

详细方案

团队编号:1905514 小鸡炖蘑菇

目录

1. 项目背景.....	2
2. 项目目标.....	2
3. 问题分析.....	2
4. 解决方案.....	2
4.1 数据处理解决方案.....	2
4.1.1 数据说明	2
4.2 数据分析	4
4.2.1 出行分析.....	4
4.2.2 驻留分析.....	5
4.3 web 前端展示	6
5. 系统架构和设计	7
6. 项目管理.....	7
6.1 项目进度安排	7
6.2 项目管理过程.....	9

1. 项目背景

智慧交通是城市化进程不断向前推进的产物，是城市交通问题日益严峻和土地资源短缺状况共同作用下的必然结果。随着城市规模的不断发展，相应增加了城市人口数量和汽车保有量，这种形式下，逐渐加剧了不断增长的交通流和有效交通用地之间的矛盾。受多方面因素的影响，城市交通基础设施建设远跟不上交通流增长的步伐，更无法满足人们顺畅出行的需要，而智能交通系统建立的主要目的在于通过增加交通安全、提高交通效益、提高公交效率，减少交通负荷，提高交通决策的合理性来提高城市交通管理水平，充分利用已有的交通资源，满足人们便捷顺畅出行的需求。

2. 项目目标

- (1) 找出信令数据中的潜在价值和规律；
- (2) 利用算法模型，合理实现人口的出行分析和驻留分析；
- (3) 利用分析模型，实现人群出行方式的标签化；
- (4) 了解区域的拥挤程度、高峰时段、是否有异常聚集的现象。
- (5) 将数据分析结果转换为直观的图形、文字、数字可视化信息。

3. 问题分析

数据来源巨大，高吞吐，为避免数据丢失，可以采用 kafka。出行方式分时，应该离线分析，驻留的分析应该实时在线。正好 kafka 支持实时和离线两种解决方案，一部分数据通过 Flink 做实时计算处理，一部分到 hadoop 做离线分析。

人群密度分析，可以根据基站坐标在 web 前端展示，实时在线更新。添加报警系统，人群密度过高，系统报警。

前端展示可以用 spring 框架，echarts 做数据展示。政务数据，前端展示应该考虑大屏展示。

4. 解决方案

4.1 数据处理解决方案

4.1.1 数据说明

手机信令通过全球移动运营商(GSM)在移动基站获取的数据。当移动电话开启时，它会自动和移动基站连接。这时，基站的位置信息也会被 GSM 记录下来。此外，当以下三种活动发生时，GSM 会更新手机所在基站的位置：

- (1) 该手机用户发生通话

- (2) 该手机用户接收或发送短信
- (3) 该手机用户改变了移动基站的位置

此外, GSM 还会在一个相对固定的时间间隔内, 更新手机用户所在的基站位置。(一般为 30min)

手机信令数据包含 imsi、timestamp、lac_id、cell_id 等信息其中, IMSI 是手机用户的唯一标识。。TimeStamp 则代表了 GSM 更新基站所在位置的时间。lac_id 代表地区区域码, cell_id 代表一个移动基站。通过组合 lac_id 和 cell_id 就可以获得该手机所在的基站经纬度信息。

timestamp	imsi	lac_id	cell_id
1538498482636	460000095007329000	16789	67567924
1538502547368	460000095007329000	16789	67567924
1538481509242	460000095007329000	16789	67567924
1538526105074	460000095007329000	16789	67200820
1538526153986	460000095007329000	16789	67200820
1538527924038	460000095007329000	16789	67200820
1538530014006	460000095007329000	16789	67200820
1538544930607	460000095007329000	16789	67475508
1538526826182	460000095007329000	16789	67200820

longitude	latitude	laci
123. 4159698	41. 80778122	16789-67567924
123. 4396362	41. 80488968	16789-67200820
123. 4312515	41. 80846024	16789-67475508
123. 4372406	41. 7996788	16640-67195659
123. 4344177	41. 81195831	16789-67314199
123. 4344177	41. 81195831	16789-67313941
123. 4121704	41. 8152504	16789-67681055

4.1.2 基本数据清洗

将 20181003 当天的数据整理为时间序列流。为了达到模拟的效果, Proucer 模块将会每 5s 向 kafka 中发送一次数据, 也就相当于 kafka 每 5s 收到一次日志收集数据. 日志收集数据存放于 kafka 的 “timeStream” topic 中。

对于题目要求和信令数据的特点设计了基本的数据清洗算法。

首先将数据以流的形式读入, 然后按以下几个步骤对数据进行清洗:

- 1) 使用 flink 的 filter 算子过滤掉 imsi 包含 ‘#’, ‘*’, ‘^’ 特殊字符的条目
- 2) 使用 flink 的 filter 算子过滤掉 imsi、lac_id、cell_id 为空的条目
- 3) 使用 flink 的 map 算子将时间戳进行转换
- 4) 使用 flink 的 filter 算子过滤掉不是 20181003 的数据
- 5) 使用 flink 的 filter 算子结合本地的经纬度表去除 lac_id 和 cell_id 关联为空的数据
- 6) 最后将数据以特定格式输出

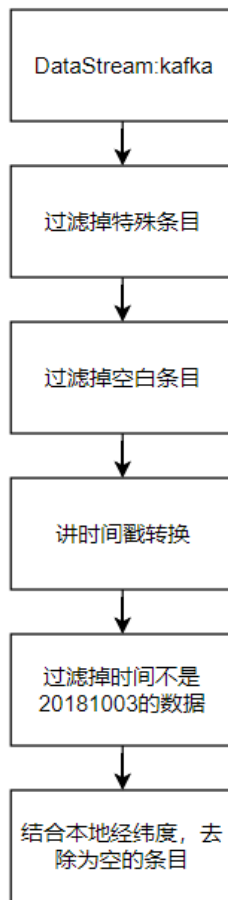


图 1 数据清理图

4.2 数据分析

4.2.1 出行分析

在进行数据清洗后,我们得到了一系列以人为单位时间升序的信令数据。但是,我们还不能直接对这些数据进行分析。对于一个人的数据来说,其中可能存在着驻留数据,乒乓数据,漂移数据。我们需要对这些数据进行过滤。为了进行出行链的提取,我们必须对信令数据的特点和出行的定义进行介绍。然后,我们再进行介绍如何实现过滤驻留数据,乒乓数据,漂移数据,最终提取出出行链。出行是指从起点到终点移动的全过程。

一次出行包含以下几个常见属性:

- 1) 包含出行起点和终点
- 2) 一次出行使用一种或几种交通方式
- 3) 一次出行要达到一定的时间和距离

上海市第 4 次居民出行调查中是这样定义出行的:单程 400m 以上或步行时间超过 5 分钟的过程. 广州 2005 年居民出行调查定义出行为:使用某种交通工具出行距离超过 500m 或步行时间超过 5 分钟。

根据以上资料,本文中定义出行为:包含起点和终点(且起点和终点不相同)的一次距离大于 500 或时间超过 5min 的过程。

常规计算出行距离的两种方式:

- 1) 精确计算 将识别出来的出行点和路网匹配计算实际的距离
- 2) 近似计算 直接计算出行点两点之间的直线距离

由于数据量巨大,如果对大量数据点进行精确计算匹配路网,计算性能上会有很大的损耗。所以,本文采取近似计算方式。但是,对于近似计算,我们可以使用道路的非直线系数来模拟真实的道路距离情况。道路的非直线系数是指道路起点和终点之间实际的距离与起点和终点空间直线距离的比值。参考<城市道路交通规划设计规范 GB50220-95>。城市路网的平均非直线系数一般为 1.15~1.2。项目使用的数据来源是沈阳市的手机用户。沈阳市是一个新一线城市,所以本项目对于沈阳市的道路非直线系数设定为 1.2。

对数据进行聚类分析,把数据分成多份,并且打上标签。

对于聚类分析后得出的标签化数据,为了进一步的提高标签的准确性.我们将公交车,汽车,地铁三种方式进一步结合站点的经纬度信息进行判断。

首先考察到,沈阳市的地铁站与地铁站之间的平均距离不超过 1.5km,公交车站与公交车站的距离不超过 1km。

我们将之前聚类后的平均出行速度前 3 的出行方式依次进行如下判断:

首先进行地铁的检验。

对于一条出行链,如果出行链上的每一个点都在地铁站点的一定范围内,那么我们判断,这次出行的出行方式是地铁。

然后进行公交车的检验。

对于一条出行链,如果出行链上的每一个点都在公交车站站点的一定范围内,那么我们判断,这次出行的出行方式是公交车。

最后,在经过地铁检验和公交车校验后的数据,我们可以比较有把握的判断,这些数据的出行方式属于汽车。

4.2.2 驻留分析

我们设置距离阈值为 500,时间阈值为 30min. 如果当前出行点和上一个出行点的距离小于距离阈值且出行时间小于时间阈值. 则判定该用户是一动不动的。

此外,对于当前出行点的经纬度等于下一个出行点的经纬度的情况,我们也

判定为驻留,保留最后一个点。

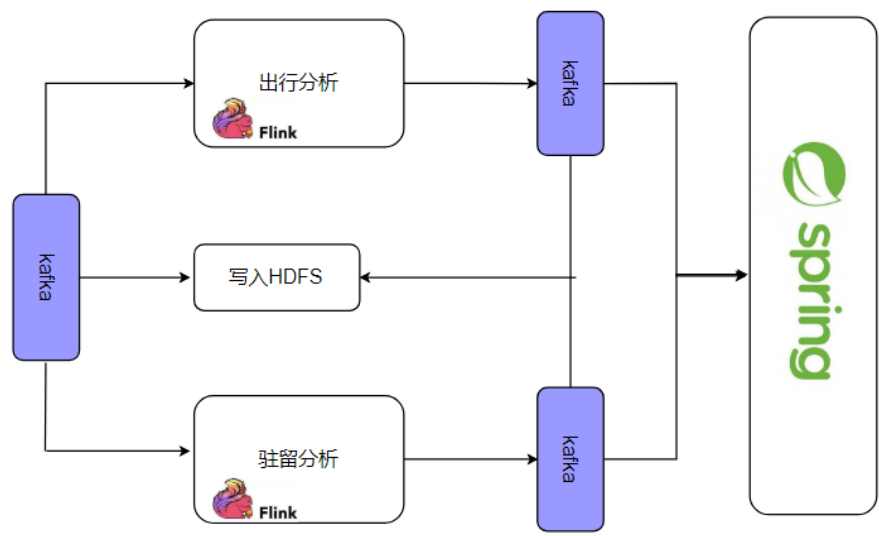
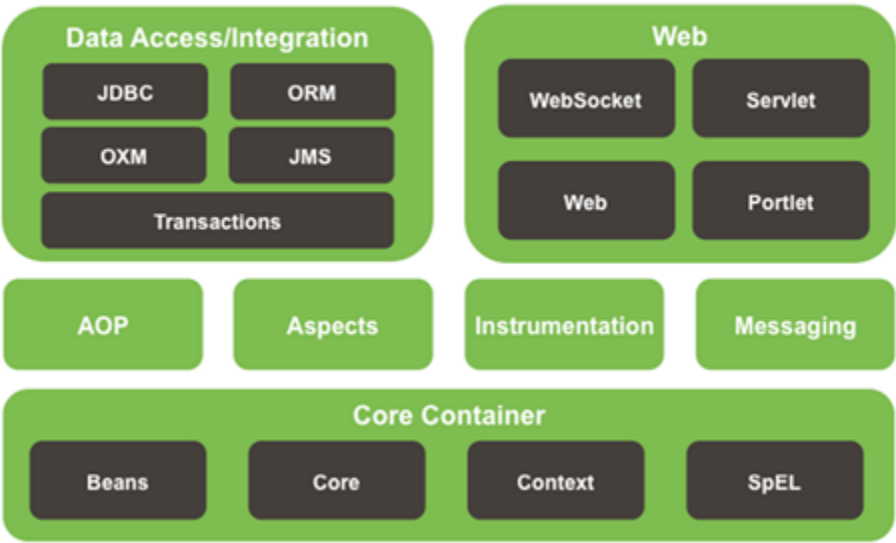


图 2 系统流程图

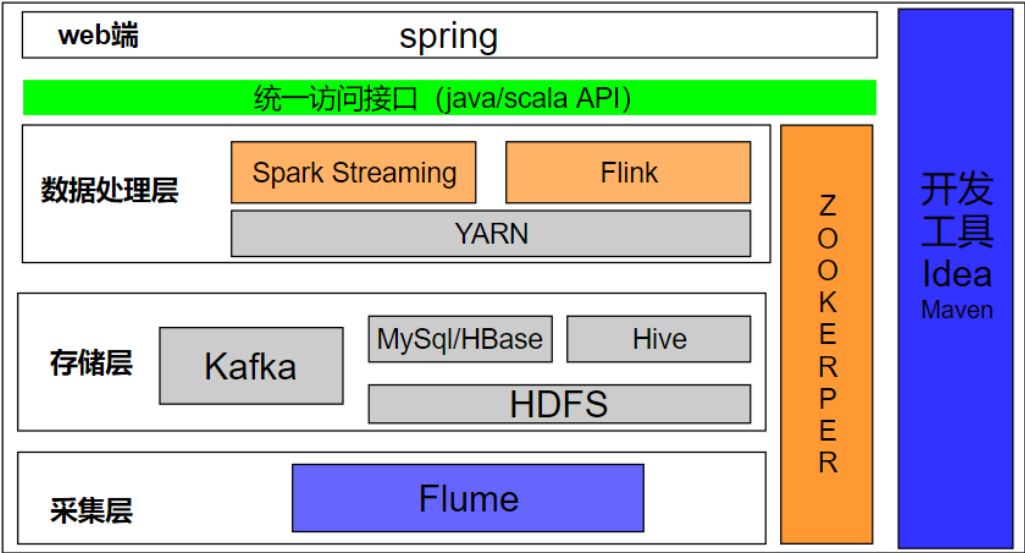
4.3 web 前端展示

SSM 框架整合,即整合 SpringMVC、Spring 和 Mybatis 框架。其中 SpringMVC 属于 SpringFarmeWork 的后续产品,它提供了构建 Web 应用程序的全功能 MVC 模块,分离了控制器、模型对象、过滤器以及处理程序对象的角色,这种分离让它们更容易进行定制。

Spring 是一个轻量级开源框架,它的主要特点是方便解耦、简化开发、面向切面(AOP)的编程支持和声明式事务支持,其主要优点有低侵入式设计、独立于应用服务器、允许将一些通用任务如日志等进行集中处理。Mybatis 是轻量级 ORM 框架,它消除了几乎所有的 JDBC 代码和参数的手工设置以及结果集的检索,使用简单的 XML 或注解用于配置和原始映射,将接口和 Java 的 POJOs 映射成数据库中的记录。其 spring 框架架构图如下:



5. 系统架构和设计



6. 项目管理

6.1 项目进度安排

表 6-1 项目进度安排表

序号	任务名称	开始时间	结束时间
1	可行性分析	2020-01-01	2020-01-07
2	需求分析	2020-01-08	2020-01-13

3	技术准备	2020-01-14	2020-01-19
4	概要设计	2020-01-14	2020-01-19
5	详细设计	2020-02-10	2020-02-18
6	系统设计	2020-02-10	2020-02-20
7	系统编码	2020-02-21	2020-03-21
8	系统功能集成测试	2020-03-22	2020-03-23
9	系统测试	2020-03-23	2020-03-26
10	系统修正	2020-03-27	2020-04-18

表 6-2 项目任务分解与进度安排

任务名称	开始时间	结束时间	参与人员	工作量	工作成果
可行性分析	2020-01-01	2020-01-07			可行性分析报告、团队分工
需求分析	2020-01-08	2020-01-13			项目管理计划、需求规格说明书
技术准备	2020-01-14	2020-01-19			项目开发计划、概要说明书
概要设计	2020-01-14	2020-01-19			项目开发计划、概要说明书
详细设计	2020-02-10	2020-02-18			数据库设计、测试设计、详细设计
系统设计	2020-02-10	2020-02-20			概要设计、数据库设计、测试设计
系统编码	2020-02-21	2020-03-21			源代码
系统功能集成测试	2020-03-22	2020-03-23			源代码、功能集成
系统测试	2020-03-23	2020-03-26			功能测试、系统测试、测试文档
系统修正	2020-03-27	2020-04-18			源代码定稿

6.2 项目管理过程

- 1) 项目组制定项目开发计划，建立人员组织，并进行人员分配。
- 2) 根据项目开发生命周期启动项目。
- 3) 召开项目会议，一周一次小会，一月一次大会，并建立会议文档，保证项目进度和保证项目过程出现问题的解决。
- 4) 小组长扮演项目监督角色，对项目生命周期中的正常运行情况进行监督并对出现的问题进行处理。