软件学报 ISSN 1000-9825, CODEN RUXUEW
*Journal of Software*,2016,27(7):1700−1714 [doi: 10.13328/j.cnki.jos.005035]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

# 面向海量低质手机轨迹数据的重要位置发现*

章志刚, 金澈清, 王晓玲, 周傲英

(华东师范大学 计算机科学与软件工程学院 数据科学与工程研究院,上海 200062)

通信作者: 金澈清, E-mail: cqjin@sei.ecnu.edu.cn

**摘 要**: 重要位置是指人们在日常生活中的主要活动地点,比如居住地和工作地.智能手机的不断发展与普及为人们的日常生活带来了极大的便利.除了通话、上网等传统应用之外,手机连接基站自动生成的日志记录也是用于用户行为模式挖掘的重要数据来源,例如重要位置发现.然而,相关工作面临着诸多挑战,包括轨迹数据规模庞大、位置精度低以及手机用户的多样性.为此,提出了一个通用解决框架以提高轨迹数据可用性.该框架包含一个基于状态的过滤模块,提高了数据的可用性,以及一个重要位置挖掘模块.基于此框架设计了两种分布式挖掘算法:GPMA(grid-based parallel mining algorithm)和 SPMA(station-based parallel mining algorithm).进一步地,为提高挖掘结果的准确性和精确度,从 3 个方面进行优化:(1) 使用多元数据的融合技术,提高结果的准确性;(2) 提出了无工作地人群的发现算法;(3) 提出了夜间工作人群的发现算法.理论分析和实验结果表明,所提算法具有较高的执行效率和可扩展性,并具有更高的精度.

**关键词**: 低质;轨迹挖掘;重要位置;数据修正

**中图法分类号**: TP311

中文引用格式: 章志刚,金澈清,王晓玲,周傲英.面向海量低质手机轨迹数据的重要位置发现.软件学报,2016,27(7):1700−1714. http://www.jos.org.cn/1000-9825/5035.htm

英文引用格式: Zhang ZG, Jin CQ, Wang XL, Zhou AY. Discovering important locations from massive and low-quality cell phone trajectory data. Ruan Jian Xue Bao/Journal of Software, 2016,27(7):1700−1714 (in Chinese). http://www.jos.org.cn/1000-9825/5035.htm

## Discovering Important Locations From Massive and Low-Quality Cell Phone Trajectory Data

ZHANG Zhi-Gang, JIN Che-Qing, WANG Xiao-Ling, ZHOU Ao-Ying

(Institute for Data Science and Engineering, School of Computer Science and Software Engineering, East China Normal University, Shanghai 200062, China)

**Abstract**: Important locations mainly refer to the places where people spend much time in the daily life, including their home and working places. The development and popularization of smart cell phones bring great convenience to people's daily life. Besides making calls and surfing the Internet, the logs generated when visiting the base stations also contribute to users' pattern mining, such as important location discovery. However, it's challenging to deal with such kind of trajectory data, due to huge volume, data inaccuracy and diversity of cell phone users. In this research, a general framework is proposed to improve the usability of trajectory data. The framework includes a filter to improve data usability and a model to produce the mining results. Two concrete strategies, namely GPMA (grid-based parallel mining algorithm) and SPMA (station-based parallel mining algorithm), can be embedded into this framework separately. Moreover, three optimization techniques are developed for better performance: (1) a data fusion method, (2) an algorithm to find users who have no work

* 基金项目: 国家重点基础研究发展计划(973)(2012CB316203); 国家自然科学基金(61370101, U1501252, 61532021); 上海市教委科研创新重点项目(14ZZ045)

Foundation item: National Basic Research Program of China (973) (2012CB316203); National Natural Science Foundation of China (61370101, U1501252, 61532021); Innovation Program of Shanghai Municipal Education Commission (14ZZ045)

收稿时间: 2015-09-25; 修改时间: 2016-01-12; 采用时间: 2016-02-22; jos 在线出版时间: 2016-03-22

CNKI 网络优先出版: 2016-03-22 13:23:27, http://www.cnki.net/kcms/detail/11.2560.TP.20160322.1323.003.html

places, and (3) an algorithm to find people who work at night and fix their important locations. Theoretical analysis and extensive experimental results on real datasets show that the proposed algorithms are efficient, scalable, and effective.

**Key words**: low quality; trajectory mining; important location, data repairing

信息技术极大地推动了国民经济与社会的发展.随着数据采集技术的不断进步和新型应用的不断涌现,待管理数据的规模也飞速增长.能否高效地管理、分析数据业已成为衡量一个国家综合国力的重要指标.人类社会活动所产生数据是人类行为研究的主要依据,也是大数据研究的重要内容.相比过去,由于社会的进步和交通方式的进步,人类活动的形式和范围发生了巨大改变.但受限于地理和社交关系等因素,人类的行为活动也往往会呈现出规律性[1−3].其中一条规律就是,人们围绕其居住地、工作地做周期性的位置变迁[4].因此,发现以居住地和工作地为代表的重要位置具有较高的研究意义和应用价值[4−9].例如,在北京、上海等国际化大都市,堵车现象日趋严重.这里面隐藏的社会现象就是居住地和工作地距离过远,从而上班族需要花费大量的时间在上下班路上.如果能够通过技术手段分析出用户的居住地和工作地等信息,将有助于城市规划者优化城市各类资源配置,减缓环境污染和社会矛盾,为城市居民提供更优质的服务.

长期以来,由于缺乏相关数据的支持,发现用户重要位置仅能够采用问卷调查等人工手段完成,费时费力且效果不佳.随着智能手机的不断发展与普及,手机在人类生活中所扮演的角色越来越重要.除了电话、短信、上网等功能之外,手机与其相邻基站进行连接所产生的日志记录,即手机轨迹数据,在一定程度上记录了用户的日常行为.该数据具有数据量大、用户基数大及覆盖人群广等特点,并为发现用户重要位置提供了可能.以上海市为例,运营商在上海布置了上万个各种类型的基站,这些基站覆盖了上海全市范围.当用户需要使用运营商业务的时候,则由其所处位置邻近的运营商基站提供服务.现有资料表明,截止 2014 年 6 月 1 日,上海市有约 2 300 万常住人口,而移动电话持有数高达 3 234.6 万部,人均持有 1.4 部手机[10].

然而,“量大”并不意味着“质高”.对于重要位置分析来说,手机轨迹数据的低质主要体现在以下 4 个方面:

(1) 数据不准确性.轨迹数据本身是用户连接基站的记录集合,用户每连接一次基站,意味着用户在该基站周边的区域范围内,但不知道用户的具体位置.因此,数据本身存在着位置不确定性问题.同时,基站类型具有多样性,其包含微站、宏站、直放站和射频拉远站等类型,各类型的基站覆盖范围从几百米到几公里不等;

(2) 数据分布密度不均.基站的分布情况在城市的不同区域差异显著,例如市中心区域的基站密度远高于郊区的基站密度.因此用户可能在市中心区域出现的数据较多,而在郊区出现的数据较少;

(3) 数据中包含噪声.手机轨迹数据除了包含用户在重要位置的停留信息外,还混杂着大量在其他非重要位置上停留的记录(称为“噪声”数据),如用户上下班途中连接基站所产生的记录.对重要位置分析来说,必须考虑噪声数据对分析的影响;

(4) 数据中存在基站跳变现象.换言之,用户所处位置恰巧处于多个基站的服务范围之内,手机信号会在多个基站间切换.

图 1 介绍了某用户一天内连接基站的情况,矩形框中的时间表明,该用户曾经于该时间点与基站通信.例如,f 基站的附加信息是“11:52|12:20”,这表明该用户曾经于这两个时间点分别访问基站 f.其中,标记为 L 的蓝色标签指示了用户的居住地,标记为 W 的蓝色标签指示了用户的工作地.用户在 L 点停留期间,其手机会在 a、b 和 c 这 3 个基站间切换;同时在 W 点停留期间,其手机也会在 d 和 e 两个基站间进行切换.目前最好的处理方法[11]是针对精确轨迹数据进行分析,在海量低质情况下,不仅运行效率不高,而且结果准确率较低.

用户的多样性也为发现用户居住/工作地增加了难度.部分用户同时拥有一个以上工作地或居住地,还有些用户的居住/工作地会随用户搬家、换工作等原因而发生变化;部分离、退休人员无工作地;一些人员夜间工作而白天休息.现有工作未考虑用户的多样性问题.本文结合数据融合等分析技术以解决用户多样性问题,提升结果的准确度.

针对轨迹数据质量的问题,本文首先提出了一个通用解决框架,并提出了两种挖掘算法.进一步提出了 3 种处理方案,以提高结果的准确度和精度.实验结果表明,本文提出的算法不仅准确率和召回率高于已有方法,挖掘出的重要位置的精确度也有所提升.此外,所提出的算法具有很好的运行效率和可扩展性.


Fig.1 Example of one user's trajectory in one day

图 1 某用户一天的基站连接日志

本文的主要贡献如下:

- 针对海量、低质手机轨迹数据,提出了一个通用处理框架用于挖掘用户的重要位置,在该处理框架中嵌入了两种执行策略:GPMA 和 SPMA,以保证结果的准确性,降低分析结果的误差.
- 提出 3 种优化措施:(1) 基于外部数据的重要位置修正方法,以提高重要位置的精度;(2) 深入分析无工作地的用户;(3) 分析夜间工作用户的轨迹行为,提出了修正方案.
- 理论分析了所提算法框架的时间复杂度,并在真实数据集上进行了分析.实验结果表明,在准确性上,本文算法比已有最好方法高出 5%~6%,误差降低了 50%;运行效率上也比相关算法提高了 3~5 倍.

本文第 1 节介绍相关工作.第 2 节形式化定义研究问题和若干重要概念.第 3 节详细介绍重要位置挖掘框架、两种嵌入该框架的算法以及针对两种算法所求得的结果.第 4 节从 3 个方面对结果进行修正.第 5 节通过实验验证所提方法的性能.最后,第 6 节总结全文.

## 1 相关工作

数据可用性成为大数据的一个重要方面.近年来,随着大数据的爆炸性增长,数据质量问题得到了广泛的关注.李建中等人从一致性、准确性、完整性、时效性和实体同一性这 5 个方面来考虑大数据的可用性[12].手机轨迹数据的准确度不高,无法准确描述手机每次连接基站时用户所处的位置.文献[13]提出了基于语义的数据精确度评估方法,并给出了一种数据清洗方案.但是这个工作并非针对轨迹数据.

重要位置发现是轨迹数据挖掘的重要内容,可以基于多种轨迹数据进行分析[6−9].文献[6–8]使用社交媒体中用户的签到数据来挖掘用户的重要位置.签到数据的位置精度高,但是覆盖用户规模较小、人群窄,而且还存在数据稀疏性问题(即:签到位置仅集中于少数几个地点).文献[9]使用市民刷卡记录数据来分析居住地和工作地,其覆盖人群仍不够广泛,所获得的结果精度不够高.鉴于手机已得到极大的普及,目前有一些工作集中于使用手机大数据来分析用户的重要位置[5,7,11,14,15].

基于移动轨迹数据进行重要位置挖掘的方法主要分为两类:基于网格统计的方法和基于聚类分析的方法.基于网格统计的方法是最早提出的方法.文献[7,14]先将研究区域栅格化,再将基站位置与栅格相对应,接着统计用户在每个栅格的出现次数,并将次数最多的栅格看作是包含重要位置的栅格.使用栅格的中心点或在该栅格出现位置的平均值作为用户的重要位置所在地.基于聚类分析的方法是目前最新的研究方向.文献[5,15]直接对用户连接过的基站位置点进行聚类,并将聚类中心作为用户重要位置.文献[11]则提出了先聚类再过滤的改进方案.它考虑用户在各簇中的停留时间和次数,从而筛选出潜在合理的簇.如图 1 所示,若使用文献[11]中的方法,由于 c 基站的覆盖范围较广,聚类算法在不同的参数下会将 a,b,c 这 3 点聚为一簇,或者将 a,b 聚为一簇,c 点单独成为一簇.当 a,b,c 聚成一簇时,现有方法会使得结果偏向 c 点,影响结果的精度;当 c 点单独聚成一簇时,又会使用户多出一个"居住地",降低了结果的准确度.同时,对于第 2 个簇中 f 基站虽然满足聚类阈值,但由于在其

有效停留时间段为 11:52 到 12:20,相对于用户在其他基站的停留时间则较短.显然,该基站的重要性对于分析工作地来说没有 d,e 两个高.

基于网格统计的方法简单、易行,但所求结果准确度和精度不高.基于聚类的分析方法虽然考虑了用户在各个簇中的停留持续时间和次数,但未考虑簇中各基站的连接时间和持续时间,所以结果的精度仍有待提高.进一步地,以上方法未考虑用户的多样性等问题,所求得的结果还存在较大偏差.为此,针对低质手机轨迹数据,本文提出了处理框架和挖掘算法,算法不仅有效地找出用户的多个重要位置,还能解释用户的重要位置的变化情况.同时,本文从 3 个方面对结果进行优化,增强了结果的准确性和精度.

## 2 相关定义

### 2.1 问题定义

用户的重要位置是指用户长时间、频繁停留的位置.居住地和工作地是两个最典型的重要位置.本文只关注这两类重要位置.以下是查询定义.

**定义 1(重要位置查询)**. 给定用户轨迹 *tra*、观察时间范围窗口[$T_{begin}$,$T_{end}$]、停留时间阈值 *dur* 和停留次数阈值 *frq*.本查询返回在给定时间窗口范围内连续停留时间超过 *dur* 的次数超过 *freq* 的所有位置.

例如,假设给定的时间窗口范围为晚上 8 点到第 2 天早上 6 点,*dur* 设为 6 小时,*freq* 设为 15.某用户在晚上 8 点到早上 6 点这一时间窗口内,若在某个位置连续停留时间超过 *dur* 的天数为 24(超过 *freq* 阈值),则该位置就是这个用户的一个重要位置,且很可能是居住地.

重要位置查询若是针对精确数据,查询设计仅需一条 SQL 语句就能实现.但考虑到手机轨迹数据中位置的模糊性、基站跳变以及用户的多样性,一条 SQL 语句或现有方法针对精确轨迹数据设计的方法,就无法使用.本文针对低质轨迹数据设计了新的查询框架,下一小节将介绍与查询及本文算法相关的一些重要概念.

### 2.2 查询相关定义

由于基站具有一定的覆盖范围,无法根据用户连接的基站记录确定用户的精确位置,往往只能给出一个区域范围.因此,在本文所提出的框架中,先找出重要位置所在区域范围,然后再找出具体的位置.对于如何找出区域范围,本文总结已有工作,设计了两种区域范围的方法:一种是网格区域表示方法.这种方法将整个用户活动区域连续划分成若干连续尺寸固定的网格,所有基站根据其坐标位置对应到相应网格中.用户连接到某基站,则认为用户在该基站对应的网格中;另一种方法是基站覆盖的表示方法.该方法认为,用户每连接一个基站,则用户实际位置在以基站为圆心、一定长度为半径的覆盖范围内.同时,为了表示用户在某一区域范围的停留时间,引入了“状态”这一概念.对应于两种区域范围表示方法,分别给出了网格状态和基站状态这两个定义.

**定义 2(网格状态)**. 网格状态是指用户在某网格的一次停留情况,它由网格号、进入网格时间和离开网格时间 3 个属性构成.

如图 2(a)所示,当用户 $t_0$ 时刻第 1 次连接 *a* 网格内基站时,会给 *a* 网格生成一个状态,进入和离开时间均为 $t_0$,若用户接着连续多次连接 *a* 网格内基站,则会更新该状态的结束时间,直到用户不再连接.

**定义 3(基站状态)**. 基站状态是指用户对基站的一次连续接入情况,它由基站号、接入起始时间和接入结束时间 3 个属性构成.

与网格状态一样,基站状态也是描述用户在一个区域的活动情况.使用网格状态和基站状态这两个概念,不仅能有效地减小所需处理数据的规模,还能够有效地保持数据的原始信息.由于原始基站数据中大量存在信号跳变这一情况,需要进一步扩大用户所在区域的可能范围,为此提出了“邻居”这一概念.对应于两种区域范围表示方法,我们分别提出了邻居网格和邻居基站这两个定义.

**定义 4(邻居网格)**. 将研究区域划分成若干连续尺寸固定的网格后,若两个网格有共同的顶点,则这两个网格互为邻居网格.


(a) (b) (c)

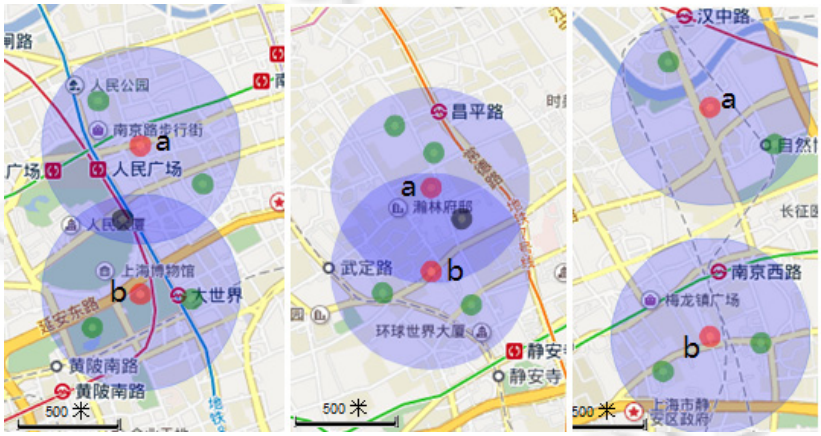
Fig.2 Neighborhood relation between two grids (the length of side of each grid is 500m)

图 2 两网格之间的邻居情况(网格边长 500m)

图 2 中介绍了任意两个网格 *a*,*b* 之间的邻居关系.图 2(a)表示 *a*,*b* 有共同的邻居,但 *a*,*b* 互不为邻居;图 2(b)表示 *a*,*b* 有共同的邻居,且 *a*,*b* 互为邻居;图 2(c)中 *a*,*b* 没有共同的邻居

**定义 5(邻居基站)**. 若两个基站中某一个基站位置位于另一基站的信号覆盖范围内,则这两个基站互为邻居基站.

从定义可以看出,每个基站若以自身位置为中心,其覆盖范围内的基站都是它的邻居基站.与图 2 类似,图 3 给出了任意两个基站间的邻居关系.




(a) (b) (c)

Fig.3 Neighborhood relation between two base stations (the coverage radius is 500m)

图 3 两基站间的邻居情况(基站覆盖半径 500m)

由于数据的不准确性和基站跳变,用户即使处于某一网格或基站所覆盖的范围内,实际所处位置也有可能在该网格或基站的邻居中.因此,每次连接不仅要考虑其当前窗口或基站的状态变化,还要考虑邻居的状态变化.以图 2(a)为例,当用户第 1 次连接 *a* 网格内基站时,不仅会为 *a* 网格生成状态,也会为其邻居生成起始和结束时间与 *a* 一样的状态.当用户持续连接 *a* 网格内基站时,需要对 *a* 及其邻居的状态所对应的结束时间进行更新.进一步地,当用户从 *a* 转移到 *b* 网格时,此时会为 *b* 及其邻居生成状态.特别地,由于 *c* 网格是 *a*,*b* 的共同邻居.此时,则无需为 *c* 生成新的状态,只需更新上一时刻 *c* 网格所对应状态的结束时间.

## 3 重要位置挖掘框架

本文提出的重要位置挖掘处理框架的主要思路是:首先,针对轨迹记录中大量与重要位置无关的噪声数据,

通过状态生成和状态过滤两个步骤消除噪声数据以及基站跳变对分析造成的影响,提高数据的可用性.接着对剩下的位置通过聚类分析,找出用户的重要位置.

### 3.1 算法框架

本框架使用 Map/Reduce 编程模型[16]实现,且仅需一个 Map/Reduce 任务就能实现.由于用户在重要位置的停留情况与时间密切相关,如用户在居住地的时间主要集中在夜间,而在工作地的时间往往在工作日的白天.因此挖掘某一重要位置,需要给出用户出现在该重要位置的时间范围[*begin*,*end*].本框架的 Map 和 Reduce 函数详细过程如下所示.

*map*(*key*,*value*) //*value* 为包含手机号、基站 ID 和连接时间 3 个字段的字符串

1. 从 *value* 中提取出 *phoneID*,*baseID*,*timeStamp* 这 3 个属性;
2. **if** (*timeStamp*>=*begin*||*timeStamp*<=*end*){
3. output (*phoneID*,⟨*baseID*,*timeStamp*⟩);
4. }

*reduce*(*key*,*values*) //*key* 为用户手机号,*values* 为用户的所有夜间连接记录

1. *regionList*=null;
2. *values* 中所有记录按时间排序,并按照给定的时间范围划分成若干子轨迹记录;
3. 对每个子轨迹,结合基站位置,调用状态生成模块,生成状态序列;//参见第 3.2 节
4. 对生成的状态调用状态过滤模块,得到过滤后的结果;//参见第 3.3 节
5. 合并所有子轨迹处理后的状态列表,并将所有状态按照其所对应的区域进行分组;
6. 遍历各区域,若某区域 *g* 所对应的状态数目大于阈值 *freq*,则认为该区域可能包含重要位置,*regionList*=*regionList*∪*g*;
7. 对 *regionList* 调用聚类分析模块,找出用户的所有重要位置及其有效时间.//参见第 3.4 节

Map 函数的功能是将用户在总时间内的满足时间限制的轨迹点发送给 Reduce 任务.Reduce 函数的第 2 步将用户每天的记录作为一个子轨迹.第 3 步调用状态生成算法对子轨迹生成状态.第 4 步调用状态过滤模块删除噪声数据所对应的状态.第 7 步过滤掉那些停留次数较少的区域.比如用户偶尔会去商业中心,虽然在这些位置上连接基站所生成的状态持续时间较长,但由于频率不高,所以要删除在这些位置所对应的状态.第 7 步对前面所找出的区域使用传统聚类算法进行聚类,并从结果中分析出重要位置的地点及有效时间.

### 3.2 状态生成模块

状态生成模块的目的是根据手机连接基站的记录,分析用户在各区域的停留情况,得到用户在各个区域停留所产生的状态列表.该模块首先根据用户选定的范围化方法(网格或基站覆盖范围),将用户的连接基站的历史记录转换成对应的状态序列.在此过程中,若使用网格范围进行区域范围化表示(即用户每一次连接基站,表示用户位于该基站所处网格中),则生成网格状态序列;选用基站覆盖范围进行区域表示(即用户每一次连接基站,表示用户位于该基站信号所覆盖的范围内),则生成基站状态序列.考虑到基站跳变现象及数据精度低这样的问题,当用户每一次连接基站时,用户实际位置仍有可能在当前网格或基站的邻居中.因此,每一次连接时,不仅要对当前网格或基站生成状态,还要生成或修改邻居的状态.下面以生成网格状态(基站状态生成算法类似)为例介绍状态生成算法.

**算法 1**. 状态生成算法.

输入:$Tra\langle objID,p_0,p_1,...,p_n\rangle$为用户从 $t_0$ 到 $t_n$ 时间段内的轨迹数据;

输出:状态列表 *stateList*.

1. *cacheList*=null; //缓存结束时间未定的状态
2. *stateList*=null; //存放结束时间已经确定的状态

3. $a$ =*region*($p_0$); //找出 $t_0$ 时刻连接基站所在网格
4. 对 $a$ 及其邻居构建状态网格状态,并将这些状态加入到 *cacheList* 中;
5. **for** ($i$=1;$i$<$n$;$i$++){
6. $b$=*region*($p_i$); //找出 $t_i$ 时刻连接基站所在网格
7. **if** ($a$=$b$){//$a$,$b$ 对应相同网格
8. 更新 *cacheList* 中 $a$ 及其邻居的状态,将结束时间设为 $t_i$;
9. }**elseif** ($a.neighbors \cap b.neighbors \neq \varnothing$) {//$a$,$b$ 有共同的邻居
10. 更新 *cacheList* 中 $a$,$b$ 共同邻居所对应状态的结束时间;
11. **if** (*a.neighbors.contains*($b$) = false) {//如图 2(a)所示,此时 $a$,$b$ 互不为邻居
12. 将 $a$ 及只属于 $a$ 的邻居所对应的状态从 *cacheList* 中移出并加入到 *stateList* 中;
13. 为 $b$ 及只属于 $b$ 的邻居构建状态,并加入 *cacheList* 中;
14. }**else** {//如图 2(b)所示,此时 $a$,$b$ 互为邻居
15. 将仅属于 $a$ 的邻居状态从 *cacheList* 移到 *stateList* 中;
16. 为 $b$ 邻居中仅属于 $b$ 的网格生成状态,并加入 *cacheList* 中;
17. }
18. }**else** {//如图 2(c)所示,此时 $a$,$b$ 无共同邻居
19. 将 $a$ 及其邻居的状态从 *cacheList* 移到 *stateList* 中;
20. 为 $b$ 及其邻居网格生成状态,并加入 *cacheList* 中.
21. }
22. $a$=$b$; //用 $b$ 更新上一时刻所在网格
23. }
24. 将 *cacheList* 中状态添加到 *stateList* 中;
25. output ⟨*stateList*⟩;

算法 1 以生成网格状态为例,介绍了状态生成算法.其中,步骤 1~步骤 2 分别定义了 *cacheList* 和 *stateList*. *CacheList* 存放状态生成时的中间结果,*stateList* 存放已经生成好的状态.步骤 3 对 $t_0$ 时刻所连基站 $p_0$ 找出其所在 $a$ 网格,步骤 3 对 $a$ 网格及其邻居构建状态,表示用户进入这些网格所包含的区域.若 $t_1$ 时刻还在 $a$ 网格,则按照步骤 8 更新 $a$ 及其邻居状态的结束时间;若 $t_1$ 时刻不在 $a$ 网格,则存在图 2 所示的 3 种情况.其中,图 2(a)和图 2(b)均能表示 $a$、$b$ 有共同邻居,需要按照步骤 10 更新共同邻居所对应状态的结束时间.进一步地,图 2(a)还表示相邻两时刻所在网格互不为邻居的情况,此时按照步骤 12~步骤 13 来处理.图 2(b)为相邻两时刻所在网格互为邻居时的情况,此时需要按照步骤 15~步骤 16 来处理.图 2(c)表示相邻两时刻所在网格没有共同邻居时的情况,此时按照步骤 19~步骤 20 进行处理.步骤 5~步骤 23 循环处理每次连接基站的情况,与前一次进行对比.步骤 24 将处理完后 *cacheList* 中的结果添加到 *stateList* 中.状态生成算法能够有效地消除位置不准确性因素及基站跳变的影响.步骤 25 将生成好的状态输出交给下一阶段分析.

### 3.3 状态过滤模块

状态过滤模块主要是从获得的状态列表中消除“噪声”数据所对应的状态.并通过状态合并来消除基站跳变所带来的影响.下面以对网格状态为例,介绍状态过滤算法,算法具体过程如下.

**算法 2**. 状态过滤算法.

输入:网格状态列表 *stateList*;

输出:网格状态列表 *filterStateList*.

1. 将 *stateList* 中的状态按其所在网格进行分组;
2. 若同一组里的相邻两状态间隔小于某一阈值(30 分钟),则将前一状态的结束时间设为后一状态的结束时间,并删除原来的后一种状态;

3. 删除同一组中持续时间低于阈值 *dur* 的状态;

4. 合并所有组里的状态,得到新的状态列表 *filterStateList*.

5. output ⟨*filterStateList*⟩;

步骤 1 将 *stateList* 中的状态网格进行分组.步骤 2 合并同一组里的状态,进一步消除信号跳变对网格状态的影响.由于步骤 3 过滤掉持续时间较短的状态(如用户上下班途中所产生的状态),从而缩小了搜索范围.

### 3.4 聚类分析模块

聚类分析模块对已找出的包含重要位置的区域进一步缩小搜索范围,找出真正包含重要位置的区域.由于用户同一类型的重要位置有可能同时有多个,如用户可能同时有多个工作地.此外,用户的重要位置会发生改变.因此,聚类算法需要能够自动找出用户的多个重要位置以及重要位置的变化情况.本文使用传统的聚类方法 DBSCAN[17]来进行聚类,并对聚类结果进行深入的分析,以找出每个重要位置及其有效时间.

**算法 3**. 聚类分析算法.

输入:状态列表 *regionList*;

输出:重要位置及该重要位置的有效时间.

1. 将 *regionList* 中所有区域的中心点作为输入,使用 DBSCAN 算法对这些点进行聚类;

2. 若聚类后获得多个簇,则查看簇与簇之间所对应的状态是否存在时间重合的情况.若含有重合的,则删除用户连接次数较少的簇;

3. 对每个簇进作一步分析,获取用户的重要位置;

4. 分析每个重要位置的有效时间范围.

第 1 步对状态所对应的区域使用 DBSCAN 算法聚类,聚得的每个簇代表一个候选的包含重要位置的区域.第 2 步对含有多个居住地(或工作地)的结果进行分析,以避免出现“用户同时停留在两个居住地”的结论.图 1 中,当聚类算法将 c 基站单独聚为一个簇后,由于其与 a,b 所构成的簇中出现时间交叉现象,因此应删除 c 基站所形成的簇.第 3 步从聚类结果中确定用户的重要位置,在此过程中,首先得到用户在该簇中所连接过的基站,得到由这些基站的位置所组成的向量 $P$.接着计算基站的权重向量 $W$,具体做法是先计算在每个基站的总停留时间 $d_i$,接着使用 $w_i = d_i \Big/ \sum_{i=1}^{n} d_i$ 作为该基站的权重.最后使用基站位置向量 $P$ 和对应权重向量 $W$ 的乘积值作为用户的最终位置.第 4 步,从状态列表中计算出用户在每个重要位置的起始停留时间和结束时间,以便刻画用户重要位置的变化情况.

### 3.5 框架性能分析

**定理 1**. 该框架对所有用户的轨迹进行重要位置分析的时间复杂度为 $O(k\times(n+l+m\log m))$.

证明:假设单个用户连接基站的记录数为 $n$,状态生成步骤由于只需遍历一遍所有连接记录数就能生成网格状态序列,所以状态生成算法的时间复杂度为 $O(n)$.对于状态过滤步骤,有 $l$($l$ 的大小与 $n$、网格的边长或基站的覆盖半径相关)个状态需要合并,算法的时间复杂度上界为 $O(l)$.经过前两个步骤处理后,聚类的输入数目为 $m$,使用 DBSCAN 算法的聚类的时间复杂度为 $O(m\log m)$.最后根据聚类结果分析位置的算法复杂度也为 $O(n)$.所以单个用户的算法时间复杂度为 $O(n+l+m\log m)$,对于整个数据集而言,若有 $k$ 个用户,则基于网格的并行挖掘算法的时间复杂度为 $O(k\times(n+l+m\log m))$.

算法若使用网格范围进行范围化表示时,称其为 GPMA 算法;当使用基站覆盖范围进行范围化时,则称其为 SPMA 算法.GPMA 算法中,用户每一次连接所产生的邻居数是固定的,而 SPMA 算法中,由于基站密度不均,所以在基站密集的地方,有的基站的邻居个数能达到几十个.同时,当用户连接的基站发生变化时,若变化前后的基站处于同一网格中,则 GPMA 算法不会产生新的状态,而 SPMA 算法需要对新的基站查找邻居并构建状态.因此,SPMA 比 GPMA 产生更多的状态,同时过滤后所产生的用于聚类的点数也比 GPMA 要多,最终导致 SPMA 算法的运行时间比 GPMA 算法要长.但是,由于 SPMA 算法能够刻画更细范围内用户的活动,因此所获得的结果

精度和准确度高于 GPMA 算法.

## 4 优化措施

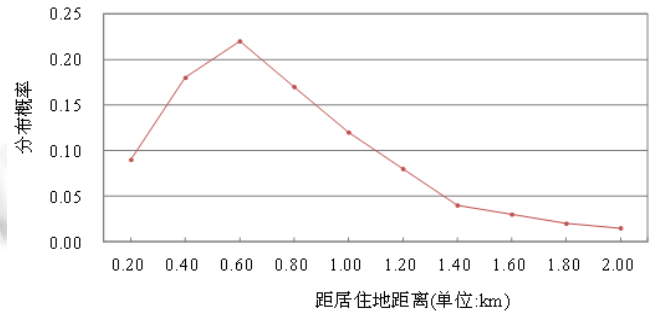
前述算法能够较准确地获得用户的重要位置,但仍存在 3 个问题:首先,所获得的居住地信息可能部分偏移到非住宅区域;其次,可能将无工作人群的居住地误判为其工作地;最后,夜间工作而白天休息的人员的工作地和居住地可能混淆.鉴于此,本节提出了相应的优化措施.

### 4.1 居住地精度调整

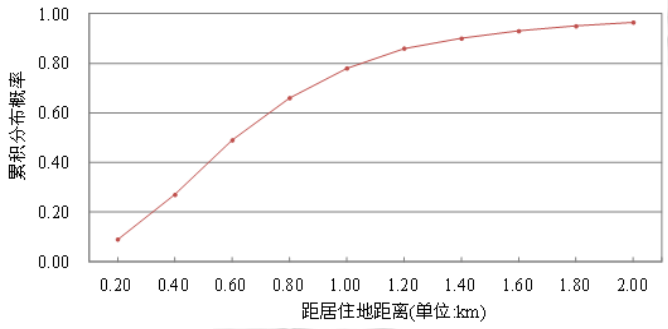
可结合外部信息源来调整用户的居住地信息.通过爬虫程序,爬取“搜房网”和“Q 房网”等房地产信息网站的楼盘信息,包括每个住宅区的位置信息.再为这些楼盘构建 R-tree 索引.当给定一个待修正的居住地时,通过查找最邻近的住宅区,作为修正后的居住地.

### 4.2 无工作人群位置修正

图 4(a)介绍了随机选取的 100 位达到退休年龄的用户在白天连接的基站距离其居住地距离的分布情况.可以看出,这些用户白天连接距其居住地 400m~600m 范围内的基站概率最高.图 4(b)为该概率分布的累积分布曲线图,约 95%的用户连接所对应的基站位于用户居住地 1.8km 范围内.因此可以认为,当一个达到退休年龄的用户,若求得的“工作地”距离其居住地小于 1.8km,则认为该“工作地”只是其白天在居住地周围活动所产生的结果.




(a) 用户连接概率分布函数




(b) 用户连接累积概率分布函数

Fig.4 Distribution function of distance from users' connections to living places

图 4 用户连接基站与居住地距离的分布

### 4.3 夜间工作人群位置修正

分析基站连接日志之后可以发现,此类用户在工作期间连接基站的频率普遍高于在家连接基站的频率,尤其是在用户睡觉期间手机连接基站的频率较低.因此,如果能够检测出用户连接基站频率最低时段,并进一步分

析用户在该时间段主要停留在什么地方,就能判别此类用户.换言之,若用户连接频率最低的时间主要集中在白天,且在此期间所连基站距算法找出的“工作地”较近,那么该用户很有可能就属于夜间工作的用户.

将用户的活动行为看作围绕工作地和居住地这两种位置之间做周期性移动[4,7],并将用户在连接基站时所处位置分别以一定的概率对应于“工作”(*W*)和“在家”(*H*)状态.这个概率不仅由时间决定,还由其对应时刻所在位置确定.如夜间用户处于“在家状态”可能性较大,但是,若某一用户夜间所处位置离其工作地很近,但离其居住地很远,那么该用户当时则以更高的概率处于“工作”状态.因此,我们使用贝叶斯模型判断用户轨迹中某一点(包含时间和空间两个维度)所对应的状态.

$$P(s_t=H\mid x_t=x)=\frac{P[x_t=x\mid s_t=H]\times P[s_t=H]}{P[x_t=x]};\ P(s_t=W\mid x_t=x)=\frac{P[x_t=x\mid s_t=W]\times P[s_t=W]}{P[x_t=x]},$$

其中,$P[s_t=H]$ 和 $P[s_t=W]$ 反映了时间对用户所处状态的影响,$P[x_t=x\mid s_t=H]$ 和 $P[x_t=x\mid s_t=W]$ 反映了用户所处位置(即空间维度)对用户所处状态的影响.

本文使用如下模型来分析时间因素对用户所处状态的影响:

$$N_{H(t)}=\frac{1}{\sqrt{2\pi\sigma_H^2}}\exp\left[\frac{(t-\tau_H)^2}{2\sigma_H^2}\right];N_{W(t)}=\frac{1}{\sqrt{2\pi\sigma_W^2}}\exp\left[\frac{(t-\tau_W)^2}{2\sigma_W^2}\right],$$

$$P[s_t=H]=\frac{N_{H(t)}}{N_{H(t)}+N_{W(t)}};P[s_t=W]=\frac{N_{W(t)}}{N_{H(t)}+N_{W(t)}},$$

其中,$\tau_H$代表用户处于“在家”的平均时间,$\sigma_H^2$ 代表“在家”时间的方差.同理可求出$\tau_W$和 $\sigma_W^2$. $P[s_t=H]$ 代表 *t* 时刻用户处于“在家”的概率.

同时,当一个用户处于“在家”状态时,所连接的基站越靠近居住地,那么用户处于“在家”的可能性越高,对应工作地亦是同理.为此,本文使用如下模型来刻画位置与所处状态的关系:

$$P[x_t=x_i\mid s_t]=\begin{cases}\sim N(u_H,\Sigma_H), s_t=H\\ \sim N(u_W,\Sigma_W), s_t=W\end{cases},$$

其中,$x_t$代表用户 *t* 时刻的位置,$s_t$=*H* 代表当前所处位置意味着用户正处于“在家”;$s_t$=*W* 代表当前用户所处位置意味着用户正处于“工作”状态.$\Sigma_H$ 和$\Sigma_W$ 分别代表用户在“在家”和“工作”两种状态下所连接基站距离,与本文前述算法求出的居住地和工作地距离的协方差矩阵.同理,$u_H$和 $u_W$代表在两种状态下所连基站距离求出的居住地和工作地的距离的平均值.

最终使用如下形式来判断用户在某一时刻点所处状态:

$$l(x)=\frac{P[x_t=x\mid s_t=H]\times P[s_t=H]}{P[x_t=x\mid s_t=W]\times P[s_t=W]}.$$

当 $l(x)\geqslant 1$ 时,证明用户在该时刻点位于“在家”状态;否则,用户处于“工作”状态.

根据以上分析,将每天划分成连续的 *n*(默认 24)个时间段,统计用户在每个时间段内的平均连接频率,找出连接频率非 0 的最低 *k*(默认 3)个时间段.接着分析在这些时间段内,用户处于“在家”和“工作”两种状态出现的比率.若在连接频率最低的时间段内,“工作”状态出现较多(白天休息),则说明该用户属于夜间工作人群,需要交换该用户的居住地和工作地.

## 5 实验结果及分析

### 5.1 实验数据集

本文使用的数据源为上海某移动通信运营商提供的基站访问日志.该数据源为该运营商共 3 211 342 名用户,从 2014 年 8 月 15 日~2015 年 3 月 1 日(共 208 天)期间手机通过连接基站的记录以及基站位置信息,数据源大小约为 1.6TB.用户拨打电话、收发短信、上网等行为均会产生访问日志信息.

此外,本文随机选取了 98 名志愿者的数据进行算法准确性和精度验证.表 1 介绍了这些志愿者年龄的分布

信息.表2介绍了志愿者工作地的分布情况,其中,8.1%的志愿者为退休人员,没有工作地.表3介绍了志愿者居住地分布情况,其中,10.2%的志愿者拥有两个以上的居住地.这98名志愿者共含有241个重要位置.

**Table 1** Distribution of volunteers' age

**表1** 志愿者年龄分布

| 年龄范围 | 18~30 | 30~40 | 40~50 | 50~60 | >60 |
|---|---|---|---|---|---|
| 百分比(%) | 32.7 | 26.5 | 18.4 | 14.3 | 8.1 |

**Table 2** Distribution of volunteers' working place

**表2** 志愿者工作地数目分布

| 工作地数目 | 0 | 1 | 2 | >2 |
|---|---|---|---|---|
| 人数 | 4 | 82 | 12 | 0 |
| 百分比(%) | 4.1 | 83.7 | 12.3 | 0 |

**Table 3** Distribution of volunteers' home place

**表3** 志愿者居住地数目分布

| 居住地数目 | 0 | 1 | 2 | >2 |
|---|---|---|---|---|
| 人数 | 0 | 71 | 17 | 10 |
| 百分比(%) | 0 | 72.5 | 17.3 | 10.2 |

### 5.2 实验环境及相关设置

实验运行在一个Hadoop-2.0.2搭建的云计算平台上,该平台由20台配置一样的节点组成.每个节点运行的操作系统为CentOS Linux version 6.4,处理器为Intel(R) Xeon(R) CPU E5620,主频为2.40GHZ.

对于工作地挖掘,系统只选取了用户在工作日6点~18点的连接记录;对于居住地挖掘,则使用用户每天从22点到次日早晨6点的连接记录.过滤算法中进行状态合并时设置的阈值为30分钟.状态过滤步骤中,考虑到用户在白天可能会有多个工作地,对于工作地求解,将过滤阈值*dur*设为3小时.对于居住地求解将过滤阈值*dur*设为4小时.并行挖掘算法中将在*freq*的值设为60,即用户在重要位置对应的时间窗口内,平均每周应至少停留2次.本文采用DBSCAN算法进行聚类,其核心点的搜索半径等于网格边长或基站的半径,半径范围内至少包含的点数设置为2.

### 5.3 算法准确度和精度评价

算法准确度评价从准确率(Precision)、召回率(Recall)、$F_1$值($F_1$-measure)这3个方面进行考量,精度使用平均误差(mean_error).若算法所求得的位置与用户实际位置之间的距离小于2km(考虑到基站的分布密度不均,及存在信号跳变现象,2km是一个比较合理的值),则认为正确地找出了该用户的一个重要位置.假设算法找出的重要位置个数为*P*,其中正确找出的重要位置个数为*Q*,实际的重要位置个数为*R*,于是有*Precision*=*Q*/*P*,*Recall*=*Q*/*R*,$F_1$-*measure*=2×*PR*/(*P*+*R*).假设实际重要位置为*l*,算法找出的位置为*f*,则

$$mean\_error=\sum_{i=0}^{i<Q}ed(l_i,f_i)/Q,$$

其中,*ed*为*l*与*f*两个位置之间的欧式距离.

表4介绍了算法GPMA和SPMA优化前后的算法性能.其中,GPMA算法的网格边长和SPMA基站半径均设为500m.从表中可以看出,两种算法经过修正后性能显著提高.这是因为,修正了无工作地和夜间工作人员的重要位置后,导致算法找出的重要位置数(*P*)减少,正确找出的重要位置数(*Q*)增加,而实际重要位置数不变.所以准确率和召回率均有所上升,进一步导致$F_1$值提高.同时,由于将用户位置修正到邻近的住宅区内,所求得的重要位置更加靠近实际情况,因此平均误差也有所降低.从表中可以看出,SPMA算法在准确度和精度方面均高于GPMA算法.

表5介绍了在不同网格尺寸下,GPMA算法在志愿者数据上准确度(优化后)的性能比较.从表中可以看出,随着网格尺寸的变小,召回率有所提高,原因是尺寸越小,算法所能找出的正确位置数越多.但准确率有所下降

的原因是,找出的非重要位置数目在所有找出的位置中比重有所增加.总体上讲,网格粒度变小后,$F_1$ 值有所提高,算法准确度得到提升,同时找出的重要位置的平均误差也迅速降低,这意味着找到的结果精度与网格的尺寸密切相关.

**Table 4** Effect of the optimization

**表 4** 优化措施的影响

| 算法 | 准确率(%) | 召回率(%) | $F_1$ 值 | 平均误差(m) |
|---|---|---|---|---|
| GPMA 优化前 | 88.1 | 91.9 | 0.899 | 421 |
| GPMA 优化后 | 94.1 | 95.4 | 0.947 | 325 |
| SPMA 优化前 | 91.2 | 94.3 | 0.927 | 357 |
| SPMA 优化后 | 95 | 96.3 | 0.956 | 289 |

**Table 5** Performance of GPMA on different edge size

**表 5** 不同网格半径下 GPMA 算法性能

| 网格边长(km) | 准确率(%) | 召回率(%) | $F_1$ 值 | 平均误差(m) |
|---|---|---|---|---|
| 2 | 96.4 | 81.3 | 0.882 | 1 215 |
| 1.5 | 95.5 | 86.7 | 0.909 | 974 |
| 1 | 94.7 | 91.2 | 0.929 | 674 |
| 0.5 | 94.1 | 95.4 | 0.947 | 325 |

由于无法获取每个基站的覆盖半径,因此本文将基站覆盖半径作为一个参数.表 6 介绍了在不同半径参数下,SPMA 算法(修正后)的准确度和精度.可以发现,随着半径的缩小,SPMA 算法的性能逐步提高.这与 GPMA 算法在不同网格边长下的结果一致.

**Table 6** Performance of SPMA on different radius

**表 6** 不同基站半径下 SPMA 算法性能

| 半径(km) | 准确率(%) | 召回率(%) | $F_1$ 值 | 平均误差(m) |
|---|---|---|---|---|
| 2 | 98.6 | 84.2 | 0.908 | 1 098 |
| 1.5 | 97.9 | 88.3 | 0.928 | 921 |
| 1 | 96.2 | 92.7 | 0.944 | 584 |
| 0.5 | 95 | 96.3 | 0.956 | 289 |

表 7 为本文提出的两种算法与文献[7,11,15]各自提出的算法在志愿者数据上的准确度比较.实验中,文献[7]中算法使用的网格边长和本文提出的基于网格过滤的并行算法所使用的边长均为 500m.由于无法从运营商处获得每一个基站的覆盖范围信息,本文将基站覆盖范围也设为 500m.从表 7 中可以发现,GPMA 和 SPMA 算法(修正后)的实验结果的准确度比文献[7]和文献[15]更高.文献[11]算法与本文提出的算法性能在 $F_1$ 值上相差不大,但其精度不及本文算法.文献[15]中算法性能最差,这是因为,该算法将用户所有基站一视同仁进行聚类,影响了对用户重要位置的判断.文献[7]效果虽然不是很好,但也反映了一个事实,即大多数用户使用手机最频繁的地方往往集中在重要位置及其附近.在精度方面,本文提出的算法精度也较高.同时 GPMA 算法要比 SPMA 算法的准确度性能要好.

**Table 7** Performance comparison

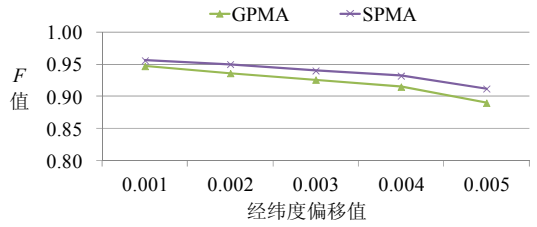
**表 7** 算法性能比较

| 算法 | 准确率(%) | 召回率(%) | $F_1$ 值 | 平均误差(m) |
|---|---|---|---|---|
| 文献[7] | 67.2 | 63.9 | 0.655 | 923 |
| 文献[12] | 41.9 | 71..9 | 0.529 | 1 097 |
| 文献[13] | 90.9 | 90.2 | 0.905 | 731 |
| GPMA | 94.1 | 95.4 | 0.947 | 325 |
| SPMA | 95 | 96.3 | 0.956 | 289 |

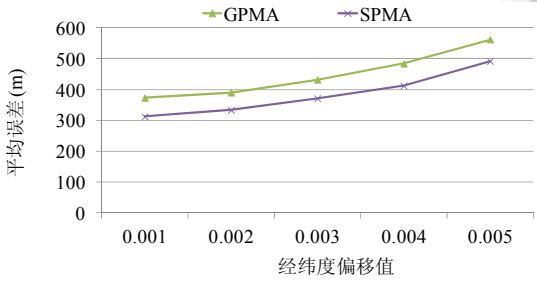
### 5.4 数据质量对分析结果质量的影响

运营商在发布基站位置时,为不暴露其基站真实位置,会在真实位置的基础上增加一些偏移量.在此,本文也考虑了不同数据质量对分析结果质量的影响.实验模拟了在增加不同偏移量的情况下,即进一步降低数据准确度的情况下,验证算法的鲁棒性.横坐标代表经纬度的偏移情况,比如 0.001 表示经度值和纬度值各随机变化

一个[–0.001,0.001]之间任意值(经纬度每变化 0.001,空间距离变化约 100m).从图 5(a)和图 5(b)中可以看出,准确度和精度随着偏移量的增大有所下降,但变化不大.因此,所提方法在低质数据下表现良好.




(a) 不同经纬度偏移后的准确度




(b) 不同经纬度偏移量下的精度

Fig.5 Performance under different deviation values

图 5 不同经纬度下的性能

### 5.5 算法运行性能评价

为了测试网格尺寸对 GPMA 的时间性能的影响,实验在总体数据集进行了不同计算节点下的性能测试,结果如图 6 所示.从图中可以看出,网格尺寸越小,算法所需时间越长.同时,GPMA 算法具有较好的线性加速性.图 7 介绍了在不同基站覆盖半径下,算法的时间性能.从图中可以看出,半径越小,算法的执行效率越高,这与 GPMA 算法恰好相反.这是因为,基站覆盖半径越大,则每个基站所包含的邻居数越多,所生成的状态也越多,从而进一步导致聚类的输入也越多,算法运行时间随之增加.而 GPMA 算法网格边长越大,所对应的网格数越少,最终导致运行时间缩短.

图 8 介绍了 GPMA 和 SPMA 算法与准确度最高的文献[11]中的算法进行并行化后的时间性能对比.该组实验使用 60 台计算节点进行时间性能测试.这 3 种方法均包含聚类这一步骤,但聚类对象各不相同.其中,GPMA 算法对网格进行聚类,聚类前需对网格进行过滤,所以聚类的输入较少,聚类时间很短;SPMA 算法是对过滤后的基站进行聚类,由于在城市内部,基站分布密度较高,且为获取基站的状态需要更多的计算量,这导致 SPMA 算法整体运行时间高于 GPMA 算法;而文献[11]中算法是对用户所有连接过的基站进行聚类,所需聚类的基站个数超过 SPMA 算法,因此聚类运行时间更久.同时,由于文献[11]中算法需要对聚类后的结果使用逻辑回归的方法进行重要性分析,从而找出用户的重要位置,且其运行结果的分析时间也远超本文算法,而本文对聚类结果分析的时间复杂度只有 $O(n)$.因此,总的来说,本文所提出的算法其效率远高于现有算法,且 GPMA 算法虽然准确度没有 SPMA 算法高,但其时间性能更好.

图 9 介绍了 GPMA,SPMA 和文献[11]中 3 种算法在不同节点数目下的运行时间.其中,GPMA 算法中网格边长和 SPMA 算法中基站半径均设为 500m,此时,这两种算法的准确度均达到最高.从该图中可以看出,随着节点数量的增加,这两种算法的运行时间逐渐减少,且 GPMA 的运行时间少于 SPMA 算法.总体而言,本文所提算法具有较好的加速性能.对比图 9 和表 7 可以发现,GPMA 和 SPMA 算法各有优点,GPMA 算法运行较快,而 SPMA 算法准确度更高.

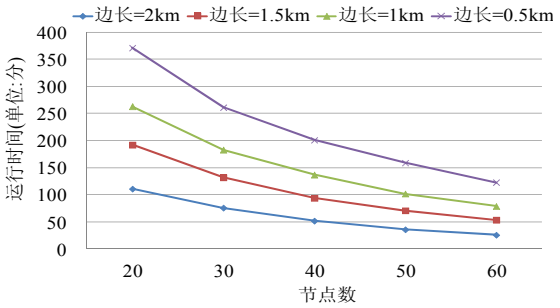



Fig.6 Running time of GPMA

图 6 不同网格粒度下的时间性能

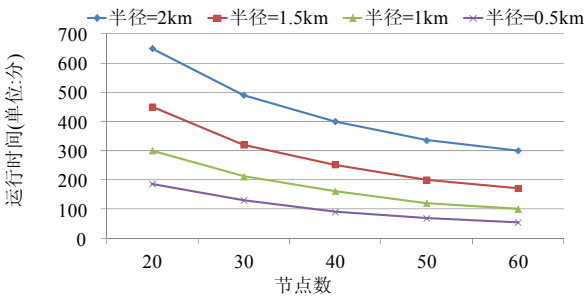



Fig.7 Running time of SPMA

图 7 不同半径下的时间性能

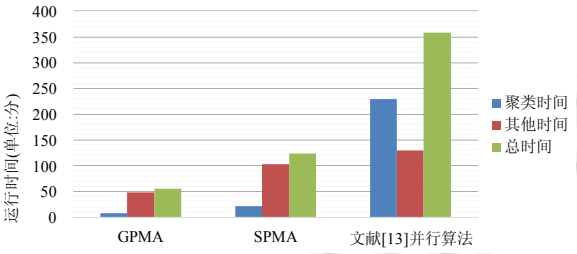



Fig.8 Comparison of running time

图 8 算法运行时间对比图

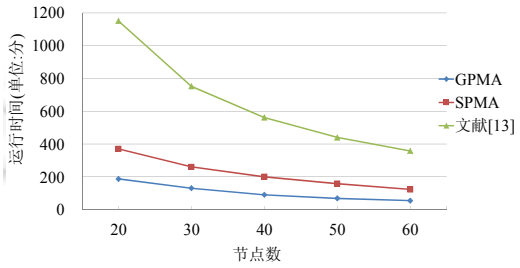



Fig.9 Scalability of proposed algorithms

图 9 算法扩展性

## 6 结束语

本文介绍了从海量低质的手机轨迹数据中挖掘用户重要位置的通用框架,以及基于该框架的两种算法:GPMA 和 SPMA,进一步从 3 个方面修正所挖掘出的重要位置.本文所介绍的两种算法均是先从用户的活动记录中找出可能包含重要位置的区域,再对区域进行聚类,最后从聚类结果中分析出用户的重要位置.两者的区别是,GPMA 使用网格区域来表示可能包含重要位置的区域,而 SPMA 使用基站覆盖范围来表示.相比已有工作,本文围绕重要位置挖掘这一核心点,考虑了如何提高低质轨迹数据的可用性.同时,还融合其他数据以提高结果的精度,并针对两类特殊人群,设计了退休等无工作地人群的发现算法和夜间工作人群挖掘算法,以提高结果的合理性.实验结果表明,本文算法相比已有算法,其结果的 $F_1$ 值更高,位置结果更精确.同时,本文所提两种算法中 GPMA 的时间性能更好,SPMA 的准确度和精确度更高.

**References**:

[1] Gonzalez MC, Hidalgo CA, Barabasi AL. Understanding individual human mobility patterns. Nature, 2008,453(7196):779−782. [doi: 10.1038/nature06958]

[2] Song C, Qu Z, Blumm N, Barabasi AL. Limits of predictability in human mobility. Science, 2010,327(5968):1018−1021. [doi: 10.1126/science.1177170]

[3] Song C, Koren T, Wang P, Barabasi AL. Modelling the scaling properties of human mobility. Nature Physics, 2010,6(10):818−823. [doi: 10.1038/nphys1760]

[4] Li Z, Ding B, Han J, Roland K, Peter N. Mining periodic behaviors for moving objects. In: Proc. of the 16th Int'l Conf. on Knowledge Discovery and Data Mining (SIGKDD 2010). New York: ACM Press, 2010. 1099−1108. [doi: 10.1145/1835804.1835942]

[5] Ashbrook D, Starner T. Using GPS to learn significant locations and predict movement across multiple users. Personal and Ubiquitous Computing, 2003,7(5):275−286. [doi: 10.1007/s00779-003-0240-0]

[6] Sadilek A, Kautz H, Bigham JP. Finding your friends and following them to where you are. In: Proc. of the 5th ACM Int'l Conf. on Web Search and Data Mining. New York: ACM Press, 2012. 723−732. [doi: 10.1145/2124295.2124380]

[7] Cho E, Myers SA, Leskovec J. Friendship and mobility: User movement in location-based social networks. In: Proc. of the 17th Int'l Conf. on Knowledge Discovery and Data Mining (SIGKDD 2011). New York: ACM Press, 2011. 1082−1090. [doi: 10.1145/2020408.2020579]

[8] Bao J, Zheng Y, Mokbel MF. Location-Based and preference-aware recommendation using sparse geo-social networking data. In: Proc. of the 20th Int'l Conf. on Advances in Geographic Information Systems (SIGSPATIAL 2012). New York: ACM Press, 2012. 199−208. [doi: 10.1145/2424321.2424348]

[9] Yuan NJ, Wang Y, Zhang F, Xie X. Reconstructing individual mobility from smart card transactions: A space alignment approach. In: Proc. of the 13th Int'l Conf. on Data Mining (ICDM 2013). IEEE, 2013. 877−886. [doi: 10.1109/ICDM.2013.37]

[10] Wu WQ. 本市人均拥有手机 1.4 部.2014. http://www.jfdaily.com/shanghai/bw/201406/t20140625_481869.html

[11] Isaacman S, Becker R, Cáceres R, Kobourov S, Martonosi M, Rowland J, Varshavsky A. Identifying important places in people's lives from cellular network data. In: Pervasive Computing. Berlin, Heidelberg: Springer-Verlag, 2011. 133−151. [doi: 10.1007/978-3-642-21726-5_9]

[12] Li JZ, Liu XM. An important aspect of big data: Data usability. Journal of Computer Research and Development, 2013, 50(6):1147−1162 (in Chinese with English abstract).

[13] Cheng R, Chen J, Xie X. Cleaning uncertain data with quality guarantees. In: Proc. of the Very Large Data Bases Endowment (VLDB 2008). 2008,1(1):722−735. [doi: 10.14778/1453856.1453935]

[14] Scellato S, Noulas A, Lambiotte R, Mascolo C. Socio-Spatial properties of online location-based socialnetworks. In: Proc. of the 5th Int'l Conf. on Weblogs and Social Media. Palo Alto: AAAI Press, 2011,11:329−336. http://www.icwsm.org/2011/index.php

[15] Chen J, Hu B, Zuo X, Yue Y. Personal profile mining based on mobile phone location data. Geomatics and Information Science of Wuhan University, 2014,39(6):734−738.

[16] Dean J, Ghemawat S. MapReduce: Simplified data processing on large clusters. Communications of the ACM, 2008,51(1):107−113. [doi: 10.1145/1327452.1327492]

[17] Ester M, Kriegel HP, Sander J, Xu X. A density-based algorithm for discovering clusters in large spatial databases with noise. In: Simoudis E, Han JW, Fayyad U, eds. Proc. of the 2nd Int'l Conf. on Knowledge Discovery and Data Mining (SIGKDD 1996). Palo Alto: AAAI Press, 1996,96:226−231. http://www.aaai.org/Press/Proceedings/kdd96.php

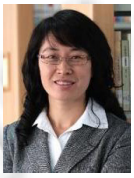
**附中文参考文献:**

[10] 吴卫群.本市人均拥有手机 1.4 部.2014. http://www.jfdaily.com/shanghai/bw/201406/t20140625_481869.html

[12] 李建中,刘显敏.大数据的一个重要方面:数据可用性.计算机研究与发展,2013,50(6):1147−1162.

[15] 陈佳,胡波,左小清,乐阳.利用手机定位数据的用户特征挖掘.武汉大学学报•信息科学版,2014,39(6):734−738.

**章志刚**(1988－),男,江苏海安人,博士,主要研究领域为分布式计算,基于位置的服务.



**王晓玲**(1975－),女,教授,博士生导师,CCF 会员,主要研究领域为半结构化数据管理,数据服务,隐私保护.

**金澈清**(1977－),男,博士,教授,博士生导师,CCF 会员,主要研究领域为海量数据管理,包括基于位置的服务,数据质量,不确定数据管理.

**周傲英**(1965－),男,博士,教授,博士生导师,CCF 杰出会员,主要研究领域为 Web 数据管理,数据密集型计算,内存集群计算,大数据基准测试和性能优化.