**数据融合算法设计**

## 算法目的

对OIDD、移动DPI、无线WCDR和移动DDR数据数据中的重复数据进行清洗并对位置准确性产生干扰的信息通过相应的算法进行过滤。

对OIDD、移动DPI、无线WCDR和移动DDR数据数据中的重复数据进行清洗并对位置准确性产生干扰的信息通过相应的算法进行过滤。

## 算法思想

程序架构使用多个mapReduce相互组合，针对mapReduce的特性对用户分组，在每个组内进行时间重复的清洗、分钟粒度的提取、超速数据清洗，乒乓切换数据清洗和bsid转换经纬度。

### 2.1：时间重复，分钟粒度提取算法设计

该算法的目的：时间重复清洗主要完成单一数据文件采集数据的要求不同，相邻记录的时间间隔小，相邻位置距离短，无效的数据造成数据库文件大，对于数据应用的处理过程中造成效率低。

目前数据概况：DPI数据每分钟产生多条记录，WCDR数据中约30秒产生一条消息记录，移动DDR和OIDD信令数据消息记录间隔在分钟级别内重复点很少。

第一次清洗：

读取的各类数据格式参见数据融合需求文档中【2.1.1，2.1.2，2.1.3，2.1.4】各章节。

对各类型的数据以【手机号+时间（201610150101分钟级别）+城市代码】作为分组标记，在该分组内每组可能有1 到n条数据，然后取该组中秒数最大的这条数据输出，其他小于该秒的数据忽略不输出，此时得到该类型数据的分钟粒度的数据。针对各种类型数据格式和字段的不同第一次清洗后的数据格式统一后如下：

mdn\_time\_cityid\_sec\_lon,lat\_bsid\_grid\_servicetype\_eventType datasource

第二次清洗：

同时读上述四类数据的处理结果，以同样的方法分组取出该组中秒数最大的这条数据输出，此时得到多个类型文件库融合后的分钟粒度的数据，作为后续其他清洗工作的基础数据。

通过第二次数据清洗得到的数据格式如下：

mdn\_time\_cityid sec\_lon,lat\_bsid\_grid\_servicetype\_eventType\_datasource

### 2.2：超速数据清洗算法设计

该算法的目的：由于无线信号受到反射、折射、绕射和多径等因素的影响，根据无线电波计算经纬度的偏差比较大，在合并的数据中存在明显不合理的经纬度位置，使应用中在定位时出现明显偏差。因此，针对此类记录点通过超速数据进行剔除。

超速数据清洗以四类数据融合后的分钟粒度数据为基础数据进行清洗。读取的数据来源为算法1中第二次清洗后的数据，格式如下：

mdn\_time\_cityid sec\_lon,lat\_bsid\_grid\_servicetype\_eventType\_datasource

1）首先以用户手机号分组，对每组（每个手机号下的所有数据）按时间先后进行排序。

2）以组为基础，对一组数据开始处理：

顺序读取（因为是按照时间排序好的数据），读第一条数据的时候，将该条数据记为待计算数据，记录该条待计算的时间和待计算经纬度为：time1、lonlat1

3）从前三条数据中寻找起始正常点，如果第一条正常则以第一条为正常点，（速度计算参照第4步）否则计算第二条和第三条的之间的时间和距离差值计算速度判断，如果正常则以第二条为起始正常点，否则计算第一条和第三条的速度判断，如果正常则以第一条为起始正常点

4）从第3）步骤完成以后继续向后读取数并与上一条数据做计算，记该条为第n条数据，取第n条数据的时间和经纬度为time2、lonlat2

计算（time2,time1）得到两个时间点的秒差值secdiff

计算（lonlat2,lonlat1）得到两个经纬度之间的距离distance (米)

通过distance/secdiff得到第n-1条数据的穿越速度V (米/秒)

判断V 是否小于等于97.22 （通过350km/h转化为97.22米/秒）

如果V小于等于97.22则说明第n-1条数据没有超速，第n-1条数据为合格数据，输出第n-1条数据，否则第n-1条数据忽略

4）如果V小于等于97.22则把time2的值给time1，把lonlat2的值给lonlat1，否则不做交换

5）继续读取数据重复步骤3和4

6）当读到最后一条数据的时候，直接将最后一条数据输出，至此该组数据（一个手机号下的所有数据）超速数据清洗完毕，各条数据的输出格式如下（\t分割）：

mdn time sec cityid lon,lat bsid grid servicetype eventType datasource

### 2.3：乒乓切换数据清洗算法设计

乒乓数据是在两个或多个小区信号强度差不多的时候，手机在两者之间频繁的进行切换，造成乒乓效应。体现位置上，在短时间内手机在两个BSID间做频繁切换，经纬度相差比较大。该算法主要过滤清洗这些产生乒乓效应的数据。

乒乓清洗的数据来源为算法2.2:超速数据清洗之后的数据，格式如下：

mdn time sec cityid lon,lat bsid grid servicetype eventType datasource

1）首先以用户手机号分组，对每组（每个手机号下的所有数据）按时间先后进行排序。

2）以组为基础，对一组数据开始处理：

首先对已经按照时间排好序数据进行判断查找该组连续出现的乒乓数据数据组出现的次数K，如果K为偶数则执行下面偶数算法，如果K为奇数则执行下面奇数算法，偶数算法和奇数算法规则如下：

偶数算法：以下面一组按照时间排序的乒乓数据为例： CABABABABABD 为连续8个乒乓组数据（其中乒乓数据标黄标识），从乒乓数据开始到第8/2=4条数据，分界点即CABABA BABABD 黑色标识即为分界点，将分界点前面的所有bsid为B的每一条数据中的bsid和经纬度改为与之相邻的前一条数据的bsid和坐标，分界点后的所有bsid为A的每一条数据中的bsid和经纬度改为与之相邻的前一条数据的bsid和坐标，改后为CAAAAA BBBBBD最后将该组所有数据输出

奇数算法：以下面一组按照时间排序的乒乓数据为例 CABABABABABAD 为连续9个乒乓组数据，从乒乓数据开始到第9/2-1=4条数据，分界点即 CABABA BABABAD 黑色标识即为分界点，将分界点前面的所有bsid为B的每一条数据中的bsid和经纬度改为与之相邻的前一条数据的bsid和坐标，分界点后的所有bsid为A的每一条数据中的bsid和经纬度改为与之相邻的前一条数据的bsid和坐标，改后为CAAAAA BBBBBD最后将该组所有数据输出

乒乓清洗后的数据格式即为最终数据的存储格式，如下：

mdn time sec cityid lon,lat bsid grid servicetype eventType datasource

### 附：基站id转换经纬度设计

对于DDR和DPI数据中没有经纬度，读取扇区基础配置表，根据BSID匹配扇区中心点的坐标作为DDR和DPI数据中的记录点经纬度。

首先读取redis中存的扇区基础配置表并将所有信息放入缓存供程序快速通过bsid找到对应的经纬度，通过在gis-grid-1.1.jar包的com.navinfo.grid中，调用Grid.getGridID500M(lon, lat);方法得到网格id