数据处理平台

Spark 是 AMP lab 提出的类似于 Hadoop MapReduce 的开源大数据处理和计算框架, 可以在流处理、迭代计算、批处理等不同场景下使用. Spark 提出了弹性分布式数据集的概念(Resilient Distributed Dataset)简称 RDD. 每个 RDD 都被分为多个分区, 这些分区运行在集群中的不同节点上. 一般数据操作分成 3 个步骤: 创建 RDD、转化已有的 RDD 以及调用 RDD 操作进行求值. Spark 数据分析流程如图 1 所示, 运行程序时, Spark 会自动把 RDD 发布到集群上, 并行化执行 RDD. 首先把数据上传到 HDFS, Spark 计算引擎通过 textfile 把数据从 HDFS 上面加载到集群内存, 程序通过输入数据创建一系列的 RDD, 之后通过调用 Spark 的算子如 map、flatmap, filter 等对 RDD 进行操作, 这个过程也称为 RDD 的转化(transformation). 由于 RDD 具有只读性, 每次计算都产生新的 RDD, 一次操作 RDD 便转换为另一个 RDD, 执行完程序把结果存储到 HDFS.

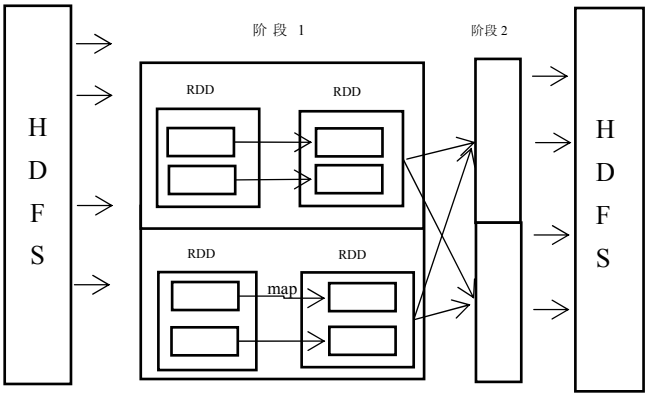

图 1　Spark 数据分析流程

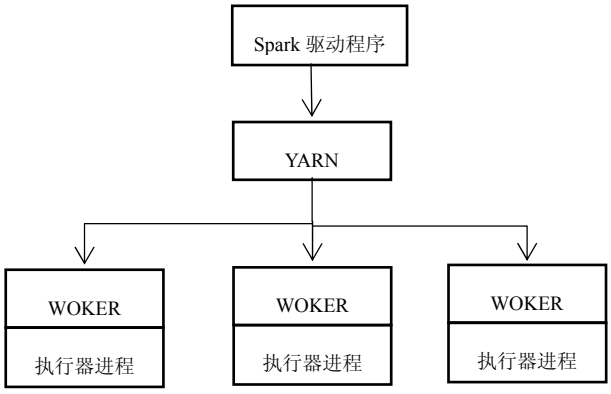

图 2　分布式 Spark 应用组件

图 2 是 Spark 集群运行程序时的流程, 从图中可以看出它是由一个 Spark 驱动器节点、YARN 集群管理器和一系列的集群工作节点(简称 worker)组成的. 实际上每个 worker 节点就相当于一台计算机, 而 Spark 驱动器是执行程序中 main 方法的进程. 在分布式集群环境下, Spark 集群采用的是主/从结构, YARN 集群管理器负责中央协调, 负责各个分布式工作节点的调度. 当用户提交程序到集群上, Spark 驱动器把用户程序转

化成一个一个的任务，接着 Spark 驱动器节点依赖 YARN 集群管理器启动每个 worker 节点的执行器进程，Spark 驱动器节点可以同时和大量的执行器节点进行通信，它可以根据当前的执行器节点集合把任务分配给合适的执行器进程，每个执行器节点代表一个能够处理任务和存储 RDD 数据的进程，驱动器节点和所有的执行器节点一起被称为一个 Spark 应用.

Spark 支持 Java、Scala、Python，本次研究所有代码都是基于 Scala 语言中实现的，Spark 将 Scala 用作其应用程序语言，特别适合生成速度快，数据量特别大的应用场景.

3 数据预处理与分析

3.1 数据预处理

由于使用的出租车 GPS 数据是车载 GPS 设备采集的，会由于设备或者其他的异常原因出现一些错误，如果对这些异常和错误不进行分析、过滤而直接加以应用，会直接或者间接影响结果的准确性，所以要对出现的一些不正常的 GPS 数据加以处理，以保证结果的准确性.

采集城市出租车 GPS 数据，每条记录包含序号、出租车 ID、时间、经度、纬度、瞬时速度、方向、状态.

表 1 GPS 数据示例

标识号	车牌号	GPS 时间	经度	纬度	速度	方向	车辆状态
42	陕 AU****	2011-06-17 00:00:17	108.91010	34.327631	0	0	5

车辆状态不同数字代表不同的含义，其中 1 为防劫; 2 为签到; 3 为签退; 4 为空车; 5 为重车; 6 为点火; 7 为熄火. 数据示例如表 1.

3.2 出租车 GPS 数据误差分析

本文对异常和误差产生的原因进行简单的分析和总结，总共有以下 3 种.

(1) GPS 设备故障

出租车车载 GPS 设备出现故障，例如: 返回重复数据; 数据方向错误; 数据接收延时而导致时间错误等等，对出现的这些错误都予以剔除.

(2) 建筑物或者其他信号的干扰

出租车车载 GPS 接收机的信号经常会由于其他信号的干扰或者高层建筑的遮挡，而导致信号接收不到，信号接收延迟，此类数据一般要进行接收识别和处理.

(3) 出租车司机错误或者不正确的操作

出租车司机人为的造成 GPS 数据错误，司机由于设备不熟悉或者其他的一些原因导致操作失误，例如经常会出现重复打表的情况，此类数据一般要予以剔除.

3.3 出租车 GPS 数据误差处理

综合考虑前面 3 种原因所造成的异常和错误，主要考虑对以下 4 种出租车异常 GPS 数据进行相应的处理.

(1) 经纬度越界

由于研究受 GPS 数据量的限制，本文主要研究范围为西安市. 通过百度软件查询得知，西安市坐标范围为 107.40 度～109.49 度和北纬 33.42 度～34.45 度之间，此范围以外的坐标位置数据要予以剔除.

(2) 重复数据去重

如果出租车司机重复打表、车辆堵车或者车辆在某段时间内静止，对此类数据进行判断并进行处理.

(3) 采集的数据时间间隔默认为 30s，剔除掉对时间间隔异常的数据

如果车载 GPS 设备故障、建筑物遮挡等原因导致时间接收延迟，对这些数据予以剔除.

(4) 数据格式异常

由于处理的数据量相对较大，会出现的各种各样的格式错误和异常，例如: 数据时间缺失，数据 GPS 缺失等等，对出现的这些异常予以直接剔除.

3.4 算法设计

Spark 对数据的操作主要分 3 个步骤，创建 RDD、转化已有的 RDD 以及调用 RDD 操作进行求值. 首先提交 Spark 程序，Spark 运行程序的 main()方法并构建一个 Spark context，然后创建 RDD，运用 Spark 提供的算子(如: map、reduce、distinct)对数据进行处理和转化，本质上就是 RDD 转化过程，对常用算子介绍如表 2 所示.

表 2 Spark 算子介绍

Spark 常用算子	功能
map	对每条数据简单的一对一映射处理，结果返回新的 RDD
filter	对 RDD 进行过滤操作
sortByKey	对(key, value)形式按 key 进行排序
distinct	对数据进行去重
reduceByKey	对(key, value)形式把 key 相同的数据的 value 进行处理

flatmap	作用与 map 类似, 区别是 flatmap 将结果整合到一起, 构成一个新的 RDD

(1) 算法 1 是计算每小时运行车辆数, 车辆每个小时都有多条运行数据, 先统计每辆车出现的时间段, 对这些时间段运行的多条车辆数据去重, 使用 reduceByKey 把相同小时时间内运行车辆数相加.

算法 1. 计算每小时运行车辆数算法-Spark

1. 输入清洗后的出租车 GPS 轨迹数据
2. 使用 Filter 过滤出状态为非 5 的数据
3. 使用 map 设置 key=车牌号: 时间, value=GPS 数据
4. 使用 sortByKey()按车牌号和时间排序
5. 使用 distinct 去除车辆重复出现的数据
6. 把输出每辆车每小时行驶数据使用 Map, map 设置 key=时间, value=1//value 设 1 用来计数
7. 使用 reduceByKey 把相同小时时间内运行的车辆数相加
8. 输出一天每小时车辆运行数

算法 2. 提取出租车行驶距离算法-Spark

输入: 清洗后的出租车 GPS 轨迹数据，相邻出租车数据时间 T

输出: 出租车出行次数，出租车行驶距离，出租车平均出行量

1. 输入清洗后的出租车 GPS 轨迹数据
2. 使用 Filter 过滤出状态为非运行状态的数据
3. 使用 map 设置 key=车牌号, value=GPS 数据
4. 使用 sortByKey 按车牌号排序, 得到每辆车的轨迹, 再用 flatmap 处理所有数据
5. While 数据不为 null
6. If (两条数据车牌号相等 && 两条数据小时时间是否相等)
7. If (两条数据的时间差==T)
8. 求出这 T 时间内距离 diatance, 出行次数加 1
9. 累加 sum1+=distance //sum1 为出租车一次出行的距离
10. End if
11. sum+=sum1 //sum 为出租车当前小时总出行量
12. sum1=0
13. End if
14. 每小时平均出行量=sum/运行车辆数
15. 每次平均出行量=sum/该时间所对应的出行次数
16. sum=0
17. End while

(2) 算法 2 是基于 Spark 提取出租车行驶距离算法. 该算法中, 采集相邻的数据默认时间为 30s, 所以设置时间 T 为 30s. 通过遍历所有数据, 先判断是同一辆车的数据, 再判断相邻两条数据时间间隔符合 30s 的情况下, 计算小时时间内行驶的距离, 进而可以得到一次行驶的出行量和该辆车小时时间内的总出行量. 本文把平均出行量分为每小时平均出行量(km/h)和每次平均出行量(km/次数)两个方面, 根据上面结论, 计算所有车一天每小时对应总出行量, 除以对应时间, 可得每小时平均出行量(km/h). 每次平均出行量可由所有车一天每小时对应总出行量除以该时间所对应的次数求出.

(3) 算法 3 是计算每小时实载率. 该算法中, 先计算每辆车在一天内每个小时运行的时间, 再计算所有车辆每个小时运行总时间, 再除以当前小时运行的车辆数可得到每个小时实载率.

算法 3. 计算每小时实载率-Spark

输入: 清洗后的出租车 GPS 轨迹数据，相邻出租车数据时间 T

输出: 每小时出租车对应的实载率

1. 输入清洗后的出租车 GPS 轨迹数据
2. 使用 Filter 过滤出状态为非 5 的数据
3. 使用 map 设置 key=时间, value=T
4. 使用 reduceByKey 求出每小时所有车辆运行时间
5. 平均实载率=每小时运行时间/小时内运行车辆数

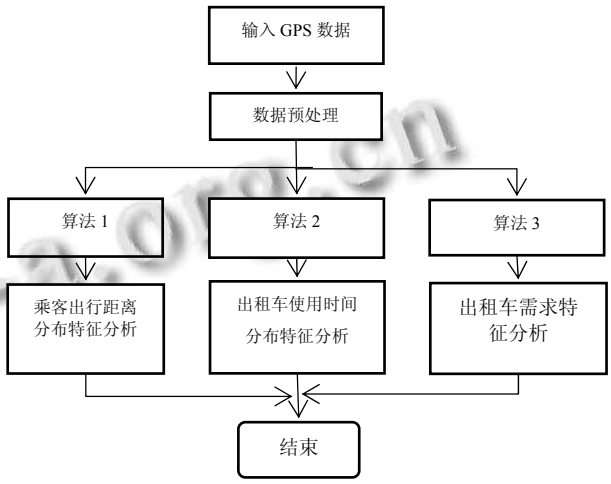

图 3　实验过程流程图

4　实验结果和分析

4.1 实验过程流程图

实验总体分为 3 个部分: 首先输入出租车的 GPS 数据, 进行预处理, 通过上面给出的算法 1 的结果分析出租车乘客的出行距离分布特征分析; 通过算法 2 的结果对出租车使用时间分布特征分析; 通过算法 3

的结果对出租车需求特征进行分析. 总体流程图如图 1 所示.

4.2 出租车乘客出行距离分布特征分析

实验是基于一周的出租车 GPS 数据进行统计, 数据记录的时间为一天 0 点到 24 点, 共 24 小时, 数据量为每天大约 3000 万条. 图 4 是根据西安市 2011 年 6 月 2 号全天的出租车 GPS 记录统计一天内每小时车辆运行数量, 从图中可以看出西安市出租车运营有两个高峰, 分别是白天高峰上午 7 点到下午 3 点, 晚上高峰 18 点到 22 点, 并进一步统计一天内总运行车辆数为 9243.

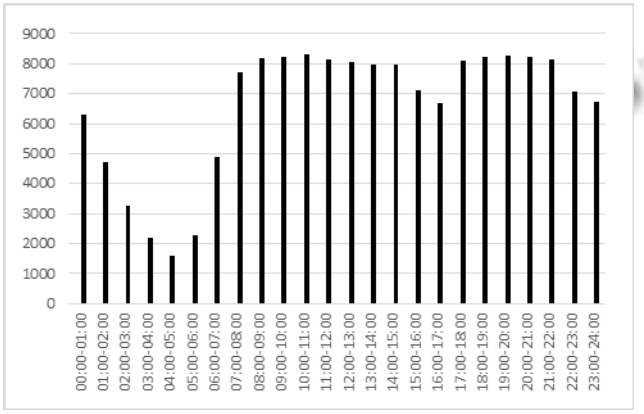

图 4 一天每小时车辆运行数

图 5 是统计一天出行距离分布情况, 是出租车每隔 2km 的出行次数. 图 6 是在图 5 的基础上对一周的出租车数据进行统计, 按 4km 的范围做统计, 总共统计所有出租车出行次数 2703793, 其中有 1370003 次出行是在 4km 之内的, 占据了总出行次数的 50%的比重. 表 3 在图 5、6 的基础上进一步求出不同距离的频率与累计频率, 得到不同距离的频率与累计频率表, 从图中可以看出出租车的出行频率最大值在 1~2km 和 2~3km, 经调查得出其原因在于西安市的起步价在 3km 之内, 该因素导致乘客出行大多集中在这一范围内. 从表中可以看出, 出租车出行频率随着距离范围的增加而减少, 这符合居民出行的规律的, 90%的出行在 12km 之内, 而 3km 内出行所占比例最大, 占到了 42.35%, 但是在 42km~44km 处, 出行频率出现了反差, 出现了异常的升高, 为此对这一范围进行了进一步的分析, 统计结果表明, 该区间出行中与西安市咸阳机场相关的出行占到了极大比重, 说明长距离出行中机场相关的出行比例最大, 而且需求比较稳定.

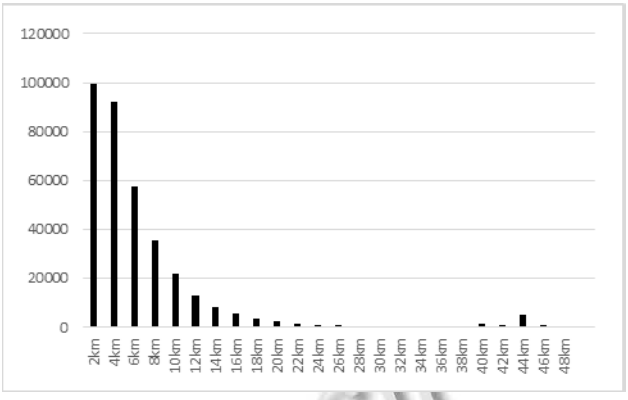

图 5 出租车出行距离分布图

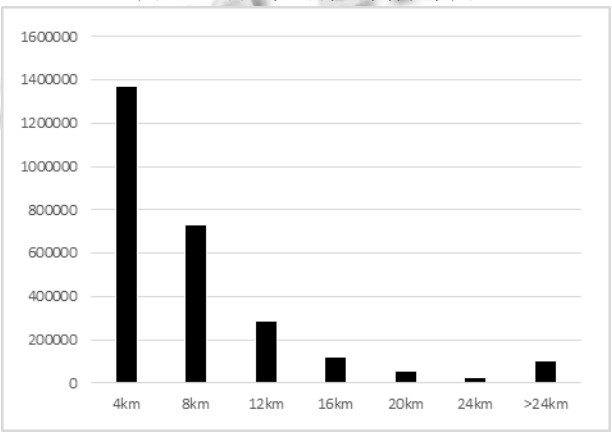

图 6 一周距离分布图

表 3 不同出行距离的频率和累计频率

距离/km	频率	累计频率	距离/km	频率	累计频率
0-1	0.119627751	0.119627751	24-26	0.002271318	0.966815082
1-2	0.160335452	0.279963203	26-28	0.001654173	0.968469256
2-3	0.143560053	0.423523256	28-30	0.001217382	0.969686637
3-4	0.115098421	0.538621677	30-32	0.00093558	0.970622218
4-5	0.089804507	0.628426185	32-34	0.000650961	0.971273179
5-6	0.071957261	0.700383446	34-36	0.000414248	0.971687426
6-7	0.056169313	0.756552758	36-38	0.001380826	0.973068253
7-8	0.044311472	0.80086423	38-40	0.003832497	0.97690075
8-9	0.034625458	0.835489688	40-42	0.003564786	0.980465536
9-10	0.02663302	0.862122708	42-44	0.014526856	0.994992392
10-11	0.020640802	0.88276351	44-46	0.002978639	0.997971031
11-12	0.015509437	0.898272947	46-48	0.002028969	1

4.3 出租车使用时间分布特征分析

图 7 根据一天的完整数据的统计, 本文得到了一天内不同时间段的平均出行量, 从图中可以得出出租车出行的平均出行量有三个比较平缓的高峰阶段, 上午的高峰阶段为: 8:00-11:00,下午的高峰阶段为: 14:00-17:00, 晚上的高峰阶段为: 19:00-22:00, 在图 7 的基础上对一周

每天内不同时间段的平均出行量进行统计和分析, 得到了一个相似的结果, 如图 8 所示, 只不过周末的上午高峰、下午高峰、晚上高峰的出行距离相对比较大. 反应了周末人们逐渐的有工作转向休闲的节奏.

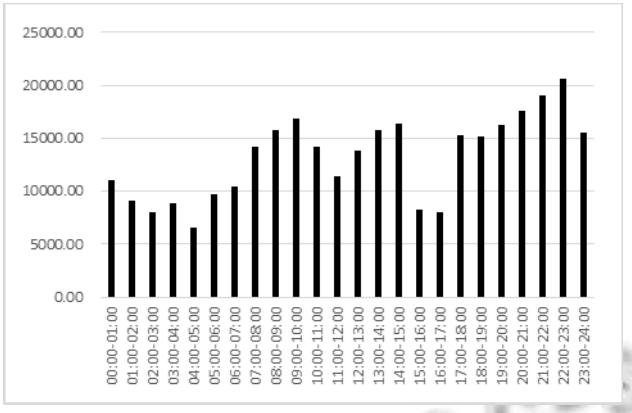

图 7　一天每小时车辆平均出行量

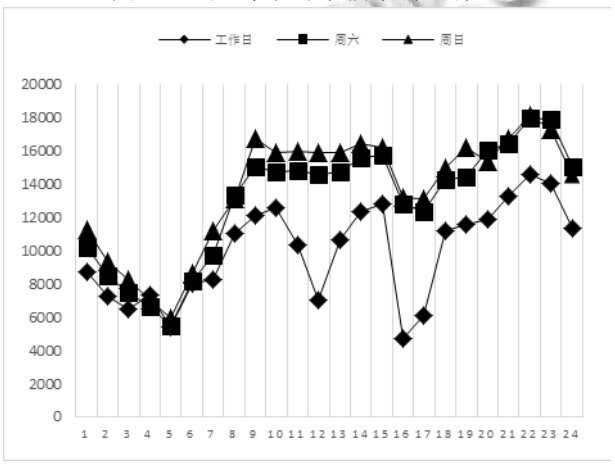

图 8　一周出行量统计

4.4 出租车需求特征分析

首先统计所有出租车每小时各自运行的时间, 进而得出所有出租车在每小时的平均实载率如图 9 所示.

从图中可以看出出租车平均实载比率出现 3 个低峰, 凌晨低峰从在 0 时到 7 时, 是一天内平均实载率最低的; 中午低峰出现从在 12 时到 13 时; 晚上低峰从在 15 时到 17 时, 经过分析凌晨低峰主要出现原因是居民多处于休息的状态, 中午低峰和晚上低峰多是处于用餐的. 本文用出租车的数量与实载比率的乘积来反应出租车的需求状况. 从图 4 与图 9 的结果可以得出结果如图 10 所示, 可以看出出租车需求状况主要与平均实载率相符合, 图 10 显示的主要特征是: 凌晨 0 时到 7 时是一天内需求的最低时段, 有明显的早晚高峰; 我们进一步对一周的数据求出相应的出租车需求状况如图 11 所示, 得出节假日的需求变化特征明显与工作日不同, 除了同比工作日早高峰时间略高外, 其他时间段需求明显均比较高.

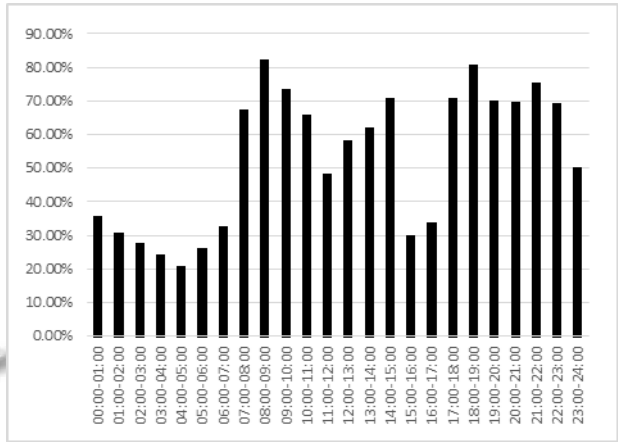

图 9　平均实载率

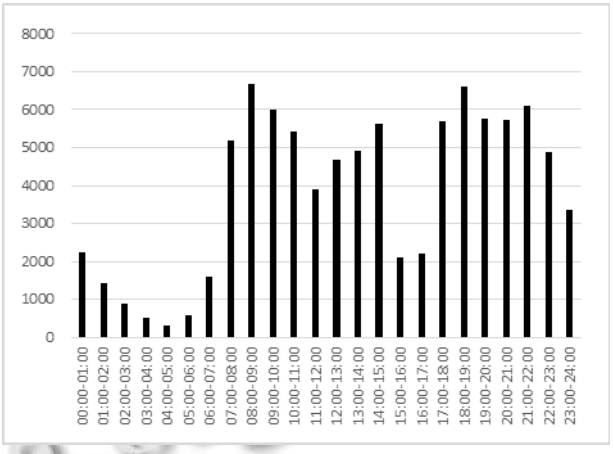

图 10　出租车需求

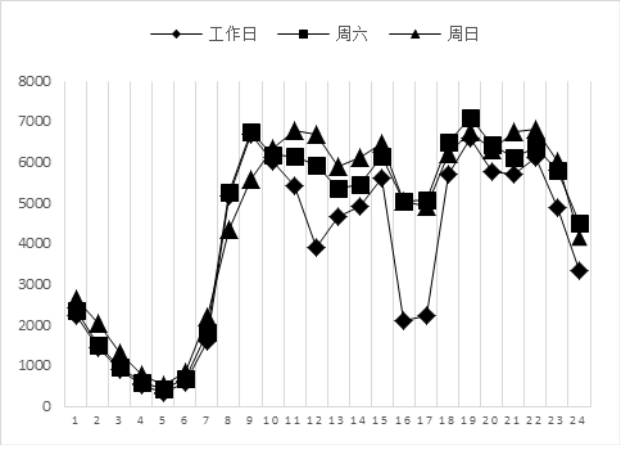

图 11　一周出租车需求

5 总结

本文提出基于 Spark 平台对海量出租车数据进行分析. 在该平台中, 能并行对海量出租车数据进行处理和分析, 并能迅速统计出结果. 文章首先简单介绍了 Spark 这个大数据平台, 之后介绍如何对 GPS 数据做预处理, 包括分析数据产生异常和错误的原因和其相对应的异常数据的处理方法, 最后给出 3 种算法用来提取城市出租车数据中的有效信息, 基于这些信息, 主要分析了出租车乘客出行距离分布特征、使用时间分布特征及出租车出行需求特征, 分析结果符合城市出租车运行特点. 本文未来工作是进一步分析出租车使用空间特征.

参考文献

1 刘丽娜,陈艳艳,张文阁.北京市出租车乘客需求预测模型研究.交通标准化,2010,(13):89–92.

2 丁圣勇,闵世武,樊勇兵.基于 Spark 平台的 NetFlow 流量分析系统.电信科学,2014,30(10):48–51.

3 李艳红,袁振洲,谢海红,等.基于出租车 OD 数据的出租车出行特征分析.交通运输系统工程与信息,2007,7(5):85–89.

4 童晓君.基于出租车 GPS 数据的居民出行[硕士学位论文].长沙:中南大学,2012.

5 魏祥.基于大数据挖掘的出租车行为检测方法的研究[硕士学位论文].南充:西南石油大学,2015.

6 翁剑成,刘文韬,陈智宏,等.基于浮动车数据的出租车运营管理研究.北京工业大学报,2010,(6):779–784.

7 李明珠.基于浮动车数据的出租汽车 OD 分布及运营特点研究[硕士学位论文].北京:北京交通大学,2009.

8 李宇光,熊普选,乐阳.基于大样本浮动车数据的武汉市车辆行驶速度获取与分析.交通信息与安全,2009,27(4):26–29.

9 秦玲,张剑飞,郭鹏.浮动车交通信息采集与处理关键技术及其应用研究.交通运输系统工程与信息,2007,7(1):39–42.

10 Tang J, Liu F, Wang Y, et al. Uncovering urban human mobility from large scale taxi GPS data. Physica A Statistical Mechanics & its Applications, 2015, 438: 140–153.