

分类号 TP393

学号 14063020

U D C   

密级 公开

## 工学硕士学位论文

# 基于智能手机传感器数据的用户关系计算研究

硕士生姓名 王峰

学科专业 计算机科学与技术

研究方向 移动感知

指导教师 刘东波 研究员

国防科学技术大学研究生院

二〇一六年十月



# **Users Relationship Strength Measurements Based on the Mutuilayer Hybird Model**

**Candidate: Feng Wang**

**Advisor: Professor Dongbo Liu**

**A dissertation**

**Submitted in partial fulfillment of the requirements  
for the degree of Master of Engineering  
in Computer Science and Technology**

**Graduate School of National University of Defense Technology**

**Changsha, Hunan, P. R. China**

**October 10, 2016**



# 独 创 性 声 明

本人声明所呈交的学位论文是我本人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表和撰写过的研究成果，也不包含为获得国防科学技术大学或其他教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示谢意。

学位论文题目： 基于智能手机传感器数据的用户关系计算研究

学位论文作者签名： \_\_\_\_\_ 日期： 年 月 日

# 学 位 论 文 版 权 使 用 授 权 书

本人完全了解国防科学技术大学有关保留、使用学位论文的规定。本人授权国防科学技术大学可以保留并向国家有关部门或机构送交论文的复印件和电子文档，允许论文被查阅和借阅；可以将学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存、汇编学位论文。

(保密学位论文在解密后适用本授权书。)

学位论文题目： 基于智能手机传感器数据的用户关系计算研究

学位论文作者签名： \_\_\_\_\_ 日期： 年 月 日

作者指导教师签名： \_\_\_\_\_ 日期： 年 月 日



# 目 录

|                                |     |
|--------------------------------|-----|
| 摘要                             | i   |
| ABSTRACT                       | iii |
| <b>第一章 绪论</b>                  | 1   |
| 1.1 研究背景                       | 1   |
| 1.2 研究现状                       | 2   |
| 1.2.1 关系强度的研究现状                | 2   |
| 1.2.2 基于移动数据人际关系度量研究           | 3   |
| 1.2.3 基于移动轨迹数据的关系度量研究          | 3   |
| 1.3 研究内容                       | 4   |
| 1.4 论文结构                       | 6   |
| <b>第二章 相关技术研究</b>              | 9   |
| 2.1 用户轨迹预处理                    | 9   |
| 2.1.1 滤波算法                     | 9   |
| 2.1.2 轨迹停留点检测                  | 12  |
| 2.1.3 聚类算法                     | 12  |
| 2.2 时间序列相似度度量方法                | 15  |
| 2.2.1 轨迹中心距离                   | 17  |
| 2.2.2 Dynamic Time Warping     | 18  |
| 2.3 自然语言处理方法                   | 19  |
| 2.3.1 LDA 主题模型                 | 20  |
| 2.3.2 word2vec                 | 20  |
| 2.3.3 快速 Hash 算法               | 23  |
| 2.4 WiFi 和蓝牙数据的度量方法            | 23  |
| 2.5 小结                         | 23  |
| <b>第三章 RSMHD 用户关系强度计算框架模型</b>  | 25  |
| 3.1 RSMHD 模型框架描述               | 25  |
| 3.1.1 基于轨迹数据的关系强度计算模块概述        | 25  |
| 3.1.2 基于 WiFi 和蓝牙数据的关系强度计算模块概述 | 27  |
| 3.2 RSMHD 模型计算流程概述             | 27  |
| <b>第四章 RSMHD 框架的相关技术研究</b>     | 29  |
| 4.1 GPS 轨迹处理计算技术               | 29  |

---

|                                     |           |
|-------------------------------------|-----------|
| 4.2 GPS 轨迹预处理与语义化 .....             | 30        |
| 4.2.1 剔除轨迹中的异常点 .....               | 30        |
| 4.2.2 轨迹中停留点检测 .....                | 32        |
| 4.2.3 用户轨迹停留点聚类 .....               | 34        |
| 4.2.4 对用户轨迹的语义位置添加语义标签 .....        | 40        |
| 4.3 用户 GPS 轨迹数据结构化表示 .....          | 43        |
| 4.4 WiFi 感知数据处理计算技术 .....           | 45        |
| 4.5 蓝牙感知数据处理计算技术 .....              | 46        |
| 4.6 小结 .....                        | 48        |
| <b>第五章 RSMHD 用户关系强度计算方法概述 .....</b> | <b>49</b> |
| 5.1 用户关系强度计算方法概述 .....              | 49        |
| 5.2 RSMHD 框架模型的数据输入 .....           | 50        |
| 5.2.1 轨迹数据的预处理 .....                | 50        |
| 5.2.2 用户空间 GPS 轨迹数据处理和输入 .....      | 50        |
| 5.2.3 用户语义轨迹数据的处理和输入 .....          | 51        |
| 5.2.4 用户运动模式的处理和输入 .....            | 51        |
| 5.2.5 用户 WiFi 感知数据的处理和输入 .....      | 52        |
| 5.2.6 用户蓝牙感知数据的处理和输入 .....          | 52        |
| 5.3 RSMHD 计算用户支之间的关系强度 .....        | 53        |
| 5.3.1 基于用户轨迹数据的关系强度计算 .....         | 53        |
| 5.3.2 基于用户 WiFi 感知数据的关系强度计算 .....   | 56        |
| 5.3.3 基于用户蓝牙感知数据的关系强度计算 .....       | 56        |
| 5.3.4 关系强度计算结果融合 .....              | 56        |
| 5.4 小结 .....                        | 58        |
| <b>第六章 实验数据集与实验结果分析 .....</b>       | <b>59</b> |
| 6.1 实验数据集介绍 .....                   | 59        |
| 6.2 实验结果验证方法 .....                  | 61        |
| 6.2.1 获取用户关系强度结果 .....              | 61        |
| 6.2.2 实验结果评价方法 .....                | 62        |
| 6.3 实验结果与分析 .....                   | 63        |
| 6.3.1 基于 GPS 轨迹相似性计算用户之间的关系强度 ..... | 64        |
| 6.3.2 基于 WiFi 感知数据计算用户之间的关系强度 ..... | 66        |
| 6.3.3 基于蓝牙感知数据计算用户之间的关系强度 .....     | 67        |
| 6.3.4 基于 RSMHD 计算用户之间的关系强度 .....    | 68        |

---

|                            |           |
|----------------------------|-----------|
| 6.4 小结 .....               | 68        |
| <b>第七章 结束语 .....</b>       | <b>69</b> |
| 7.1 工作总结 .....             | 69        |
| 7.2 工作展望 .....             | 70        |
| <b>致谢 .....</b>            | <b>73</b> |
| <b>参考文献 .....</b>          | <b>75</b> |
| <b>作者在学期间取得的学术成果 .....</b> | <b>81</b> |



## 表 目 录

|                                   |    |
|-----------------------------------|----|
| 表 6.1 StarLog 收集的上下文数据种类和类型 ..... | 60 |
| 表 6.2 真实结果 .....                  | 63 |
| 表 6.3 优化 DTW 方法得到好友列表 .....       | 65 |
| 表 6.4 示例用户关系强度矩阵 .....            | 65 |
| 表 6.5 优化 DTW 方法得到好友列表 .....       | 66 |
| 表 6.6 优化 DTW 方法得到好友列表 .....       | 67 |



## 图 目 录

|                                                         |    |
|---------------------------------------------------------|----|
| 图 2.1 用户 GPS 轨迹示例图 .....                                | 12 |
| 图 2.2 聚类效果示意图 .....                                     | 13 |
| 图 2.3 DBSCAN 密度直达和密度可达示意图 .....                         | 14 |
| 图 2.4 关于 decision graph 的示例 <sup>[42]</sup> .....       | 16 |
| 图 2.5 聚类中各点的 $\rho$ 和 $\delta$ 取值 <sup>[42]</sup> ..... | 16 |
| 图 2.6 轨迹相似计算示例 <sup>[43]</sup> .....                    | 17 |
| 图 2.7 基于 Center of mass 轨迹计算示例 <sup>[43]</sup> .....    | 18 |
| 图 2.8 DTW 算法匹配序列结果示意图 .....                             | 19 |
| 图 2.9 LDA 的图模型表示 .....                                  | 21 |
| 图 2.10 word2vec 神经网络结构图 <sup>[48]</sup> .....           | 21 |
| 图 2.11 word2vec 神经网络结构图 .....                           | 22 |
| 图 3.1 RSMHD 用户关系强度计算模型框架图 .....                         | 26 |
| 图 3.2 RSMHD 中基于用户轨迹计算模块详细框架图 .....                      | 26 |
| 图 3.3 RSMHD 中基于 WiFi 和蓝牙计算模块详细框架图 .....                 | 27 |
| 图 4.1 基于多层次多粒度轨迹的用户关系强度计算 .....                         | 29 |
| 图 4.2 轨迹中值滤波实验结果 1 .....                                | 30 |
| 图 4.3 轨迹中值滤波实验结果 2 .....                                | 30 |
| 图 4.4 轨迹均值滤波实验结果 1 .....                                | 31 |
| 图 4.5 轨迹均值滤波实验结果 2 .....                                | 31 |
| 图 4.6 卡尔曼滤波实验结果 1 .....                                 | 32 |
| 图 4.7 卡尔曼滤波实验结果 2 .....                                 | 32 |
| 图 4.8 分段卡尔曼滤波轨迹结果 1 .....                               | 32 |
| 图 4.9 分段卡尔曼滤波轨迹结果 2 .....                               | 33 |
| 图 4.10 现实生活中的停留点示意图 .....                               | 33 |
| 图 4.11 停留点检测实验结果 1 .....                                | 33 |
| 图 4.12 停留点检测实验结果 2 .....                                | 34 |
| 图 4.13 基于 K-means 的轨迹聚类实验结果 1 .....                     | 35 |
| 图 4.14 基于 K-means 的轨迹聚类实验结果地图展示 1 .....                 | 35 |
| 图 4.15 基于 K-means 的轨迹聚类实验结果 2 .....                     | 36 |
| 图 4.16 基于 K-means 的轨迹聚类实验结果地图展示 2 .....                 | 36 |
| 图 4.17 基于 DJ-Cluster 的轨迹聚类实验结果 1 .....                  | 37 |
| 图 4.18 基于 DJ-Cluster 的轨迹聚类实验结果地图展示 1 .....              | 37 |

|                                            |    |
|--------------------------------------------|----|
| 图 4.19 基于 DJ-Cluster 的轨迹聚类实验结果 2 .....     | 38 |
| 图 4.20 基于 DJ-Cluster 的轨迹聚类实验结果地图展示 2 ..... | 38 |
| 图 4.21 轨迹聚类实验结果展示 1 .....                  | 39 |
| 图 4.22 轨迹聚类实验结果地图表示 1 .....                | 39 |
| 图 4.23 轨迹聚类实验结果展示 2 .....                  | 39 |
| 图 4.24 轨迹聚类实验结果地图表示 2 .....                | 40 |
| 图 4.25 位置语义标签示意图 .....                     | 41 |
| 图 4.26 语义标签识别示意图 .....                     | 42 |
| 图 4.27 语义标签结果示意图 .....                     | 42 |
| 图 4.28 用户轨迹的抽象表示 .....                     | 44 |
| 图 4.29 用户语义轨迹描述模型 .....                    | 44 |
| 图 4.30 语义轨迹示意图 .....                       | 45 |
| 图 4.31 WiFi 数据结构化表示示意图 .....               | 46 |
| 图 4.32 WiFi 上下文抽象化表示示意图 .....              | 46 |
| 图 4.33 WiFi 图结构抽象化表示 .....                 | 47 |
| 图 4.34 蓝牙数据结构化表示示意图 .....                  | 47 |
| 图 4.35 蓝牙数据上下文抽象化表示示意图 .....               | 47 |
| 图 4.36 蓝牙数据图结构抽象化表示 .....                  | 48 |
| 图 5.1 用户语义轨迹快速 hash 变换计算示意图 .....          | 54 |
| 图 6.1 学生志愿者信息 1 .....                      | 59 |
| 图 6.2 学生志愿者信息 2 .....                      | 60 |
| 图 6.3 学生志愿者数据采集说明 .....                    | 61 |
| 图 6.4 基于不同时间粒度的实验结果 .....                  | 64 |

## 摘要

近几年，随着传统手机逐渐被智能手机所取代，搭载了智能操作系统(如 iOS、Android 和 Windows Phone)的智能手机已经成为了人们生活中集日常通信、娱乐游戏、商务办公、感知计算等于一体的移动掌上终端平台。通过搭载了更多传感器设备的智能手机能够随时随地的获取到用户的位置、通信记录、短信记录、日常轨迹分布情况等各种体现用户与用户之间的隐藏社会关系的感知信息。人们之间的轨迹相似性、日常轨迹的共现频率和时长，用户连接 Wi-Fi 的共现使用情况以及用户蓝牙之间的交互信息都能够分析得出人与人之间的交互关系以及他们之间的关系强度。通过对用户之间的相似性和关系强度的探究和计算有利于进一步发展个人的社交网络和探究社会群体结构的发展以及演变过程等。

传统的度量用户之间的社交关系大多采用的是基于社交关系数据来分析用户之间的社会关系(社交关系数据如通话记录，短信记录，社交软件的交互等)，本文基于智能手机所采集的非社交关系数据针对如何使用非社交关系数据来分析度量人们社交活动之间的相似性以及关系强度问题展开了进一步的深入研究。设计并且实现了一个基于用户轨迹数据、Wi-Fi 感知数据、蓝牙感知数据抽象出的多个层次的度量人与人之间在社交活动中相似性以及关系强度的计算模型 RSMHD(Relationship Strength based on Multiple Hierarchy Dimension)。整个计算模型的主要涵括的内容如下：

为了便于研究的顺利开展，我们假设陌生人之间的关系强度应该是无限趋近于零的，同时我们为了便于对计算结果的准确性进行判断和数据的采集，本文研究对象主要限于在校学生群体。社会心理的理论支持以及研究成果指出：相似的人更容易产生交集成为朋友，现实生活中人们也更加倾向于同自己相似的人做朋友，也就是说相似导致喜欢，从而发展为友情或者爱情。这也就预示着相似度人与人之间的关系强度要高于非相似着，这也就为我们的研究提供了理论依据。

首先，针对用户日常轨迹数据我们分别从空间移动轨迹和现实语义轨迹出发，对用户之间的相似性进行计算，本文采用位置漂移修正算法对部分因无信号原因导致的 GPS 位置缺失的轨迹进行预测补全；针对补全后的用户轨迹，采用时间片式的卡尔曼滤波算法剔除用户 GPS 轨迹中的异常点；采用基于时间 - 密度的聚类算法得到用户空间轨迹中的停留点，在此基础上，采用基于语义规则的语义标签标注机制，通过用户参与，反地理编码等手段对停留点进行现实语义标注，将用户的空间轨迹转变为符合现实意义的日常语义轨迹，为利用位置感知数据来

计算用户的关系强度做好准备。然后针对用户空间语义轨迹，本文采用了快速 DTW 计算方法来计算用户空间轨迹之间的相似性，并采用动态加权算法对计算结果进行加权处理；针对日常语义轨迹，我们分别从两个层次出发，一方面采用了 Word2vec 来计算用户在切片时间内的语义轨迹的相似性；另一方面，根据用户的语义轨迹挖掘出用户的轨迹运动模式，通过快速计算相似度算法 Simhash 来计算出用户的日常轨迹运动模式之间的相似性并对结果采取加权处理得出用户之间的关系强度，最后对计算结果进行融合来得出用户的关系强度。

其次，针对用户 WiFi 感知数据，采用关联拓扑图表示某个时刻的用户 WiFi 上下文环境信息，然后提取出用户 WiFi 的交互情况作为计算用户关系强度的特征之一；针对时间片上的拓扑图创新的提出了通过计算图与节点之间的相似性作为某时刻 WiFi 上下文环境信息之间的相似性。针对用户的蓝牙感知数据，通过结合脸呀交互时长、交互频率、蓝牙上下文环境相似性等来计算出用户之间的关系强度。

最后，利用集成学习的思想对以上三种不同感知数据源计算结果进行融合处理，得到基于不同非社交数据源所计算得到的最终用户关系强度。在该模型算法研究的基础上，本文基于自主开发的用户上下文信息收集系统 StarLog 收集了多名学生成长时间的智能手机感知数据，利用该数据源来计算出用户之间的关系强度，并结合用户的相似度问卷调查表对结果进行有效性验证，表明该模型能够有效地利用非社交关系度量出用户之间的关系强度。

**关键词:** 关系强度；非社交数据；轨迹模式；word2vec；集成学习

## ABSTRACT

Smart phones have become an integral part of daily life communication tools, we can collect intelligently location, call logs, text messages, WeChat anywhere which reflect a variety of information daily interactions and social relations between people. People interaction frequency, time, location, distance and the similarity of trajectory information reflects the strength of the relationship and the relationship between people. Relationship strength reflects the degree of intimacy between two different persons, which is of great importance in analyzing human's social relationship as well as social network.

In this paper, we propose a URSHV(User Relationship Strength Hierarchy Vote), which can measure the relationship between people in daily life through GPS data from three levels, namely: daily trajectory, semantic locations and semantic labels. To sum up, the main research contents and contributions are as follows:

First of all, the strength of the relationship between users and semantic location with semantic labels are closely related, this paper uses segmented kalman filtering algorithm on GPS trajectory data to de-noising; using the clustering algorithm based on density of position trajectory data clustering, and form a semantic position; on this basis, the semantic annotation mechanism based on Rules, the semantic annotation of semantic encoding, geographic location by anti semantic label inference and input auto completion etc; the GPS position trajectory data sequence clustering into meaningful semantic position and semantic labels, which laid the foundation for the semantic location and semantic labels based on the strength calculation of the relationship between users.

Secondly, in order to calculate the strength of the relationship between users from the three levels of trajectory data, semantic location and semantic labels, using DTW model of spatial distance calculation between users to measure the similarity between the users, the use of trajectory sequence similarity of users every day to track the entropy weighting processing, and the strength of relationship between users the LDA were calculated using the topic model; semantic locations similarity and semantic labels based on behavior patterns among users, as the strength of the relationship between users; measurement results of three levels of the ensemble learning theory to vote, to vote as the strength of relationship between end users.

Finally, on the basis of the above study, based on the MIT reality mining project of publicly available data sets, similarity between users by using the data set of the questionnaire, users construct between a real relationship strength as a benchmark for testing proposed an inverse logarithmic induced score measurement method to measure the strength of the relationship between users based on, and effectiveness model of URSHV for experiments, the results show that the model can effectively measure the strength of the relationship between users.

**Key Words:** relationship strength; trajectory similarity; DTW; entropy; LDA; vote

## 第一章 绪论

近年来，随着传统手机逐渐被智能手机取代，搭载了智能操作系统（如ios,Android,Windows Phone）的智能手机已经成为人们日常生活中集通信，娱乐游戏，商务办公，感知计算等于一体的移动个人终端平台。随着传感器工艺界的发展以及内置手机传感器种类的增加，如今智能手机已经能够基于传感器为用户提供良好的用户体验与服务，因此智能手机被科技赋予了更多的责任，智能手机不仅仅是简单的日常通讯工具，还是能够在记录用户的日常活动轨迹，了解用户的使用习惯，为用户提供实时信息推荐，俨然已经成为了一种新型的可穿戴计算<sup>[1]</sup>的载体，随着传统社交方式逐渐转变为手机上的社交方式，借助现有的数据分析和数据挖掘技术，通过分析手机丰富的上下文信息（Context）来研究用户社交关系以及人与人之间的社会关系已经成为移动计算中一个研究热点。

### 1.1 研究背景

近年来，随着传统手机逐渐被智能手机取代，搭载了智能操作系统（如ios,Android,Windows Phone）的智能手机已经成为人们日常生活中集通信，娱乐游戏，商务办公，感知计算等于一体的移动个人终端平台。通过丰富的内置传感器，我们可以收集到更加丰富的上下文信息（context） 分析出人与人之间的交互情况以及相似性<sup>[2-5]</sup>，以及通过用户轨迹数据分析用户的社交动态<sup>[6]</sup>。关系强度在本文中表现为用户在现实生活中的亲密程度，通过掌握用户之间的关系强度有利于合理的拓展社交网络和提升社交质量。文献 [3] 通过对手机传感器数据分析得出不同社会关系群组的用户在日常社交活动中的规律。CenceMe<sup>[7]</sup> 通过收集智能手机传感器的上下文信息以及用户的社交应用（如 Facebook，MySpace 和即时通讯工具 Skype）的使用信息，分析用户的位置、活动、情绪和周围环境并将所有信息通过社交应用来分析用户的社会交互情况但是并未根据社交情况去推测用户之间的社交关系等。Dartmouth 大学学者们的研究中<sup>[8,9]</sup>，通过对班级 48 名学生成达 10 周所采集的手机传感器数据进行研究从中分析出学生之间的社交活动有助于减轻个人压力、保护精神健康和提升学业成绩的作用，研究人与人之间的社交活动和关系强度也因此具有重要的研究意义。美国社会学家格兰洛维特在研究过程中<sup>[10]</sup> 首次提出了关系强度这一概念，研究将关系强度分为强关系和弱关系。同时指明能够充当信息传递载体的纽带关系必定是弱关系，而强关系只是存在于那些你真正充分信任的人之间；强关系存在与和你更加相似的用户之间。Caroline<sup>[11]</sup> 等进一步证实了用户交互的频率以及交互的持久度决定了用户之间的关系强度，如具有较高亲密度和交互频率用户之间的关系强度要高于偶然交互和非持久交互

的用户。文献 [12–14] 基于在线社交网络 (Facebook、Twitter、LinkedIn、Instagram 等) 利用社交数据、个人资料、状态互动等分析用户之间的亲密度，结合机器学习方法采用决策树算法、MLP 算法、SVM 算法等进行用户之间关系强度的预测以及分类。以上提及的研究大都是基于用户社交数据进行的关系强度计算研究，而基于智能手机收集的非社交的传感器数据进行人与人之间关系强度的度量仍是一个值得研究的问题。

随着各种定位技术（如全球定位系统 GPS 和 WLAN 和移动网络）的发展和内嵌的定位模块，通过智能手机可以准确的获取用户的位置信息，提供各种基于位置的服务（LBS），并将用户的活动以轨迹的形式记录下来。我们阅读了大量社会心理学相关的论文书籍，从中证实得到在现实生活中关系亲密的两个用户会更加倾向于一起进行面对面的交流、共同进行社交活动等，因此通过对手机传感器数据的处理分析能够从中挖掘出人们现实生活中的关系强度。文献 [15] 立足于空间距离，提出了在空间距离上非常接近的人们在现实生活中就越可能发生面对面的交互，该文献通过调研一个小区的住户发现人们在现实生活中越是接近，就越容易成为朋友。文献 [16, 17] 进一步通过研究用户的轨迹数据发现由于在空间距离中接近的用户在现实生活中更可能产生交互行为，也就是说拥有相似日常生活轨迹的用户更可能产生交互活动。郑宇 [18] 以及其他学者 [19–21] 尝试了基于用户轨迹计算用户相似性的研究，得出相似轨迹的用户更可能成为朋友，因此，针对相似用户进行朋友关系推荐。基于以上的理论基础和研究经验本文通过收集智能手机获取到的用户位置、WiFi、蓝牙、通话记录等手机上下文数据，并且从这些数据中分析计算出现实生活中用户的交互位置、交互时间、交互频率以及交互的持续性等一系列能够反射出用户间关系强度的信息特征。

目前智能手机增加了多种传感器用于实现更加丰富的用户交互功能，由以前单一的加速度传感器，距离传感器逐步集成了压力传感器，温度传感器，心率传感器等。这使得智能手机能够更加准确地感知到更加丰富多样的周围环境信息如：用户位置，社交通讯记录，WiFi 和蓝牙连接记录等体现人与人之间交互情况以及轨迹交互情况等体现用户关系强度的数据，通过收集分析这些非社交数据，我们能够进一步得出人与人之间的关系相似性以及关系强度。

## 1.2 研究现状

### 1.2.1 关系强度的研究现状

Granovetter 关于弱关系的研究奠定了社会关系强度理论的基础<sup>[10]</sup> 被认为是社会关系理论研究的开始的标志，紧接着 Burt 根据 Granovetter 的弱关系理论研究提出了结构洞理论<sup>[22]</sup>。Granovetter 针对弱关系和强关系的度量方法提出了基于四

种维度的度量准则，即用户之间的日常互动次数、人与人之间的亲密程度、双方投入感情的程度以及用户日常的交换程度，基于这四条衡量标准就可以将用户的关系强度划分为弱关系和强关系；在 Wegner<sup>[23]</sup> 的研究中，更是对这四种度量标准进行了进一步的研究推进，采用数值化来衡量四种维度的标准使得能够以指标数值化来区分强关系和弱关系；Muncer 等<sup>[24]</sup> 提出并验证了用户的关系数量以及任意关系之间的交互频率对人与人之间的关系有影响；Paolillo<sup>[25]</sup> 从日常用户交流的角度出发，发现人与人之间关系的亲密程度与日常交流中昵称使用的频率有关。随着研究的进一步深入，度量用户的关系强度逐步形成了基于感知用户的社交数据出发，以用户交互、亲密度等出发为度量标准的研究观点<sup>[26]</sup>。

### 1.2.2 基于移动数据人际关系度量研究

Hsu 通过采集到的志愿者的手机日常 WiFi 数据<sup>[27]</sup>，将位置与 WiFi 信息关联起来得到用户的 WiFi 关联向量，并用关联向量来表示用户的行为轨迹，同时基于提出的 AMVD 模型用来计算人与人之间的距离，最后根据距离对社交关系进行聚类对社会关系进行划分，文中认为当连接过相同 WiFi 的情况下可以认为两个人在现实生活中有较强的社交关系。并且根据 WiFi 所对应的语义位置（如图书馆、教室、咖啡厅、会议室等）推测他们之间可能的社交关系。基于蓝牙感知信息，研究者通过分析手机收集的蓝牙数据对用户的社交圈进行划分，将用户的社交圈划分为室友，好朋友，工作伙伴等<sup>[28-30]</sup>。Mtibaa 等<sup>[31]</sup> 通过收集分析了 28 位参加同一个计算机国际会议参会者的手机蓝牙数据，根据分析结果绘制了关于 28 位作者的社交网络关系图。

### 1.2.3 基于移动轨迹数据的关系度量研究

在现实生活空间中的用户交互能够更加直接的反映出社会关系的情况，如面对面的交流，共同用餐等，这些现实生活中的用户交互相对于用户的传统社交数据能够更加真实的反映出用户二者之间的关系。但是目前在这方面的研究还仅仅局限于某个局部的方面，文献 [32] 创造性的提出了根据用户的日常轨迹来衡量用户之间的关系强度。Ma 等人<sup>[32]</sup> 提出了一种根据多层基于用户轨迹的层级熵关系度量方法 HERMA(Hierarchical Entropy-based Relationship Measurement Approach)，该模型根据手机收集的用户的 GPS 位置轨迹信息进行处理，从用户轨迹中提取出共同的位置记录来推断用户之间的物理交互，进一步使用用户之间的物理交互来度量用户之间的社会关系强度，最后在仿真的数据集上进行验证。

### 1.3 研究内容

本研究课题针对如何通过智能手机所收集到的用户之间的非社交数据来计算度量人与人之间的社交关系强度展开了一系列的研究，寻求通过建立一个能够同时计算处理多种不同非社交数据源数据的用户关系度量框架。以此为基础来衡量用户之间的关系强度。经过前期的研究，决定将手机感知数据中的非社交关系数据（用户轨迹数据、用户蓝牙数据、用户 WiFi 数据）作为衡量用户关系强度的数据源。为了实现基于不同感数据多维度计算用户之间的关系强度，本文分别针对每一个独立数据源展开研究，最终采用集成学习的思路将结果进行融合。

#### (1) 如何基于用户日常轨迹度量关系强度

随着各种定位技术（如全球定位系统 GPS 和 WLAN 和移动网络）的发展和内嵌的定位模块，通过智能手机可以准确的获取用户的位置信息，提供各种基于位置的服务（LBS），并将用户的活动以轨迹的形式记录下来。我们阅读了大量社会心理学相关的论文字籍，从中证实得到在现实生活中关系亲密的两个用户会更加倾向于一起进行面对面的交流、共同进行社交活动等，因此通过对手机传感器数据的处理分析能够从中挖掘出人们现实生活中的关系强度。在获取用户位置的过程中，其结果既可以由智能手机内置的 GPS 传感器提供，还可以通过基站和 WiFi 定位获取。用户的每一个位置点都是由一个三元组结构 (Latitude,Longitude,Time) 组成的，用户在一段时间内连续的位置记录就构成了一条完整的连续轨迹。根据用户的日常语义轨迹，采用滤波技术手段剔除用户轨迹中的异常点，同时利用轨迹预测算法对短时间窗口内的用户位置进行有效地补充。针对用户的轨迹从不同层次出发，计算不同轨迹形态下用户之间的关系强度加以合并。

##### a) 基于空间轨迹的用户关系度量

用户的空间轨迹中包含了一串由 (Latitude,Longitude,Time) 所组成的，寻找出用户空间轨迹中的特殊点即停留点，停留点<sup>[33]</sup>(Stay Point) 在现实生活中并不是指用户轨迹中速度为零的点，而是由一组 GPS 点构成区域，表示用户在这段轨迹中在某一个区域停留的时间超过了设定的阈值。它比传统的 GPS 位置点蕴含了更加重要的信息在，如该用户去过图书馆、体育馆等。因此用户的空间轨迹的距离现在一定程度上体现了用户之间的亲密关系，本文正是基于此尝试采用空间轨迹距离来度量人与人之间的社交关系强度。传统的距离计算主要使用欧氏距离、曼哈顿距离、马氏距离等。但是，在使用欧氏距离来度量用户空间轨迹的相似性过程中，轨迹之间的距离会受到轨迹长度的影响，导致结果出现较大偏差。通过结合传统语音识别算法 DTW<sup>[34]</sup>，降低用户空间轨迹长度对计算结果的影响；考虑到

DTW 计算结果受到序列长度的影响，轨迹序列越长，得到的轨迹距离结果越大，因此对 DTW 计算结果采用合理的归一化处理以消除不同用户轨迹长度所带来的差异性影响。

b) 基于语义轨迹的用户关系度量

基于空间轨迹的度量能够反映出用户在地理空间上的相遇或者相邻，基于语义轨迹的度量能够进一步得出用户之间的关系强度。借助自然语言处理思想，把用户每天的语义轨迹作为一条自然语句，用户所有时间段内的  $n$  条轨迹记录就生成了一篇文档，最后通过计算用户语义轨迹生成的文档之间的相似性来表示用户之间的关系强度。通过将语义标签作为分词后的结果，利用 hash 函数计算每个分词特征向量的 hash 值，然后根据词频对每个 hash 特征向量进行加权合并，最后经过计算两条语义轨迹的海明距离即可表示为语义相似度。比较了此方法和传统的主题模型以及 word2vec 模型之间的性能和效率结果。

c) 基于日常轨迹模式的用户关系度量

人们的日常活动具有很强的时间和空间的规律性，在计算用户相似度或者关系强度的研究中，还没有出现过从用户轨迹模式角度出发计算用户相似度和关系强度的研究。González 等人通过对大量用户轨迹数据的分析发现用户在日常生活中常有规律的访问相似的路径，说明用户的轨迹运动模式能反映出用户之间的相似性<sup>[35]</sup>。从用户的历史轨迹数据中所挖掘出的频繁模式和序列模式能够反映出用户的日常轨迹运动习惯和行为规律，运动模式在现实生活中表现为用户经常行走的路径序列，是用户轨迹数据规律的抽象表示。用户的日常运动模式在一定程度上代表了用户的个人喜好、意图以及活动模式，例如用户 A 经常下午去操场跑步，B 周末经常去市区逛街等，当从一个更细的粒度甚至能够根据用户的用餐地点推测出用户的口味喜好等。本研究首先采用频繁模式和序列模式挖掘算法来探寻用户的日常轨迹运动模式，然后根据计算用户运动模式之间的相似性得出人与人之间的关系强度。

(2) 如何基于用户 WiFi 感知数据度量关系强度

随着无线网络的覆盖普及，WiFi 已经覆盖了我们日常生活中的每个区域，例如在家中、在商场中、公交车等都能够随时便捷的接入 WiFi。在使用 WiFi 进行网上活动的时候，WiFi 所记录的信息也能够从网络空间维度反映出人们的活动轨迹，同时 WiFi 的数据交也是能够有助于探究人们现实生活中的交互行为，因此通过用户的日常 WiFi 数据也能够了解到人与人之间的社交关系以及计算用户之间的关系强度。在基于用户 WiFi 感知数据度量关系强度中，本文提出了从用户 WiFi 感知上下文环境出发，通过计算 WiFi 感知环境的差异性计算用户之间的相似性进而得到用户之间的关系强度，采用了一种区别于传统意义的计算方式，通过对 WiFi 数据进行数据结构的构建和模型化表示，结合图形学的分析理论，将

WiFi 相似性分析与图分析相结合，通过计算用户之间的日常 WiFi 数据之间的相似性分析人们之间的关系。

### (3) 如何基于用户蓝牙感知数据度量关系强度

随着通信技术的发展以及与局域网通信之间的融合，出现了许多支持近场通讯的标准如 NFC、蓝牙等，而蓝牙模块在手机历经了几代更迭之后依然保留在手机中。用户手机之间的近场交互情况，对研究用户之间的交互以及用户的关系有着重要的作用。在现实生活中，用户设备之间的近场交互能够暗示出用户之间的交互，如蓝牙连接传输文件，NFC 交互交换名片等。通过将收集到的用户蓝牙感知数据序列化，并且按照时间片分割进行特征向量的构建和数据结构化表示计算蓝牙交互频率以及交互时长，对用户之间的关系强度进行计算。

### (4) 如何对三种独立数据源结果进行融合

前文分别概要描述了针对每种不同的非社交感知数据做如何以及采用的计算方法来得出用户之间的关系情况，在分别对每种感知数据进行计算的基础上，如何对每种计算方式内部的结果进行融合以及最后对三维数据的最终结果进行融合是一个非常重要的问题。基于 DTW 加权得到的空间距离表示用户轨迹的空间距离越小，用户之间的相似度越高关系强度越强；而基于运动模式的计算方式表明用户的轨迹运动模式越相似，说明用户之间的关系强度越强。参照集成学习的算法，对三维数据计算结果进行加权融合，作为最终的计算结果。

## 1.4 论文结构

本文的主要结构分为六章节，各章节的主要内容描述如下：

第一章为绪论，主要讲述了本课题的研究背景、然后介绍了智能手机感知数据同社会关系以及感知数据同人际关系强度研究所解决的问题，最后在提出了本文的研究目的，以及本文所研究的基于三种主要非社交关系数据研究的主要内容，最后总结归纳，基于这三种主要研究内容，提出本文的组织结构。

第二章主要描述了与本课题研究内容密切相关的前人研究成果，从最开始的轨迹数据处理中的滤波技术、停留点检测算法到最后的轨迹停留点聚类主要方法技术，并对相关技术进行了进一步分析；其次介绍了针对用户轨迹数据所采用的时间序列相似度计算所采用的常用距离算法以及各自的意义和优缺点；紧接着介绍了利用自然语言处理用户语义轨迹以及在自然语言处理用常用的计算文本相似性的相关技术，同时介绍了本文所采用快速计算文本相似性的方法，并且和 word2vec 进行了效率对比；最后针对剩下的非社交数据介绍了关于 WiFi 和蓝牙在挖掘用户社会关系中的应用。

第三章详细阐述多层混合模型 (RSMHD) 的整体框架，主要包含了基层的轨迹数据预处理，补全模块、轨迹语义化模块、以及三维轨迹数据计算模块和 WiFi 蓝牙数据处理计算模块。每个模块的详细描述将放在后面几章进行。

第四章主要讲述了如何对用户的空间轨迹数据进行预处理，建模等技术。首先描述了轨迹滤波模块；其次详细讲述了如何针对短时间窗口内的缺失轨迹数据进行智能补全；然后对用户轨迹进行语义化；针对用户的 WiFi 数据和蓝牙数据主要描述了如何借助图形学的思路对数据进行建模表示规整。

第五章主要详细描述了针对三种非社交感知数据进行用户的关系强度计算。首先描述了 RSMHD 的计算框架；其次分别针对各个数据模块的计算流程工作进行有针对性的说明。最后分别描述了基于各个感知数据源的计算结果和结果的融合。

第六章描述了本次研究所收集的用户数据集；最后展示了本研究的实验结果并进行分析。

第七章对本研究课题进行了总结，并提出了对未来工作的展望。



## 第二章 相关技术研究

上一章的内容中详细描述了本研究课题的研究背景，并且进一步分析了国内外的相关研究进展和现状，最后提出了本课题说研究的三个主要数据源和预计采用的研究方法。依据研究对象不同讲课题研究分为三个主要部分。本章将会详细的分析和阐述一些针对感知数据预处理的关键技术和方法以及论文中所涉及的快速语义轨迹计算、缺失轨迹补全等算法。

### 2.1 用户轨迹预处理

在用户的轨迹信息的采集过程中，因受到地形、气候、GPS 传感器和 SA 干扰误差的影响，会导致用户位置的跳跃移动等称之为“GPS 漂移现象”<sup>[36]</sup>，GPS 的位置漂移使得用户的轨迹数据中存在大量的噪音数据，影响后续对数据的处理和分析。因此对采集到的用户轨迹数据采取滤波处理，消除轨迹中所蕴含的噪音数据。GPS 位置的采集还受到地形环境的影响，在室内无法获取 GPS 位置信息的时候会导致用户轨迹的缺失，一旦用户轨迹出现缺失，对后续的度量工作也会产生影响。因此本小节首先描述 GPS 轨迹的噪音消除算法以及我们所采用的轨迹补全算法，最后描述各种常用聚类算法。

#### 2.1.1 滤波算法

滤波的主要目的是消除特定频率波段的噪音，用户的日常运动是连续的，所以采集到的用户 GPS 轨迹数据也应该是由连续的位置点构成的，但是由于 GPS 采集过程中受到漂移现象的影响，导致用户的 GPS 轨迹中存在位置点跳跃现象，因此要通过采用滤波算法对用户轨迹进行异常点剔除工作。接下来主要描述了一些经常使用的滤波算法：均值滤波、中值滤波、卡尔曼滤波这三种滤波算法。

a) 均值滤波：均值滤波也被称之为线性滤波，主要思想是采用结合中心点周围的数值，采取邻域平均法来表示这个邻域。数学公式表示如2.1所示，假设当前点为  $p$ ，则设置点  $p$  为采样中心。将  $p$  前  $m$  个采样点和后  $m$  个采样点的平均值作为当前点  $p$  的取值。

$$g(p) = \frac{1}{2m} * \sum_{j=i-m, j \neq i}^{i+m} g(j) \quad (2.1)$$

b) 中值滤波：中值滤波是一种基于排序统计理论的提出信号噪声点的非线性的信号处理方法，其基本原理就是点  $p$  的取值是由其邻域内各个点取值的中值来

决定的，让数据能够更加接近于真实的取值，从而有效地减少噪声数据点。中值滤波的具体数学公式见2.2，其中函数  $median()$  表示求中值。

$$g(p) = median(g(p - m), \dots, g(p - 1), g(p), g(p + 1), \dots, g(p + m)) \quad (2.2)$$

c) 卡尔曼滤波：卡尔曼滤波是卡尔曼于 1960 年提出的<sup>[37]</sup>，卡尔曼滤波器由一系列递归数学公式描述。它们提供了一种高效可计算的方法来估计过程的状态，并使估计均方误差最小。卡尔曼滤波器应用广泛且功能强大：它可以估计信号的过去和当前状态，甚至能估计将来的状态，即使并不知道模型的确切性质。接下来将介绍卡尔曼理论和实用方法。

在此之前需要引入离散随机过程，卡尔曼滤波器用于估计离散时间过程的状态变量  $x \in \Re^n$ ，这个离散随机过程的方程如2.3描述：

$$x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1} \quad (2.3)$$

定义我们所观测到的变量  $x \in \Re^m$ ，在此基础上得到卡尔曼滤波的测量方程见公式2.4：

$$z_k = Hx_k + v_k \quad (2.4)$$

其中随机的变量  $w_k$  和  $v_k$  分别表示计算过程中的激励噪声和观测到的噪声，我们假设他们二者之间相互独立，则可以得到正太分布的白色噪声如公式2.5和2.6描述：

$$p(w) \sim N(0, Q) \quad (2.5)$$

$$p(v) \sim N(0, R) \quad (2.6)$$

在实际随机过程中，激励噪声  $w_k$  的协方差矩阵  $Q$  和观测到的噪声  $v_k$  的协方差矩阵  $R$  是会随着每次迭代计算而改变的，因此为了便于推演我们假设它们都是固定的常数。当函数  $u_{k-1}$  等于 0 时或者噪声函数  $w_{k-1}$  等于 0 时，随机过程方程2.3中的  $n * n$  阶矩阵  $A$  将  $k - 1$  时刻的状态通过线性映射到  $k$  时刻的状态， $n * l$  阶矩阵  $B$  表示变量  $u \in \Re^l$  的增益，为了便于计算，这些变量在此都假设为常数。

我们定义  $\hat{x}_{\bar{k}} \in \Re^n$  为在第  $k$  项之前的状态下计算得到的第  $k$  项的先验状态估计值，设  $\hat{x}_k \in \Re^n$  表示已经得到的变量  $z_k$  时，第  $k$  步的后验状态估计值。由此根据以上描述我们可以定义出如2.7和2.8表示的先验估计误差后后验估计误差：

$$e_{\bar{k}} \equiv x_k - \hat{x}_{\bar{k}} \quad (2.7)$$

$$e_k \equiv x_k - \hat{x}_k \quad (2.8)$$

进一步得出先验估计和后验估计的协方差为：

$$P_{\bar{k}} = E[e_{\bar{k}} e_{\bar{k}}^T] \quad (2.9)$$

$$P_k = E[e_k e_k^T] \quad (2.10)$$

基于以上的理论准备，构建出卡尔曼滤波算法的数学表达式2.11，其含义为：先验估计  $\hat{x}_{\bar{k}}$  和测量得到的变量  $z_k$  同其预测值  $H\hat{x}_{\bar{k}}$  之差的线性组合共同组成了后验状态估计  $\hat{x}_k$

$$\hat{x}_k = \hat{x}_{\bar{k}} + K(z_k - H\hat{x}_{\bar{k}}) \quad (2.11)$$

根据2.11中表示可以进一步得到  $K$  的具体表达公式，其中真实变量与其预测之差  $(z_k - H\hat{x}_{\bar{k}})$  被称之为残差，该指标有效地反映了预测值与实际值之间的不一致的程度，如果残差为零，则表示二者完全相吻合。 $n * m$  阶矩阵  $K$  称之为剩余增益或者混系数，其主要作用是使得2.10中所得到的后验估计误差协方差最小，其计算步骤为：首先根据2.11代入2.8中求得  $e_k$ ，再将  $e_k$  代入2.10求得期望后对  $K$  进行求导，并令一阶导数为零就可以求得  $K$  的值如2.12：

$$\begin{aligned} \hat{x}_k &= \hat{x}_{\bar{k}} + K(z_k - H\hat{x}_{\bar{k}}) \\ &= \frac{P_{\bar{k}}H^T}{HP_{\bar{k}}H^T + R} \end{aligned} \quad (2.12)$$

从2.12中能够得知观测到的数据的噪音的协方差  $R$  越小，残余增益  $K$  越大，特别地当  $R$  趋向零时有：

$$\lim_{P_k \rightarrow 0} K_k = H^{-1} \quad (2.13)$$

另一方面，先验估计误差的协方差越小，残余增益  $K$  越小，特别当  $P_{\bar{k}}$  趋近于零时有：

$$\lim_{P_{\bar{k}} \rightarrow 0} K_k = 0 \quad (2.14)$$

综合2.12、2.13、2.14针对单个模型的测量利用上述说描述的所有公式就能够通过不断的迭代计算得出最优的估算结果。对于上述介绍的三种滤波技术而言，均值滤波在剔除数值中的随机噪音表现良好，但是不易消除脉冲误差；中值滤波能够减少偶然数据波动带来的误差影响，但是对变化快速的数据不宜使用；卡尔曼滤波最终的结果会优于前两者，但是模型较为复杂，所要求的运算时间复杂度也高于前面的滤波算法。

### 2.1.2 轨迹停留点检测

现实生活中用户的日常轨迹通常是一系列包含了地理坐标和时间戳的 GPS 位置点构成，每个坐标点包含了详细的经纬度、时间戳信息、海拔高度、移动速度等信息。如图2.1所示。我们可以通过一些算法检测出用户在一段轨迹运动中停留过的地方称之为停留点<sup>[38]</sup>，本文的停留点并不是指运动速度静止的点，而是由一系列 GPS 点构成的。如图2.1中  $p_4 \sim p_8$  构成了一个停留点 stay point 在图中由红色点表示。这个点表示用户在该区域内的停留时间超过了一个设定的时间阈值  $\Delta T$ ，相比于用户的其他轨迹位置点，这些计算得出的停留点蕴含了更重要的信息，通过语义标签甚至可以得出用户去过某家餐馆和电影院等。基于以上理论，用户的 GPS 位置轨迹可以转换为一组由停留点所组成的序列，如  $sp_1 \xrightarrow{\Delta t_1} sp_2 \xrightarrow{\Delta t_2} \dots \xrightarrow{\Delta t_{m-1}} sp_n$ ，这样由停留点所表示的序列不仅对原有数据维度进行了压缩，同时也保存了用户的重要信息。

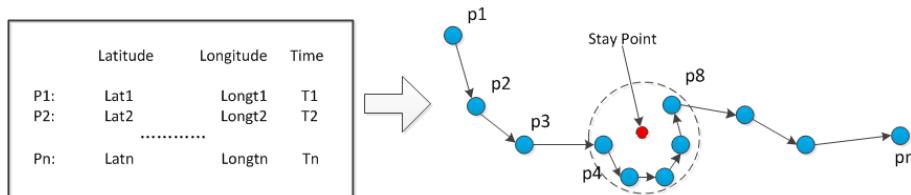


图 2.1 用户 GPS 轨迹示例图

停留点的计算过程如2.1所示。

### 2.1.3 聚类算法

由于现实生活中人们可能会经常多次访问同一个空间地点，但是所计算得到的停留点却可能并不完全相同（坐标的变差，计算的误差等因素影响），因此采用传统直接比较停留点的方法不具备可行性。我们采取对停留点进行聚类的处理方法，这样地理位置非常相近的点就会被划分为同一个类别中。接下来介绍一些常用的聚类算法。

聚类是属于无监督学习中的一种重要的方法，在其他的机器学习方法中如：回归分析、朴素贝叶斯网络等的数据都是带有类别标签  $\gamma$ ，也就是说在训练集中的样例就已经给出了样例的类别，而聚类数据样本却没有给出样本类别  $\gamma$  聚类的目标是根据组内元素距离最小，组间距离最大将原始数据划分为若干组如图2.2所示。本节主要介绍几种常用的聚类分析方法：K-Means 聚类算法、基于密度的 DJ-Cluster 聚类算法以及近年发表的一种改进的基于密度的聚类算法。

K-means 聚类算法是比较经典的聚类方法之一，由 J.MacQUEEN 在 1967 提出<sup>[39]</sup>。该算法执行效率高，在大规模的数据处理聚类时被广泛使用，该算法输入

**算法 2.1 停留点检测算法**

**已知:** 用户 GPS 轨迹  $Tra, \Delta T, \Delta_{distance}$

**求:** 用户停留点序列  $SP$

```

1:  $i = 0, PointNum = |Tra|, SP = Null$ 
2: while  $i < PointNum$  do
3:    $j = i + 1$ 
4:   while  $j < PointNum$  do
5:      $distance = Dis(Tra_i, Tra_j)$ 
6:     if  $distance > \Delta_{distance}$  then
7:        $\Delta_t = Tra_j.T - Tra_i.T$ 
8:       if  $\Delta_t > \tau_{time}$  then
9:          $p.Lat = avg(Tra_k.Lat)$ 
10:         $p.Lng = avg(Tra_k.Lng)$ 
11:         $p.T = Tra_i.T(arv|lev)$ 
12:         $SP.add(p); j++;$ 
13:       end if
14:     end if
15:   end while
16:    $i = j; break$ 
17: end while
18:
19: return  $SP$ 

```

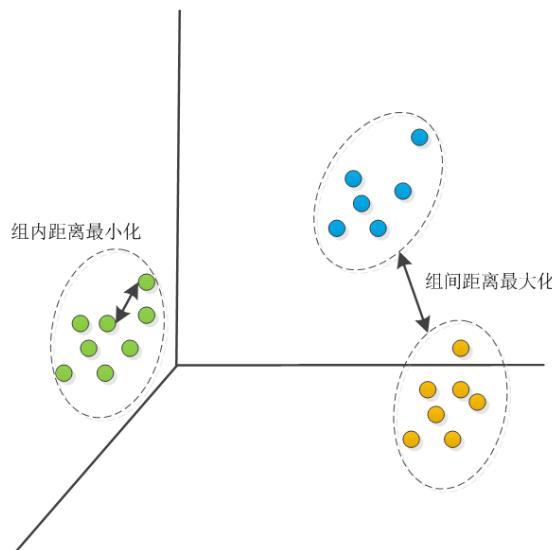


图 2.2 聚类效果示意图

$k$  作为最终聚类的个数，将待分类的  $n$  个数据分成  $k$  个簇，使得簇内数据具有高的相似度而簇间的数据存在较低的相似度。K-means 聚类算法的执行过程如下：

首先根据输入的参数  $k$  随机从原始数据中选择  $k$  个对象，每个初始化的对象代表了一个簇的中心；其次，剩余的每个对象计算与簇中心的距离将它们赋予距离最近的簇；第一轮结束后将重新计算每个簇的中心，这个过程不断重复知道准则函数收敛或者簇没有新的变化为止，通常采用平方误差的度量准则如2.15：

$$E = \sum_{i=1}^k \sum_{p \in C_i} |p - m_i|^2 \quad (2.15)$$

K-means 算法虽然简单，易于实现，但是在实际使用过程中需要用户指定  $k$  的取值，而  $k$  的取值是难以估计的，针对不同的数据事先并不可能确切的知道这些原始数据应该划分为多少个类别才正确；同时该聚类算法对异常数据非常敏感，一旦出现离群点将容易导致簇中心点出现漂移，对计算结果影响巨大。

相比于前面描述的基于距离的聚类方法，研究者于 2007 年提出了一种基于密度的聚类算法 DBSCAN<sup>[40]</sup> 以及日后基于该算法改进的 DJ-Cluster 聚类算法<sup>[41]</sup>，DBSCAN 算法的基本思想是扫描整个待分类的原始数据集，当扫描到数据对象 P 时，计算 P 的  $Eps$  邻域内所密度可达的数据对象个数是否大于等于定义的  $MinPts$ ，如果为真，则设立以 P 为核心的簇，然后尽可能的寻找与该簇密度相连的最大集合或者不断迭代查找该簇中每个数据对象的直接密度可达点，加入到该簇中。如图2.3中所示，设  $MinPts = 3$ ，从图中我们可以看出原始数据点 M, P, O 和点 R 的  $Eps$  邻域内说包含的点均大于等于  $MinPts$  因此都可以把它们标记为核心；M 是 P 的直接密度可达，Q 是 M 的直接密度可达。因此可以得知：Q 是 P 的密度可达，但是 P 不是点 Q 的密度可达，点 O, R 和点 S 是密度相连的。通过实验证明，DBSCAN 会受到  $Eps$  和  $MinPts$  取值的影响。

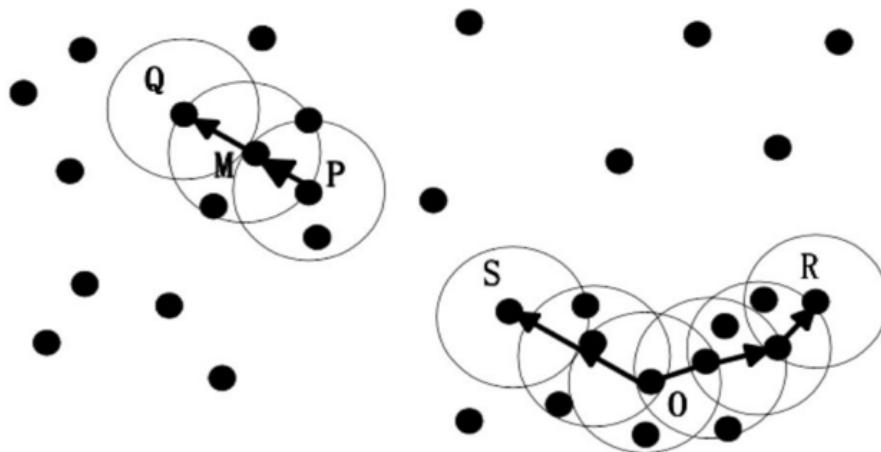


图 2.3 DBSCAN 密度直达和密度可达示意图

Rodriguez A 等人 2014 年在 Science 上发表了一种新的基于密度的聚类算法<sup>[42]</sup>。该算法相比较于之前的聚类算法，具有对参数不敏感便于输出正确的聚

类结果的有点。该算法的主要改进想法是针对所有待聚类的组标点基于它们之间的距离，提出了两个新的标准属性： $\rho$  和  $\delta$ ，基于聚类中心点具有：中心点自身的密度大，即它的密度超过邻域集合点的密度同时距离其它密度大的中心点之间的距离也足够大这样的特征，其中局部密度  $\rho$  采用 Cut-off kernel 计算方式，公式如2.16所示，其中  $p_i$  表示 S 中与数据点  $x_i$  距离小于  $d_c$  的点的数量； $d_c$  为截断距离需要用户事先设定并且保证  $d_c > 0$ 。

$$\rho_i = \sum_{j \in I_s \setminus i} \chi(d_{ij} - d_c) \quad (2.16)$$

公式2.16中的  $\chi(x)$  的计算方式为：

$$\chi^x = \begin{cases} 1 & x < 0 \\ 0 & x \geq 0 \end{cases} \quad (2.17)$$

算法中另一个指标距离  $\delta$  的定义为设  $\{q_i\}$  表示  $\{p_i\}$  的降序的下标序列，因此该序列满足：

$$\rho_{q_1} \geq \rho_{q_2} \cdots \geq \rho_{q_N} \quad (2.18)$$

则可根据公式2.19 计算出  $\delta$ :

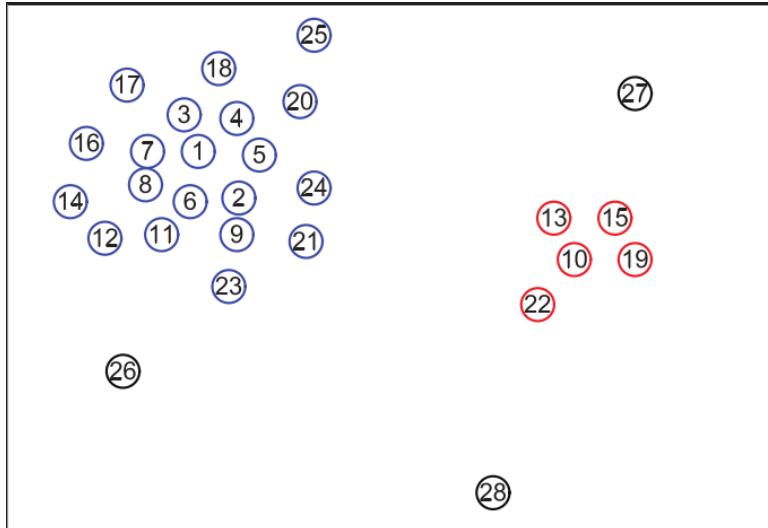
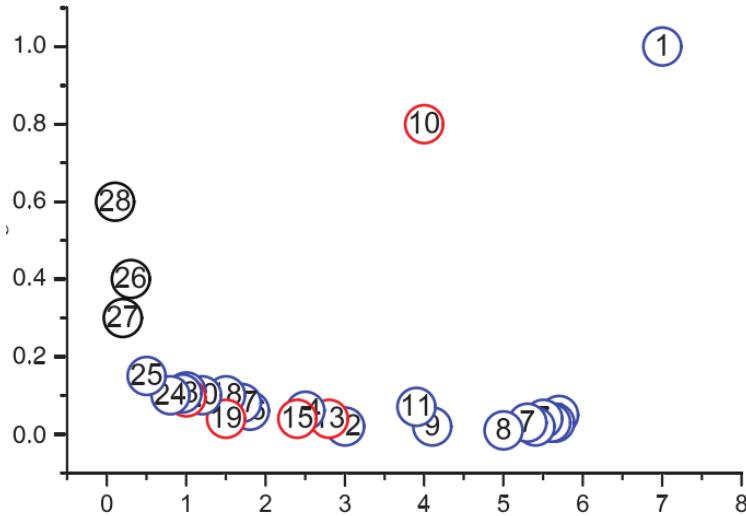
$$\chi^x = \begin{cases} \min_{q_j, j < i} \{d_{q_i q_j}\} & i \geq 2 \\ \max_{j \geq 2} \{\delta_{q_j}\} & i = 1 \end{cases} \quad (2.19)$$

该算法原理示意见图2.4和图2.5, 从图2.5可以看出，编号为 1 和 编号为 10 的坐标点具有较大的  $\rho$  和  $\delta$  取值，因此在原始数据中我们将这两个点设置为簇的中心，而在图2.4中这两个坐标点恰好是分类簇的中心点；同时还出现了编号 26-28 三个“离群点”它们的特点是  $\delta$  值很大而  $\rho$  取值很小。

总的来说，上述三种聚类算法通过精心调整参数都能取得非常好的聚类效果。不同的算法拥有不同的优缺点，在第四章我们将通过实验来展示各种算法在不同参数下的聚类结果。

## 2.2 时间序列相似度度量方法

我们阅读了大量社会心理学相关的论文学籍，从中证实得到在现实生活中关系亲密的两个用户会更加倾向于一起进行面对面的交流、共同进行社交活动等，因此通过对手机传感器数据的处理分析能够从中挖掘出人们现实生活中的关系强度。文献 [37] 立足于空间距离，提出了在空间距离上非常接近的人们在现实生活中就越可能发生面对面的交互，该文献通过调研一个小区的住户发现人们在现实

图 2.4 关于 decision graph 的示例<sup>[42]</sup>图 2.5 聚类中各点的  $\rho$  和  $\delta$  取值<sup>[42]</sup>

生活中越是接近，就越容易成为朋友。文献 [16][17] 进一步通过研究用户的轨迹数据发现由于在空间距离中接近的用户在现实生活中更可能产生交互行为，也就是说拥有相似日常生活轨迹的用户更可能产生交互活动。在现实生活中亦是如此，我们很容易和同自己经常出没同一个地点、行走在同一条轨迹上的人发展友谊关系。

用户的轨迹序列由带有时间戳的位置数据构成，位置数据可能是 GPS 也可能是基站号。因此我们可以将轨迹序列看作时间序列，从而使用一些时间序列相似度度量模型来度量轨迹的相似度，下面依次描述编辑距离和 DTW 这两种相似度度量方法以及序列熵值的计算方法。前面章节描述到用户的轨迹是由 GPS 位置点所构成的，包含了 (,,) 等详细信息，因此我们将用户的轨迹看作一条由带有时间

截所构成的坐标点序列，采用序列相似度计算方式来处理轨迹之间的相似度。接下来详细描述两种常用的序列相似度计算方法。

### 2.2.1 轨迹中心距离

传统的计算序列相似度所采用的方法如：编辑距离、汉明距离和夹角余弦等有其优缺点，如果贸然用来计算用户轨迹序列的相似性可能不太符合实际问题的需求，在此基础上，Hechen Liu 等学者提出了一种基于用户地理轨迹的相似度度量方法<sup>[43]</sup>，根据用户的轨迹形状以及时间片内轨迹的中心点之间的距离来作为度量用户轨迹之间相似度的标准。如图2.6中(a)图所示，现有两条用户轨迹  $Tra_1 = < a_1, a_2, a_3, a_4, a_5 >$  和  $Tra_2 = < b_1, b_2, b_3 >$  从观测来看两条轨迹是非常相似的，但是从整条轨迹对比来看，就很难评判两条轨迹是否相似了，因为  $Tra_1$  中有一处急转弯点  $a_3$ ，称为转折点。因此算法首先检测出轨迹中的转折点如图2.6中(b)再针对每段轨迹中中心点 (Center of mass) 来计算轨迹之间的相似性。算法中的 Center of mass 是轨迹  $Tra$  的质量中心，计算公式如2.20所示，轨迹计算结果示意图如图2.7所示，最后再计算轨迹之间的距离，采用余弦相似度来衡量轨迹之间的相似性。

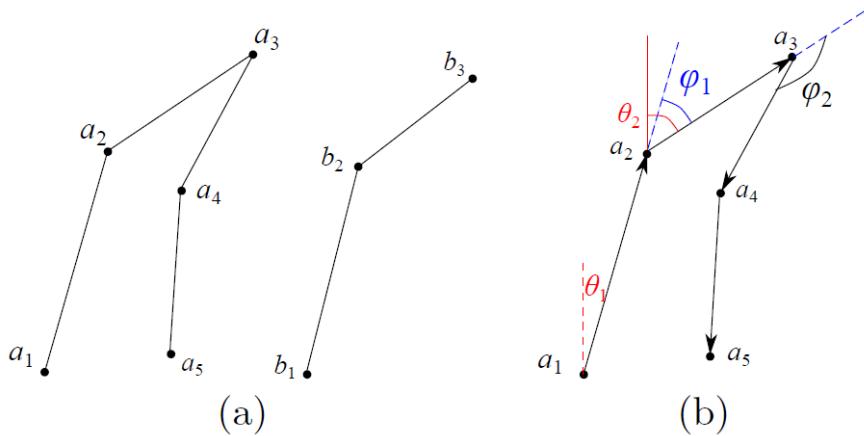
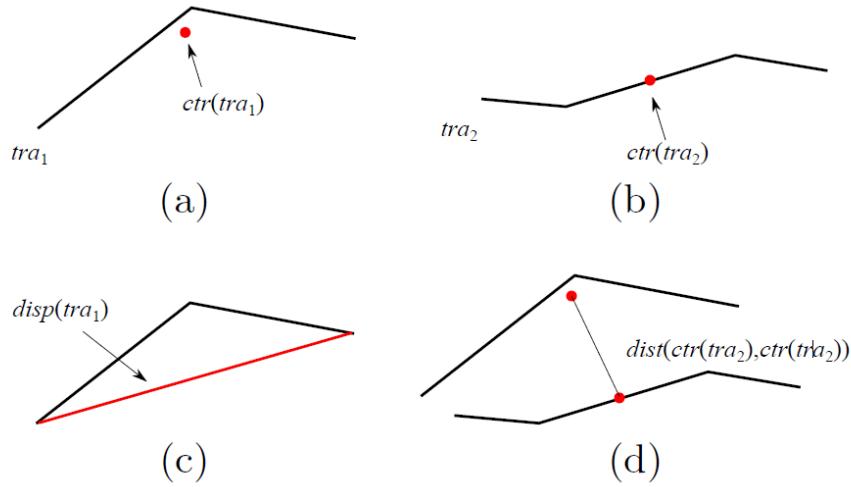


图 2.6 轨迹相似计算示例<sup>[43]</sup>

$$ctr(Tra) = (\bar{x}, \bar{y}) = \left( \frac{\int xf(x)dx}{\int f(x)dx}, \frac{\int yf(y)dy}{\int f(y)dy} \right) \quad (2.20)$$

采用该算法来计算用户轨迹之间的相似度同直接计算轨迹之间的欧氏距离相比较，能够将轨迹的形状考虑在内，这样能够结合避免直接计算轨迹之间的距离所因为离群点造成了计算结果的巨大偏差。

图 2.7 基于 Center of mass 轨迹计算示例<sup>[43]</sup>

## 2.2.2 Dynamic Time Warping

DTW(Dynamic Time Warping) 算法最初是由 Itakura 提出的的一种新型的计算距离的方法<sup>[34]</sup>，在最初的一段时间是被应用于语音识别领域，在语音识别中即使是同一个词语，由不同的人说出口但是他们的语速、语气等不同造成了单词音频的差别。DTW 正由于其计算距离的特殊处理使其能够胜任这一工作。DTW 算法采用动态时间规整的思想，将需要比较的两个序列在横轴时间维度上进行压缩、拉伸等操作，使得两条序列具有相同的长度具有更有效的匹配度，同时也消除了传统基于欧式距离计算带来的弊端。

设待计算距离的两个序列，序列  $Q$  和序列  $C$ （两条序列的表示见公式2.21、2.22）。如果两条序列的长度相同，则计算变得很简单。但是如果  $m \neq n$ ，就需要拉伸变形两条序列，使得它们的长度能够尽量对齐同时保留原有序列的信息，算法首先构造一个  $n * m$  矩阵  $d$ ，其中  $d[i, j]$  表示  $q_i$  和  $c_j$  之间的距离。采用动态规划的方法来找出序列  $Q$  和  $C$  之间的最佳匹配，其中转移方程为找出一条路径使得所有总和的距离  $\gamma[i, j]$  最小化，具体公式描述见公式2.23。路径及矩阵可视化见图2.8。其中：A) 为两个待计算比较的序列；B) 为通过 DTW 动态规划求解后两条序列计算距离时所对应的点；C) 将原有序列所对应的位置进行拉伸展开后更加直观的 DTW 匹配计算结果示意图。

$$Q = q_1, q_2, \dots, q_i, \dots, q_n \quad (2.21)$$

$$C = c_1, c_2, \dots, c_j, \dots, c_m \quad (2.22)$$

$$\gamma[i, j] = d[i, j] + \min\{\gamma[i - 1, j - 1], \gamma[i - 1, j], \gamma[i, j - 1]\} \quad (2.23)$$

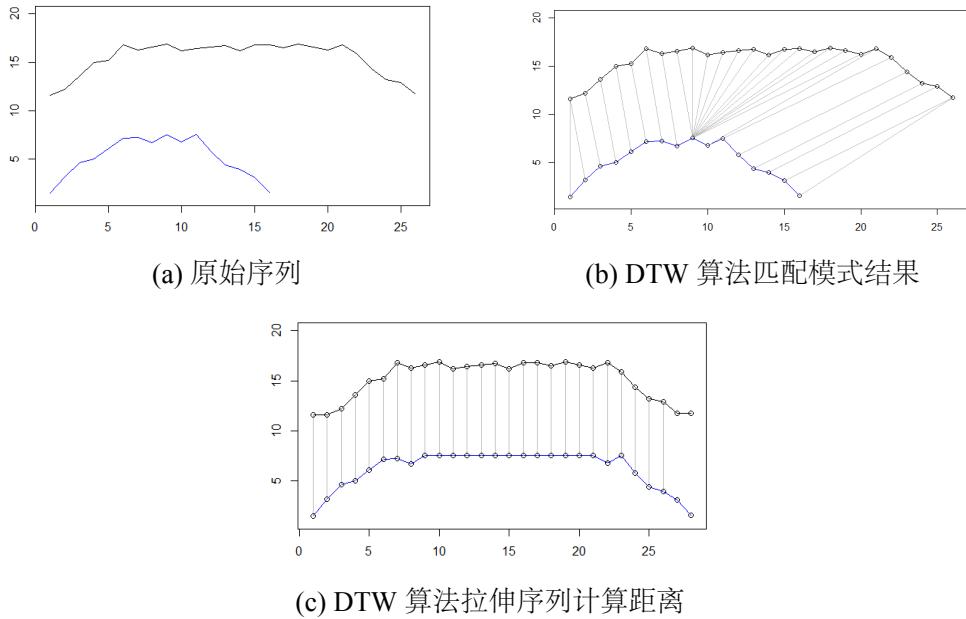


图 2.8 DTW 算法匹配序列结果示意图

通过对 DTW 算法的原理分析可以得知，序列  $Q$  和序列  $C$  的长度越长，则最终计算结果得到的距离就会越大。因此，后者研究中，采用了多种归一化的加权处理方法对结果进行加权处理 [44]，获得最优的 DTW 计算结果。

### 2.3 自然语言处理方法

相对于用户的空间轨迹度量，用户在日常生活中的语义轨迹蕴含了更丰富的上下文信息同时人与人之间在语义轨迹层次上特别是好友之间可能表现出惊人的一致性，如经常去某些地方，在一天中总是先从某个地点出发，再经过某些地点，最后在某个咖啡店相遇等等。基于轨迹的用户模式的交互能够反映出用户在某个空间中的相遇；而基于用户语义轨迹的分析能够在一定程度上展现出用户在社会活动上的相似性，在绪论部分已经详细从社会心理学的研究中描述了人们在现实生活中相遇频率能够反映出用户之间的关系强度，文献 [45] 研究发现人们会更加喜欢那些在兴趣、价值观、人格上和自己相似的人。因此通过用户的语义轨迹来在更高的层次上对用户的行为进行相似度计算，进而推测出用户的关系强度。

通过将用户的语义位置序集合同自然语言处理中的文档进行对比，可以借鉴自然语言处理的算法来处理用户的语义轨迹序列。用户每天的活动轨迹通过语义标签标注后，得到的是一个现实生活中语义轨迹的序列集合，如将一个用户的活动轨迹表示为  $\langle \dots \rangle$ ，通过结合自然语言文档相似性计算的思路，把用户每

天所计算得到的语义轨迹作为一条原始自然语句，用户在  $n$  天内所采集的所有轨迹记录就生成了一篇文档，最后基于自然语言处理方法通过计算用户语义轨迹生成的文档之间的相似性来表示用户之间的关系强度。

在以往的研究中，通常使用编辑距离来衡量语句的相似性。在用户轨迹相似分析上，文献 [46] 借助了 LDA 主题模型来计算用户语义轨迹之间的相似性，后来的研究在其基础上对计算方法进行了改进<sup>[47]</sup>。其算法改进在于基于 LDA 主题模型，将用户的所有语义轨迹看作一篇文档来训练出若干个主题。在计算用户轨迹相似性的时候，将用户输入到训练好的 LDA 模型中，然后采用余弦相似度分析比较二者输出的主题分布集合之间的相似度，作为这两个用户之间的关系强度。

word2vec 是 Google 在 2013 年开源的一种词向量计算算法<sup>[48]</sup>，word2vec 借鉴深度学习方法，将文本通过训练的得到文本的  $K$  维向量表示，通过词与词之间的距离来计算它们之间的相似度。

### 2.3.1 LDA 主题模型

2003 年文献 [49] 提出了 LDA(Latent Dirichlet Allocation) 模型对自然语言进行建模，可以用来识别文档语料中潜在的主题信息。整个计算模型采用了词袋的计算方法，计算出文档的向量表示，每一篇文档计算出一部分主题的概率分布，而每一个主题内又可以表示为很多词语的一个概率分布，LDA 的训练过程为：遍历每一个文档，在主题分布袋中随机抽取一个主题；然后在被随机抽取到的主题中再随机抽取一个单词；最后重复上述步骤直到遍历完文档中的所有词语。整个生成的过程可以用图2.9表示：对于每一份文档与设定的  $T$  个主题之间的概率分布对应  $\theta$ ，每个主题与词袋中的  $V$  个词语之间的概率分布  $\phi$ ，其中  $\theta$  和  $\phi$  分别有一个参数  $\alpha$  和  $\beta$  的狄利克雷的先验分布，这样对于任意一份文档  $d$  中的每一词语，我们从与文档相对应的概率分布  $\theta$  中选取一个主题  $z$ ，随即在根据与主题  $z$  对应的概率分布  $\phi$  中选取一个词语  $w$ ，最后重复上述过程  $N_d$  次，就能够产生文档  $d$ ，公式2.24为 LDA 的核心公式。

$$p(w|d) = p(w|t) * p(t|d) \quad (2.24)$$

### 2.3.2 word2vec

word2vec 是 Google 在 2013 年开源的一种词向量计算算法<sup>[48]</sup>，word2vec 借鉴深度学习方法，将文本通过训练的得到文本的  $K$  维向量表示，也就是说把特征转换到了  $K$  维空间进行表示，通过词与词之间的距离来计算它们之间的相似度。word2vec 采用了三层的神经网络结构即为输入层 - 隐含层 - 输出层构成，其

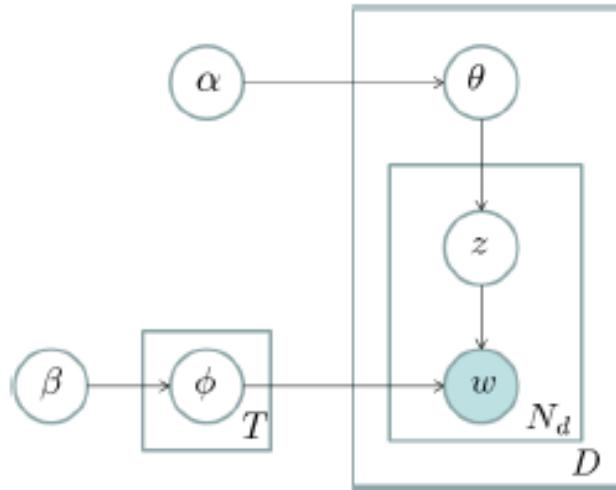
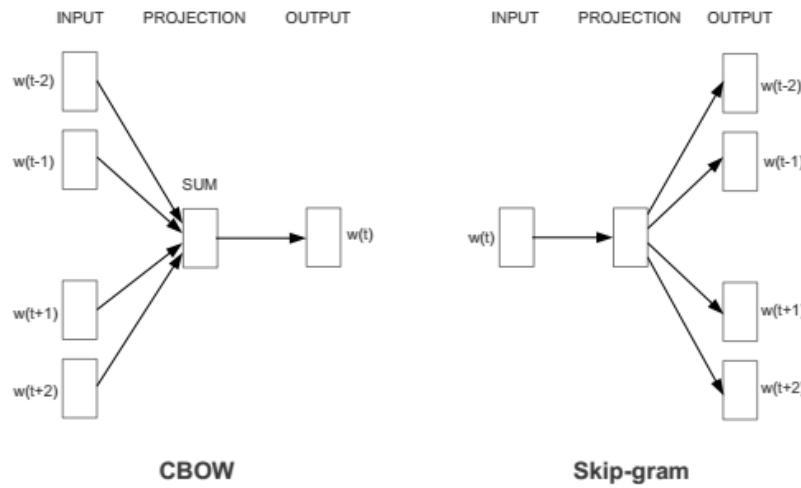


图 2.9 LDA 的图模型表示

主要方法是借助哈夫曼编码针对相似词频的词语构建出相似的隐藏层激活函数，算法采用了层次化的 Log-Bilinear 模型，其中一种是基于 CBOW(Continuous Bag-of-Words Model) 的计算模型，在 CBOW 模型中，根据上下文信息可以预测下一个词语  $w_t$ ，其公式如2.25所示，CBOW 的计算模型如图2.10所示。

$$p(w_t|context) = p(w_t|w_{t-k}, w_{t-k+1}, \dots, w_{t-1}, w_{t+1}, \dots, w_{t+k}) \quad (2.25)$$

现在的 CBOW 计算采用层次的 Softmax 算法，每个单词  $w_i$  都可以有一条从根节点出发被唯一访问到的路径，这条路径就形成了词语的编码。

图 2.10 word2vec 神经网络结构图<sup>[48]</sup>

在图2.11中，第一层也称之为输入层，它的输入是词向量；中间的层次为隐含层，得到的的输入是输入层若干个词向量的向量累加结果；第三层则是由二叉

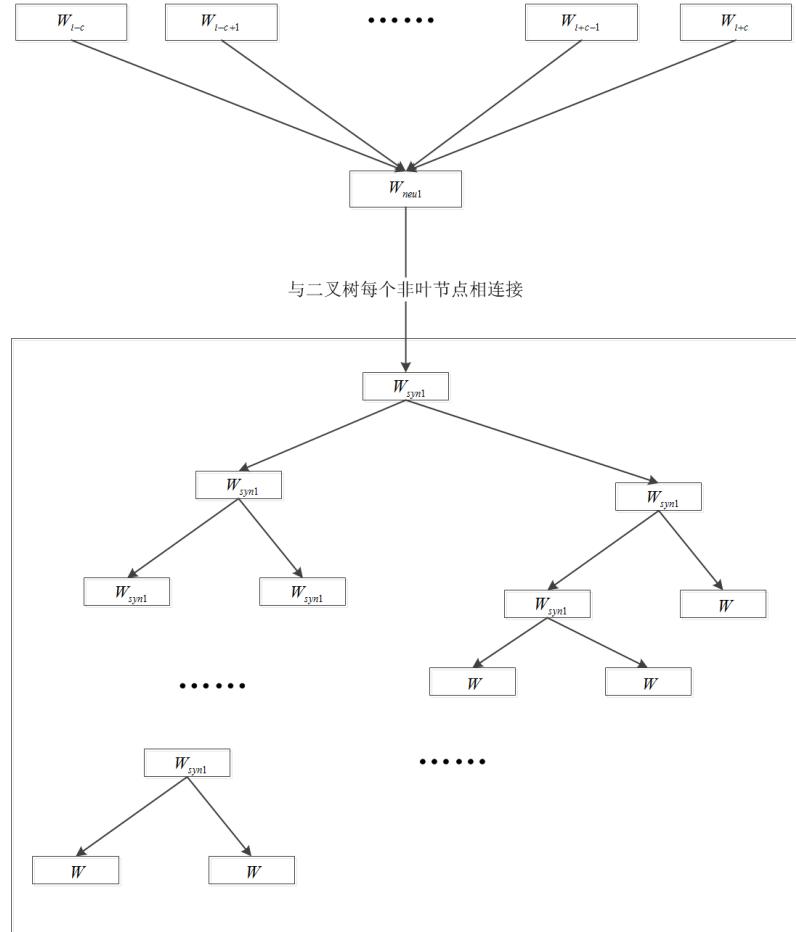


图 2.11 word2vec 神经网络结构图

树所构成的霍夫曼树所组成的输出层，其中每个非叶节点是一个计算后的向量但是不同于输入层的词向量，这里计算后的向量不代表某个词语，而是表示一个类别的词语，同时这棵霍夫曼树的所有叶子节点包含了输入词袋中的所有词。

对于词袋  $BG$  中的单词  $w$ ，在图2.11中一定能够找到有且仅有一条从 Root 节点到叶子节点  $w$  的路径。路径中的每一次分支都是一个概率问题，因此得到2.25中的数学表达式，将其中的  $w_i$  用词向量替换，得到该三层神经网络的目标公式如2.26， $Num^w$  表示从 Root 节点到词向量节点  $w$  路径中包含的节点个数， $w_j$  表示该路径上的第  $j$  个词， $v_w$  表示词  $w$  对应的词向量， $\sigma$  表示 SIGMOID 函数，通过求解目标函数的最大值，就可以得到每个词所对应的向量。

$$L = \sum_{w \in C} \sum_{j=2}^{Num^w} \{(1 - w_j) \log[\sigma(v_w^T \theta_{j-1}^w)] + w_j \log[1 - \sigma(v_w^T \theta_{j-1}^w)]\} \quad (2.26)$$

### 2.3.3 快速 Hash 算法

基于 word2vec 的计算方法针对文档的分词结果，再根据神经网络模型计算得到每个词的向量表示，但是这样无疑增加了整个计算过程的时间复杂度，本文研究目的是能够为众多用户提供一种计算关系强度的框架，因此如果采用传统的 Hash 算法进行处理，则能够减少程序的运行时间，来自于 GoogleMoses Charikar 针对海量文档快速计算相似计算提出了一种局部敏感 hash 算法<sup>[50]</sup>，其核心内容恰巧和 word2vec 相反：希望将数据进行降维处理，将原始的高维词向量通过一系列运算映射到低维的词向量，最后来计算相似性。

以上介绍了几种方法在自然语言处理中都得到了广泛的应用，因此综合考虑使用最后一种方法来作为计算方法，在后面还会详细讲解如何使用 hash 来度量用户之间的关系强度，以及在不同处理方法之间进行横向和纵向的比较。

## 2.4 WiFi 和蓝牙数据的度量方法

WiFi 和蓝牙作为一种主动对外界感知和探索所得到的感知信息，已经证实能够被用来分析用户之间的社交关系。文献 [27] 通过收集到用户的 WiFi 数据，然后将用户的 WiFi 数据同现实语义位置进行关联，得到了 WiFi 的语义向量，这样就能够得到基于 WiFi 的用户活动轨迹序列，提出了一种新的模型 AMVD(average minimum vector distance) 来计算人与人之间的距离，其计算公式如2.27所示，其中  $a_i$  和  $b_j$  表示用户 A 和用户 B 之间的联合向量， $d(a_i, b_j)$  表示计算二者向量之间的曼哈顿距离。

$$AMVD(A, B) = \frac{1}{|A|} \sum_{\forall a_i \in A} \arg \min_{\forall b_j \in B} d(a_i, b_j) \quad (2.27)$$

蓝牙数据和 GPS 轨迹数据都能够推测出用户之间的交互行为，蓝牙作为一种近距离的无线通讯手段，收到波长影响使得通讯距离只能是 5-10 米内，因此蓝牙感知数据能够进一步描述了用户之间在现实生活中的物理接触。前人的研究中，有根据蓝牙信息来简单计算出用户在现实空间中的面对面交互次数，从而推测出用户的社会关系情况<sup>[28, 29]</sup>。在本文中，我们使用了新的数据结构来表示用户的 WiFi、蓝牙感知数据，同时直接计算其感知上下文环境信息的相似度作为用户之间环境的相似度的参照。

## 2.5 小结

本章详细的从本课题研究的三个数据源出发对不同数据源的研究方法、现状进行了详细的分析和讨论。2.1 节讨论了在用户原始 GPS 轨迹数据处理中要解决的三个主要问题：剔除异常点、停留点检测以及空间轨迹的聚类。2.2 节则是介绍

了计算用户轨迹序列相似度的常用方法，并以此结果作为用户的关系强度计算结果。2.3 节从自然语言处理的角度介绍了常用的文档相似性检测计算方法，为计算用户语义轨迹的相似度提供了可行性算法。2.4 节从 WiFi 和蓝牙数据处理计算角度出发，介绍了前人研究的计算方法理论，为本研究中所采取的基于上下文环境的计算方式作出铺垫。

### 第三章 RSMHD 用户关系强度计算框架模型

上一章详细介绍了和本研究数据源相关的处理技术与方法，接下来这一章将主要介绍 RSMHD 的多维据源多维度的关系度量模型的整体框架结构。

#### 3.1 RSMHD 模型框架描述

本研究的主要内容是通过采集到的用户感知数据（GPS 轨迹信息、WiFi 数据、蓝牙数据）来开展用户之间的关系强度度量工作。因此本文希望能够提供一种基于此类非社交的感知数据建立的通用的用户关系强度计算框架。非社交数据主要是指不是来自于用户直接社交活动中所收集到的数据，在本研究中主要包含了用户轨迹数据、用户 WiFi 感知数据和用户蓝牙感知数据。因此本课题基于以上三种不同的感知数据源提出了能够统一处理多种数据源的计算框架，其整体概要结构图如图3.1所示，图中描述整体架构由以下几部分组成：上下文感知收集模块、基于用户日常轨迹的关系强度计算模块、基于用户 WiFi 感知数据的关系强度计算模块和基于用户蓝牙的关系强度计算模块。用户的日常语义轨迹是有连续的 GPS 位置点所组成的轨迹的集合，在用户的轨迹信息的采集过程中，因受到地形、气候、GPS 传感器的干扰误差的影响，会出现 GPS 漂移现象，GPS 的位置漂移使得用户的轨迹数据中存在大量的噪音数据，影响后续对数据的处理和分析。因此对采集到的用户轨迹数据采取滤波处理，消除轨迹中所蕴含的噪音数据。GPS 位置的采集还受到地形环境的影响，在室内无法获取 GPS 位置信息的时候会导致用户轨迹的缺失，造成计算结果产生巨大偏差。第四章部分将重点针对上述问题展开 GPS 轨迹的处理研究工作，以获得好的信息效果。对于剩余的 WiFi 和蓝牙感知数据来讲，主要的工作是针对二者数据源的特点以及后续的算法输入对原始数据进行提取和结构化建模表示存储。第四章的其余部分将会针对 WiFi 和蓝牙感知数据的预处理和结构化表示进行详细的描述。

##### 3.1.1 基于轨迹数据的关系强度计算模块概述

在本研究中所提出的 RSMHD 计算框架模型中，第二层需要基于用户的日常语义轨迹分别通过计算地理轨迹相似性、用户的语义轨迹相似性以及日常轨迹运动模式三个层次关系强度融合为轨迹强度结算结果，其详细框架图如图3.2所示。在这个模块中，通过计算用户日常地理轨迹的相似性推测出用户之间的关系强度；针对用户活动产生的日常语义轨迹，RSMHD 模型采利用自然语言处理的思想，通过快速编码变换计算出用户基于语义轨迹的关系强度；第三层中，RSMHD 进一步挖掘出用户轨迹中更高一层的上下文信息 (High Context) 得出用户日常轨

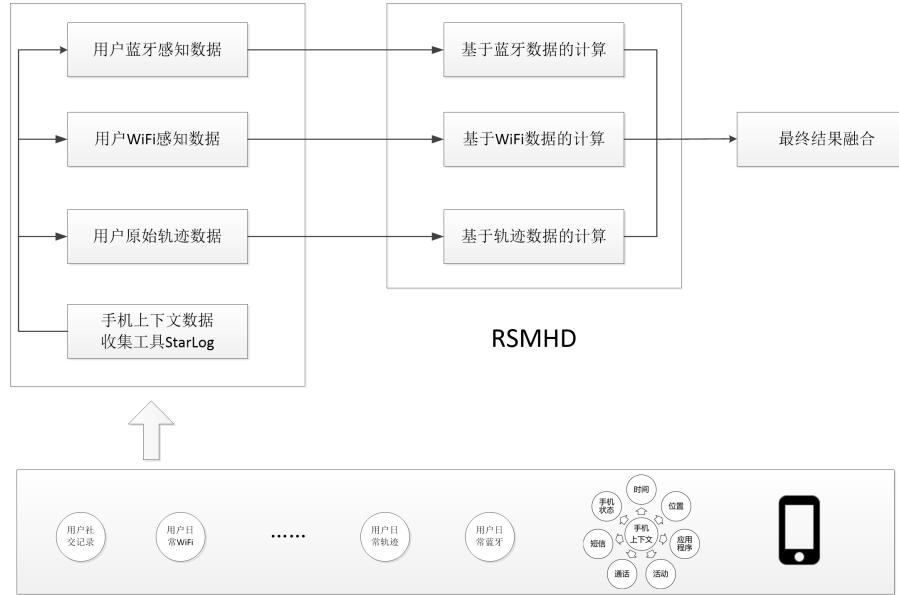


图 3.1 RSMHD 用户关系强度计算模型框架图

迹模式如：用户 A 经常喜欢到校外某地用餐、用户 B 经常下午在操场跑步等这些代表用户日常轨迹活动模式的信息，计算用户之间基于运动模式的关系强度。

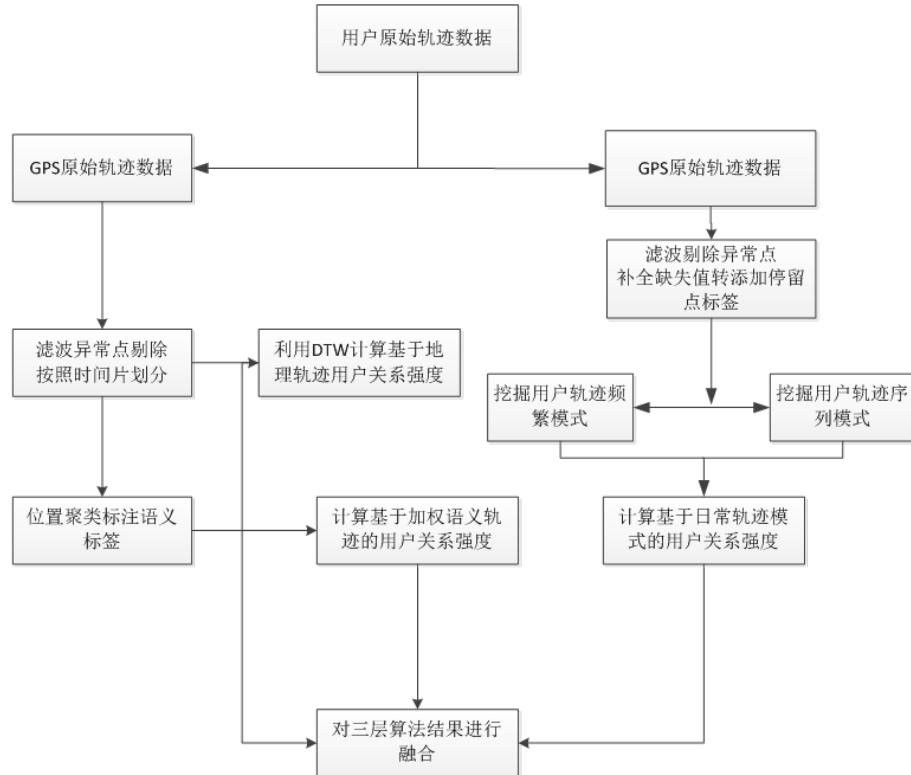


图 3.2 RSMHD 中基于用户轨迹计算模块详细框架图

### 3.1.2 基于 WiFi 和蓝牙数据的关系强度计算模块概述

在 RSMHD 计算框架模型中我们针对 WiFi 和蓝牙原始数据的特点以及现实生活中 WiFi 和蓝牙上下文环境的特征，采用了区别于原有基于 WiFi 和蓝牙计算社交关系的方法，整体的详细计算框架如图3.3所示。在 WiFi 和蓝牙数据处理计算框架中，分别针对底层手机的上下文感知信息进行数据信息的提取的规整，从复杂冗余的信息中萃取出关键的富有价值的信息；然后结合 WiFi 和蓝牙各自的上下文环境特征信息，将整理后的数据用图的数据结构进行结构化表示，使得抽象表示后的数据结构更加符合现实中的含义；最后基于结构化后的感知数据进行关系强度计算，将结果输出到下一层。

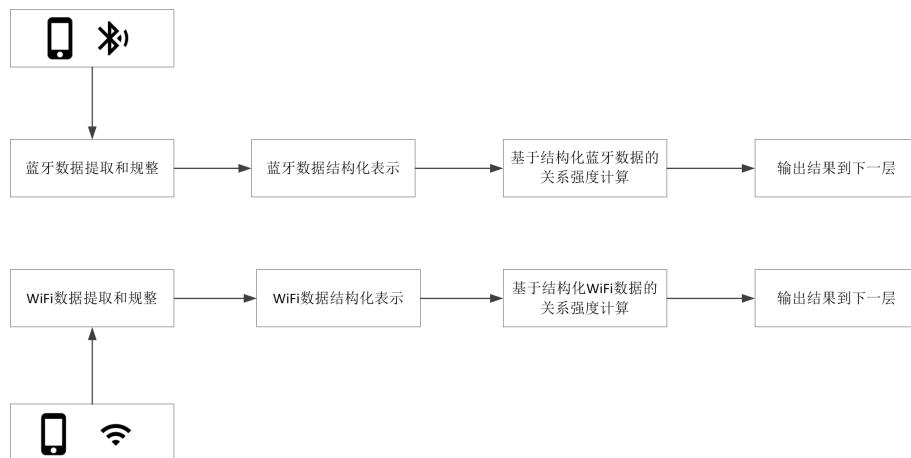


图 3.3 RSMHD 中基于 WiFi 和蓝牙计算模块详细框架图

## 3.2 RSMHD 模型计算流程概述

整个信息采集是采用我们自己编写的用户手机上下文信息收集工具 StarLog 来记录保存用户手机的上下文感知数据如：GPS 位置信息、WiFi 感知信息、蓝牙感知信息、脱敏的通话记录、短信记录、APP 使用记录等。

针对用户的 GPS 位置数据我们首先对轨迹数据进行预处理，即采用滤波算法检测出用户日常轨迹中的漂移点减少轨迹中的噪音；然后根据停留点检测算法识别出用户轨迹中的停留点，针对缺失的轨迹点进行预测补全；第一层次中针对用户日常空间轨迹相似性计算时，可用将每天的轨迹数据划分为若干时间片的数据，然后采用 DTW 加权算法对时间片内的空间轨迹相似度进行计算并按照一天的轨迹进行处理。对于基于用户语义轨迹推倒关系强度这层我们首先将得到的停留点进行聚类分析，得到聚类后的停留点集合因此对于每一个簇都是同一个语义位置，然后将聚类 GPS 转换为现实社会中的语义标签，在计算用户语义轨迹的相似性时，采用快速 hash 算法将用户分片后的语义轨迹序列作为输入，得到用户之

间每天的语义轨迹相似度，以及最终的语义轨迹相似度并将结果作为用户自己的关系强度。在上一层的基础上，我们针对得到的用户语义估计进行挖掘，寻找出用户的日常运动模式，然后计算用户运动模式之间的相似度并作为关系强度之一进行输出，综合三层的计算结果得到用户基于轨迹数据计算得到的关系强度。

其次针对用户的 WiFi 和蓝牙数据，首先从原始数据中提取出重要的信息，并将 WiFi 和蓝牙数据进行结构化存储处理，采用图的存储结构，将 WiFi 和蓝牙数据还原为现实生活中的存在环境，然后针对用户切片时间内每刻的感知环境进行相似度计算，并将一天中的计算结果进行汇总进一步得到所有时间段内的用户相似度，最后将结果作为基于两种数据源推测出的用户关系强度。

最后，在基于前面三种数据源的用户关系强度计算结果的基础上，采用集成学习的思想对计算结果进行融合处理，得到最终的用户关系强度计算结果。

## 第四章 RSMHD 框架的相关技术研究

上一章节主要是对本课题研究的 RSMHD 框架进行了详细的描述，本章接下来将针对每一个模块中的具体面对的问题以及采取的方法进行详细描述，包括噪音点的剔除、如何语义轨迹、如何计算轨迹相似性以及如何对 WiFi 和蓝牙数据进行结构化处理等。

### 4.1 GPS 轨迹处理计算技术

用户的日常 GPS 轨迹数据包含了用户的生活工作以及娱乐等丰富的上下文信息，如图4.28所示，我们基于用户的日常轨迹分别从地理位置相似性、语义轨迹相似性以及轨迹模式相似性出发度量人与人之间的关系强度。模块的输入是由 StarLog 收集的 GPS 感知数据，通过对初始数据的解析提取得到原始的 GPS 位置点构成的用户轨迹序列。预处理阶段主要是通过滤波算法对原始轨迹序列进行异常点检测，并通过轨迹预测补全缺失的部分 GPS 轨迹序列；第二层中将第一层得到的停留点赋予语义得到用户的语义轨迹序列，采用自然语义处理思想计算基于语义轨迹的相似性；再往上一层，从用户的历史轨迹数据中挖掘出的频繁模式和序列模式能够反映出用户的日常轨迹运动习惯和行为规律，运动模式在现实生活中表现为用户经常行走的路径序列，是用户轨迹数据规律的抽象表示，最后计算用户之间的关系强度。

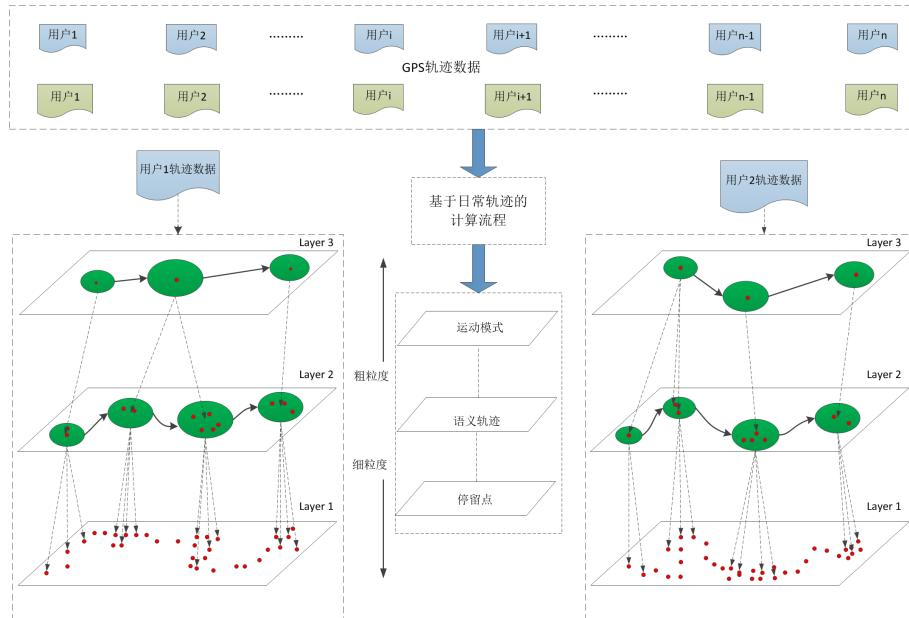


图 4.1 基于多层次多粒度轨迹的用户关系强度计算

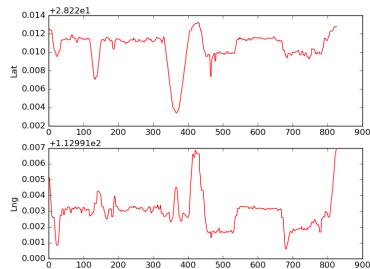
## 4.2 GPS 轨迹预处理与语义化

本节主要讲述如何对 GPS 轨迹数据进行清洗规整，以及得到用户的语义轨迹进行结构化存储。

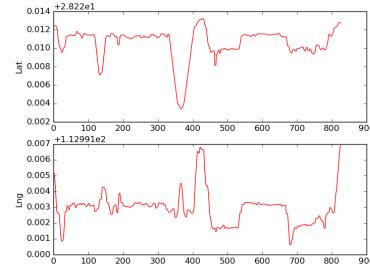
### 4.2.1 剔除轨迹中的异常点

前文已经提到，由于在获取 GPS 位置信息的时候会受到位置漂移的影响，导致在获取用户实际位置的时候可能产生采样的误差甚至跳跃，为了使得最终计算得到的关系强度结果更为准确我们需要对 GPS 轨迹数据进行滤波分析。在第二章中描述了常用的三种滤波方法，在本章中将会针对三种滤波算法进行最后结果展示并根据结果分析最终采用此滤波算法的原因。

首先接下来使用我们自己开发的用户感知数据收集软件 StarLog，来分析观察各种滤波算法对用户轨迹中异常点检测剔除的效果，如图如图4.2、4.3所示。

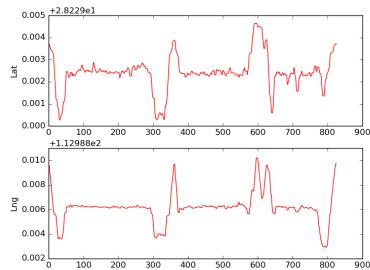


(a) 原始轨迹数据

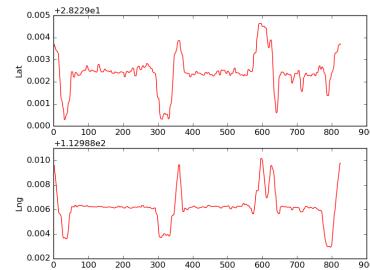


(b) 中值滤波后的轨迹数据

图 4.2 轨迹中值滤波实验结果 1



(a) 原始轨迹数据

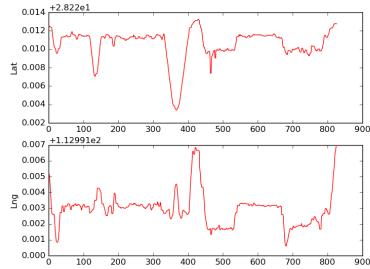


(b) 中值滤波后的轨迹数据

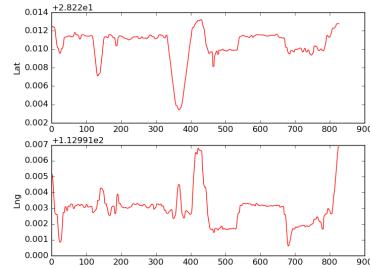
图 4.3 轨迹中值滤波实验结果 2

从实验结果中我们可以观察到虽然中值滤波能够过滤掉其中少部分的位置漂移点，但是针对于一些明显的轨漂移点却没有能够很有效的识别过滤。

接下来我们再使用均值滤波算法对用户轨迹进行分析，观察均值滤波算法对用户轨迹中异常点的检测情况，部分实验结果见图图4.4、4.5。

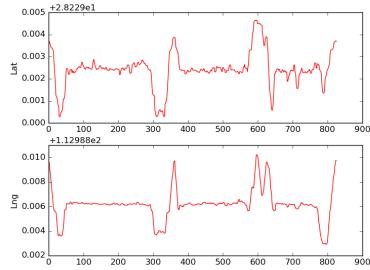


(a) 原始轨迹数据

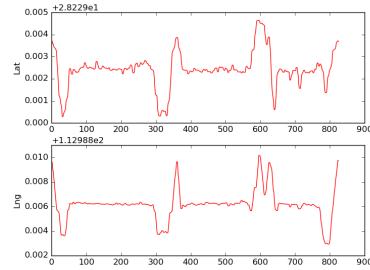


(b) 均值滤波后的轨迹数据

图 4.4 轨迹均值滤波实验结果 1



(a) 原始轨迹数据



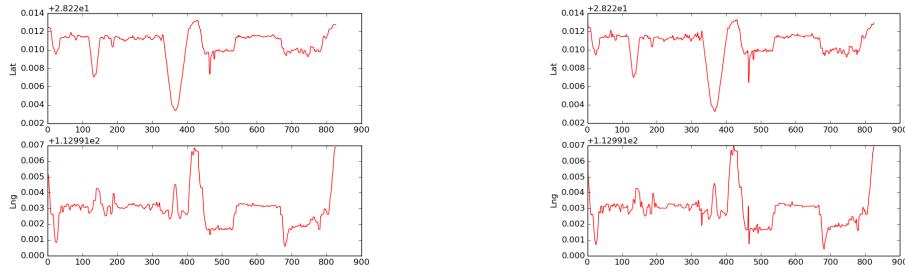
(b) 均值滤波后的轨迹数据

图 4.5 轨迹均值滤波实验结果 2

通过对比观察滤波结果图可以发现，相对于中值滤波，均值滤波能够较好的平滑用户的轨迹变化曲线，剔除 GPS 漂移点。但是，同样无法很有效的处理漂移偏差大的位置点。

通过采取卡尔曼滤波算法得到的用户轨迹如图4.6、4.7，根据观察图中滤波后的用户轨迹数据，我们可以发现虽然图形变得平滑了许多，但是却使得原有的轨迹信息收到了模糊，难以有效的将两个用户轨迹进行相似度计算。

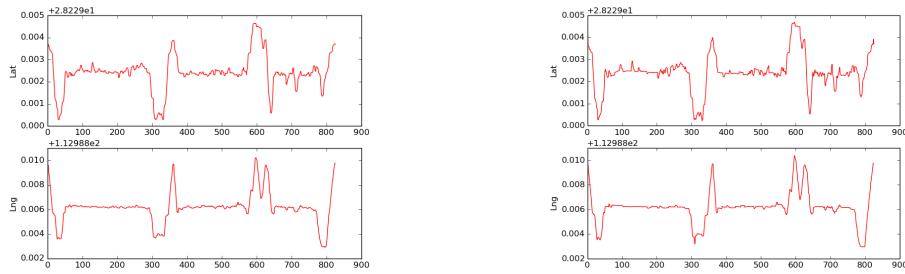
在本研究中，我们采用了一种基于速度的分段卡尔曼滤波方法。考虑到在一个时间片内，如果当前位置点的速度和它之前的位置点的速度绝对差大于  $\Delta v$  时 ( $\Delta v$  作为一个未知的参数，需要我们在实际使用中给出) 采用这样的方法将原有用户轨迹切分为  $n$  段然后针对每一段轨迹采用卡尔曼滤波算法，最终的部分轨迹滤波结果见图4.8、4.9，可见经过按照速度分段后使用卡尔曼滤波能够比较好的过滤掉漂移点。



(a) 原始轨迹数据

(b) 卡尔曼滤波后的轨迹数据

图 4.6 卡尔曼滤波实验结果 1



(a) 原始轨迹数据

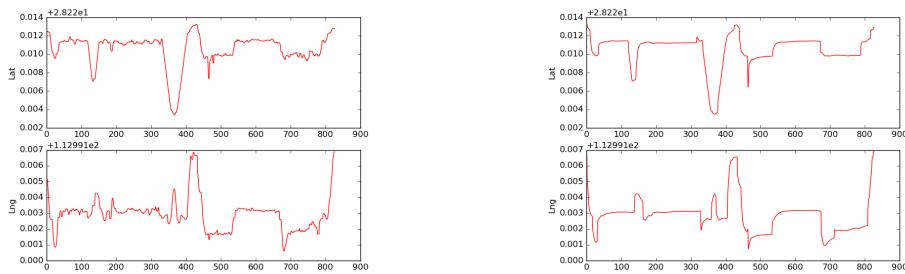
(b) 卡尔曼滤波后的轨迹数据

图 4.7 卡尔曼滤波实验结果 2

本小节主要针对前文中提及的三种常用的滤波算法进行了实验结果展示和效果分析，初步得出了使用卡尔曼滤波更加可行的判断结果。

#### 4.2.2 轨迹中停留点检测

在实际生活中用户的轨迹是由一系列 GPS 点构成的，剔除其中的噪音点和用户在路上的点，能够从中挖掘出进一步信息的位置信息即为轨迹中的停留点，通常停留点并不是指用户轨迹中速度为零的点，而是由一组 GPS 点构成区域，一个



(a) 原始轨迹数据

(b) 分段卡尔曼滤波数据

图 4.8 分段卡尔曼滤波轨迹结果 1

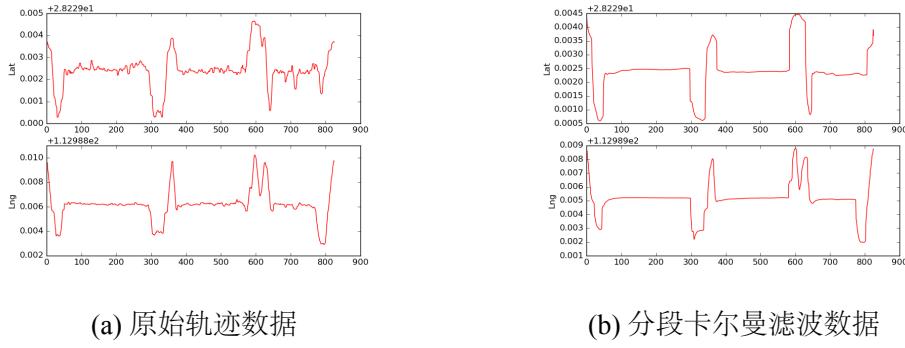


图 4.9 分段卡尔曼滤波轨迹结果 2

停留点通常对应现实生活中富有具体意义的点，能够更好的反映出用户之间轨迹的相似程度。如图4.10所示，在现实生活中的停留点可以是一个具体的建筑名称，也可以是代表一个固定的区域，甚至可以是一个具有特殊意义的点。

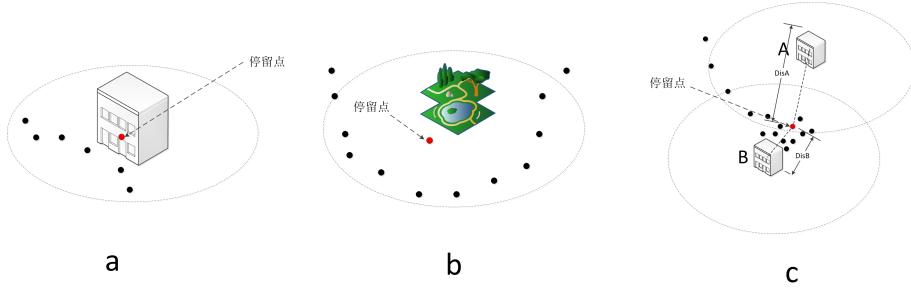


图 4.10 现实生活中的停留点示意图

通过对停留点的识别，能够让我们更加深入的了解和认识用户的日常轨迹，同时从一个更加细粒度的层次来分析用户之间的轨迹相似度，停留点检测实验结果如图4.11、4.12所示，图中所得到的每一个停留点都具有丰富的现实意义，能够有效地表示用户访问的地点、位置。

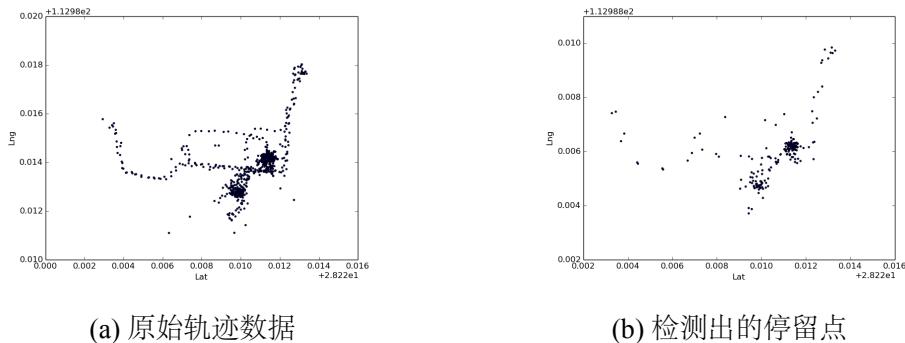


图 4.11 停留点检测实验结果 1

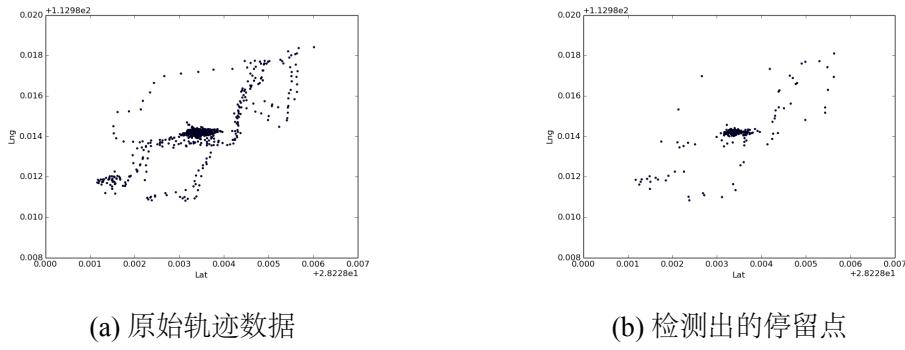


图 4.12 停留点检测实验结果 2

### 4.2.3 用户轨迹停留点聚类

本节针对上面得到的用户轨迹中的停留点，采用不同的聚类算法所得到的不同的聚类结果进行描述。在前面的章节中我们详细介绍了聚类中常用的三种算法 K-means 聚类、DJ-Cluster 的密度聚类和最新发表的改进的密度聚类算法，接下来我们将在实验中对这三种聚类算法进行比较。

首先我们对用户轨迹数据采用 K-means 算法进行聚类实验，但是 K-means 聚类算法具有以下的明显的缺点：首先是 K-means 聚类结果依赖于参数  $k$  的初始化，其中  $k$  是指聚类个数。 $k$  个数的确定往往是靠经验值设定；其次是 K-means 算法初始化时聚类中心的选择，因为该算法采用的随机初始化聚类中心点，中心点选择不同会导致聚类的结果也出现差异甚至影响聚类的时间复杂度，使得聚类结果容易得到局部最优解而非全局最优解；最后的一个缺点是 K-means 聚类算法对原始数据中的噪音点和离群点非常敏感，聚类结果很容易受到离群点的影响从而导致簇的偏移。

在本研究中因为是针对用户的空间轨迹进行聚类分析，而现实生活中用户的停留点主要是以相隔距离比较大的建筑等，因此在采用 K-means 聚类过程中，对噪音点的影响基本可以忽略不计，最主要的就是考虑不同的参数  $k$  对最终聚类结果的影响，图 4.13、4.15 主要展示不同的参数  $k$  取值对最终轨迹聚类结果的影响。图 4.14、4.16 在地图上标记了在不同参数  $k$  取值情况下的簇中心使得能够有更加直观的观察。

通过对轨迹数据的聚类分析我们可以得知，如果在聚类前我们能够确切的知道用户的轨迹数据应该划分的簇的个数，那么 K-means 聚类算法将会得出非常合理的结果，我们在观察图 4.14、4.16 时发现，当簇的个数越接近符合用户访问位置个数的时候，那么聚类点将更能够反映出真实的情况，在图 4.18 中，(a) 所对应的聚类得到的簇包含了实际上没有正确的将用户访问过的位置区分开，因为这两个访问的位置距离相隔太近，导致 K-means 算法将这两个簇合并为一个簇，但是

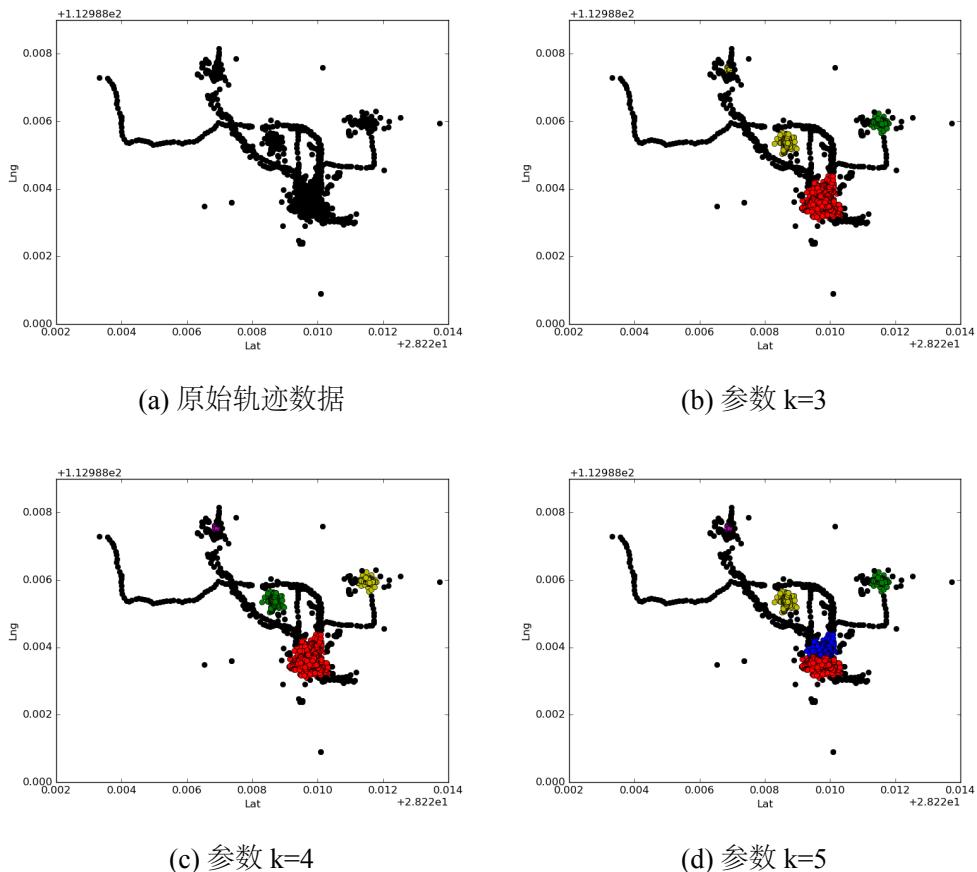


图 4.13 基于 K-means 的轨迹聚类实验结果 1



图 4.14 基于 K-means 的轨迹聚类实验结果地图展示 1

如果我们将聚类个数设置为 5 就能够成功的识别出这两个位置。在另一实验结果图4.20中也出现了类似的情况。但是根据图中实验情况分析发现对于一部分在路上的点被划分到了聚类簇中，同时有些应该被单独聚类成一个独立的簇的点被划

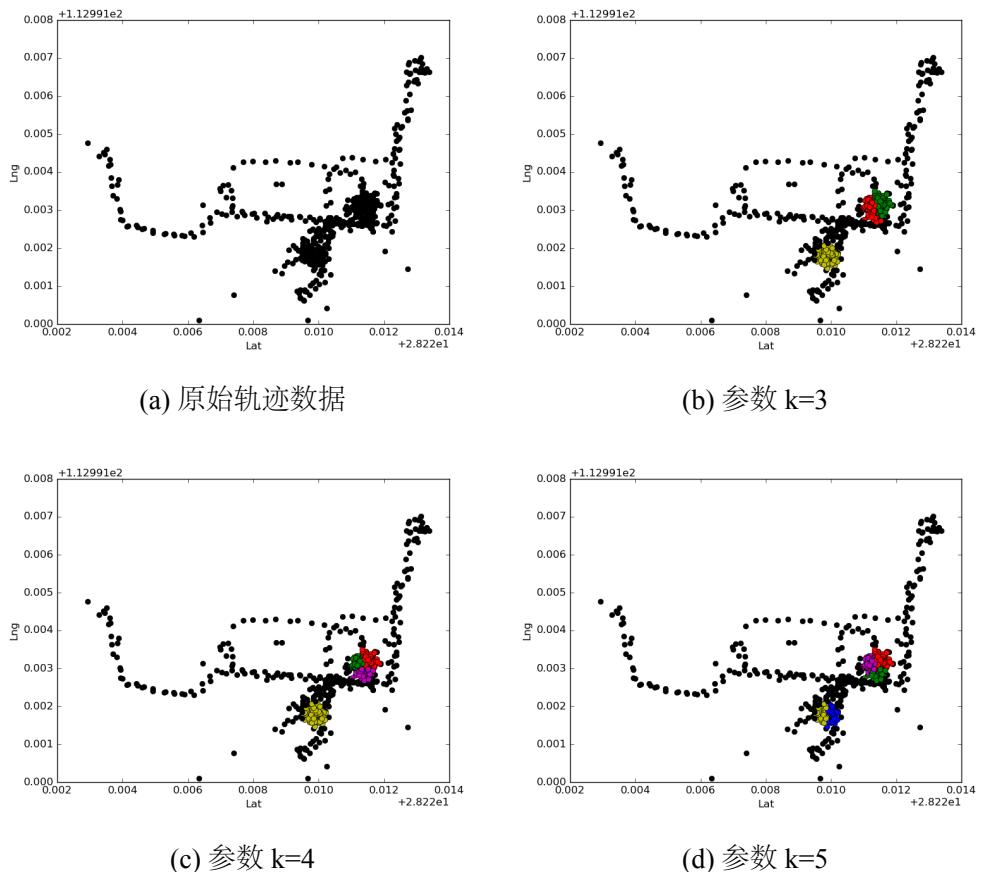


图 4.15 基于 K-means 的轨迹聚类实验结果 2



图 4.16 基于 K-means 的轨迹聚类实验结果地图展示 2

分到了在路上的点中，这些都是因为我们无法正确的给出符合当前轨迹访问点个数的聚类个数所造成的。

接下来将会基于 DJ-Cluster 算法来对轨迹数据进行聚类并对结果进行分析。DJ-Cluster 算法是基于密度聚类算法 DBSCAN 而改进的一种密度聚类算法，通过利用密度相连的原理可以发现任意形状的簇，但是算法存在有一个局限性：算法的性能和准确性依赖于参数  $Eps$ （邻域半径）和  $MinPts$ （最少点数量），如果邻域半径的取值过大那么所有的点将会划分为同一个簇，因此在实际的使用过程中，需要不断地对参数进行调整才能得到比较理想的结果，如图4.17、4.19结果所示，实际地图中结果展示如图4.18、4.20所示。

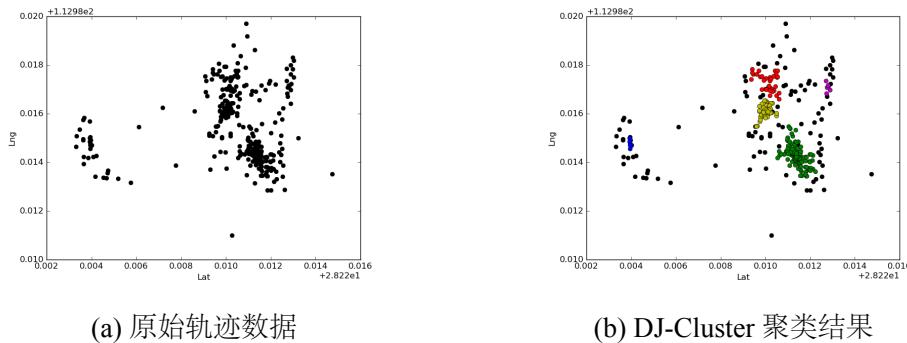


图 4.17 基于 DJ-Cluster 的轨迹聚类实验结果 1



图 4.18 基于 DJ-Cluster 的轨迹聚类实验结果地图展示 1

通过观察结果地图展示和原始数据对比能够发现，相比于 K-means 算法 DJ-Cluster 能够得到比较正确的结果，但是结果依赖于对前面所描述的参数的不断调整。从图4.18中 (a) 中描绘的用户位置点集合 (b) 中得到的聚类结果进行分析，右上角是宿舍区域，因为当前点比较少，未能够正确的将这个区域聚类成一个单

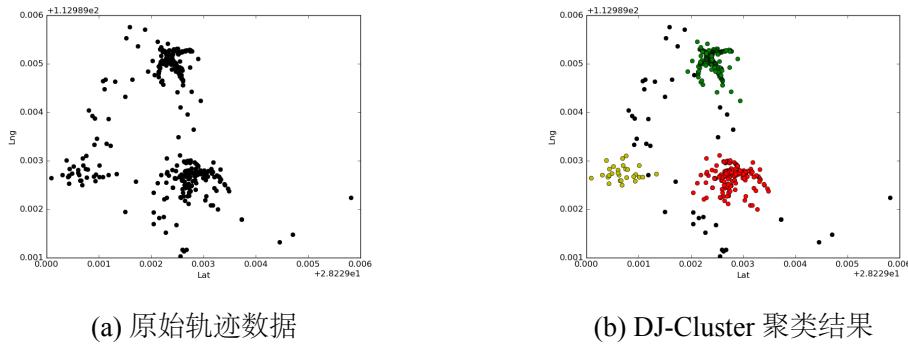


图 4.19 基于 DJ-Cluster 的轨迹聚类实验结果 2

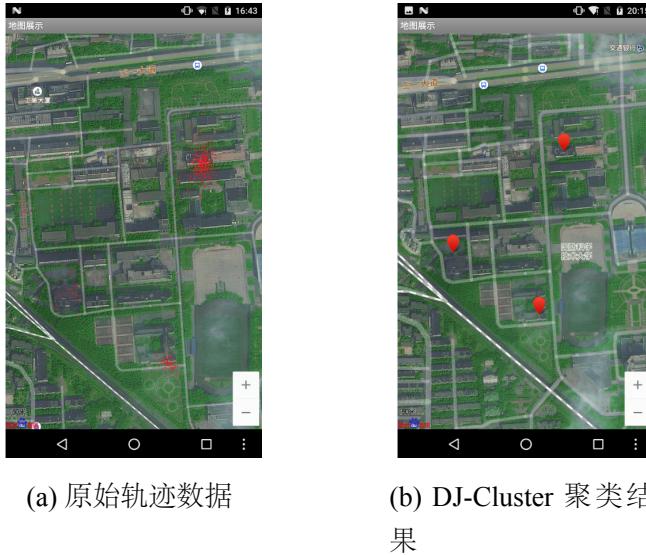
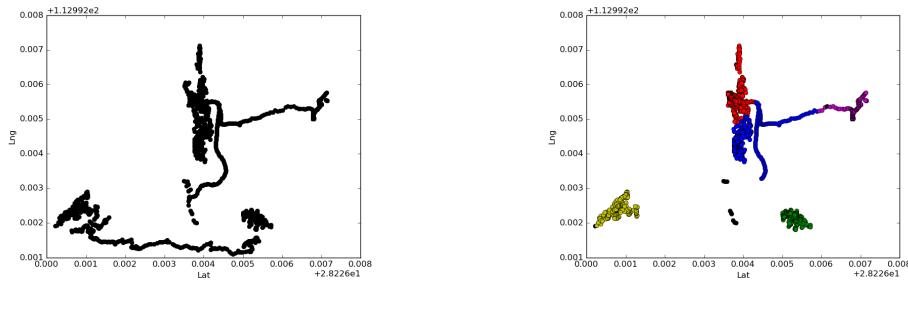


图 4.20 基于 DJ-Cluster 的轨迹聚类实验结果地图展示 2

独的簇进行标记。但是从图4.20中发现 DJ-Cluster 基本得到了非常好的聚类结果。受限于 DJ-Cluster 算法对参数的敏感性，接下来将介绍一种新的聚类算法，该算法将大大降低对参数和数据的依赖性。

接下来进一步展示在 Science 上发表的聚类算法，该算法通过结合现实理论，假设聚类的中心实质上是有局部密度低的点所环绕的，并且这点和另外密度高的点距离都比较大。这样通过计算得到每个点的当前密度值后根据划分的边界阈值进行密度点的划分，找出每个块中密度最大的点并将其周围密度小于该点密度的点加入到当前簇中得到最终的聚类结果。该算法因为考虑当将点密度排序后进行处理，保持了点密度顺序的相对稳定的同时减小了半径阈值带来的影响。聚类算法的具体的实验结果见图4.21、4.23，将聚类结果通过地图展示如图4.22、4.24所示，从图4.21中可以观察到聚类结果是比较



(a) 原始轨迹数据

(b) 聚类结果

图 4.21 轨迹聚类实验结果展示 1

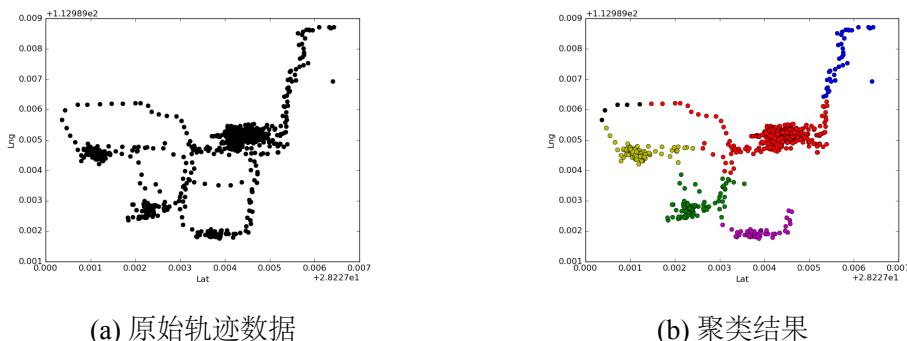


(a) 原始轨迹数据

(b) 聚类结果

图 4.22 轨迹聚类实验结果地图表示 1

通过分析图4.21和图4.21中的聚类结果发现该聚类算法能够比较正确的将用户的轨迹数据进行聚类同时也不依赖于太敏感的参数输入。在图4.21中用户正确访问的地方应该是五个，虽然(b)中的聚类结果也得出了五个簇中心位置，但是未能将寝室这一簇识别出来而是将在体育馆周围的点标注为聚类簇。虽然在最终



(a) 原始轨迹数据

(b) 聚类结果

图 4.23 轨迹聚类实验结果展示 2

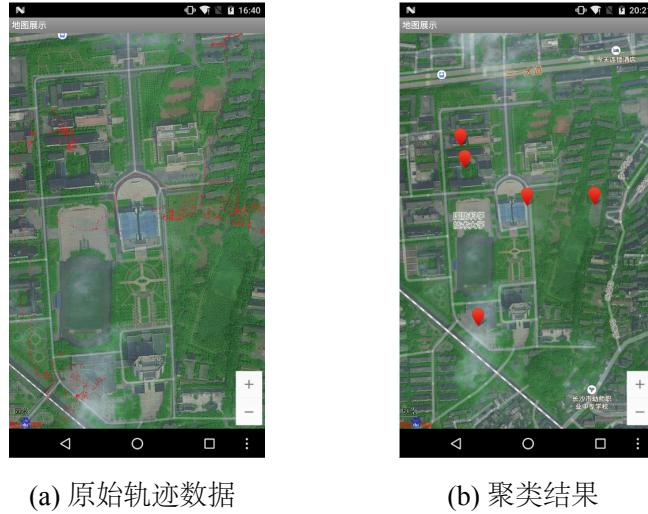


图 4.24 轨迹聚类实验结果地图表示 2

的结果上有的会出现一点误差，但是能够去除对数据和参数的敏感性减少对实验的干预，在本研究中显得更加难能可贵和适用。

本小节主要讨论了前面一章节中介绍的聚类算法在本研究中对用户轨迹聚类的效果以及每种聚类算法对参数的依赖以及影响，然后分析每一种聚类算法各自的优缺点，K-means 聚类结果虽然比较好但是需要预先设定聚类个数，最明显的问题是我们收集的用户轨迹数据中无法预先知晓用户轨迹所应该聚类的个数，导致结果出现偏差；DJ-Cluster 聚类算法同样存在类似的问题。本课题采用针对小数据进行聚类，一方面是因为数据量小比较容易发现较小的聚类簇，能够识别用户更加细粒度的访问位置，如果将用户多天的轨迹数据进行聚类，有可能将会导致较大量簇无法识别。另外采用小数据量时间切片进行分析聚类能够保证当前用户的某个聚类没有识别而下一段数据分析识别出来，形成效果叠加也能够保证最终结果的相对稳定，下一节将简要讨论如何对用户的轨迹点进行语义化。

#### 4.2.4 对用户轨迹的语义位置添加语义标签

在上面的小节中主要讲述了针对用户原始 GPS 轨迹数据进行滤波处理、轨迹停留点挖掘以及针对用户轨迹进行聚类。但是仅仅得到用户的空间轨迹信息是不够的，为此我们需要将用户轨迹中有意义的点添加上语义标签如图4.25所示，为用户轨迹中的停留点加上语义标签，得能有助于我们了解用户去了什么地方，这些地方属于什么类型等。语义轨迹有助于我们进一步分析用户之间的相似性，因此本小节将介绍如何添加语义轨迹。

现有的大多数工作是基于反地理编码技术来将空间轨迹语义化，将用户的位置转换为具体的道路门牌号等（如德雅路 138 号），然而有时候道路地址无法有效

地表示用户位置的现实意义；另外的方法采用了机器学习的方法，首选从用户的轨迹中提取用户的位置点以及位置点的类型，采用关系马尔科夫网模型，利用网中的节点表示用户的访问点，基于条件随机场对用户的访问点进行推测得到用户的访问点的语义标签，但是采用机器学习效率较低，同时最终结果还受限于数据训练集，不适合普适计算的要求。因此我们采用了以语义标签数据库为主，反地理编码为辅的语义化方法。语义标签数据库中覆盖了本研究对象校园内部及其周边大部分语义位置点，假设得到了用户一天的轨迹数据并且进行了停留点切分和聚类得到了 100 个位置，而我们的语义标签数据库中只能够查询到其中的 70 个位置的语义标签，因此剩下的 30 个位置我们将采用反地理编码进行语义化，并将相对于的语义标签存入到语义数据库中避免二次调用反地理编码接口。如图4.25所示，我们采用了三层了轨迹语义化处理模型，在最底层为用户的原始轨迹数据，中间层为上一小节中我们识别出的用户停留点或者聚类簇，最上层为用户的访问地点语义标签。

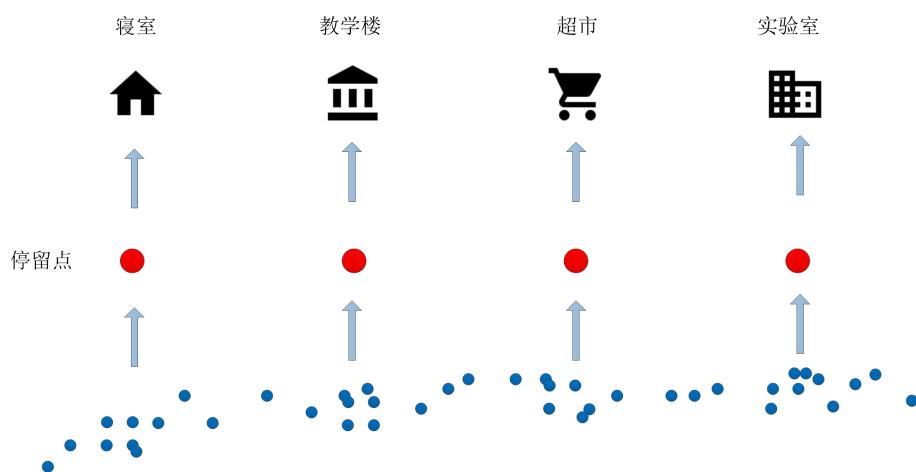


图 4.25 位置语义标签示意图

在语义化的过程中，我们首先构建了一个符合当前范围内的语义标签数据库，里面包含了学校内部各个位置语义点以及学校周边大部分的位置语义信息，在为用户轨迹添加语义标签时，首先查找语义标签数据库内是否有相匹配的位置点。具体的匹配过程如图4.26所示，图中首先我们需要人为设定一个距离阈值，用来衡量当前位置点与语义标签位置点之间的关系。图4.26中 a 所示：当停留点或簇中心点距离语义标签库中有且仅有一个位置点的距离小于设定的阈值，则把当前语义标签赋予给当前位置；若如图4.26中 b 所示，当半径阈值内不止一个符合要求的语义标签点时，我们选择和当前位置点距离最近的语义标签作为当前位置点的标签；当语义标签数据库中无法查询到与当前位置相匹配的标签点时，将会调用反地理编码，获取到当前位置的语义标签，结果示意图如图4.27所示，在

图4.27(a) 和 (b) 中, 因为这两处地理位置的语义标签都已经存在于语义标签数据库中, 所以查询时就返回数据库中的标签, 而 (c) 中的位置因为暂时没有保存在标签数据库中, 因此调用百度地图 API 将得到的中心点坐标转换为百度地图数据库中的标注地址并将结果返回和保存。

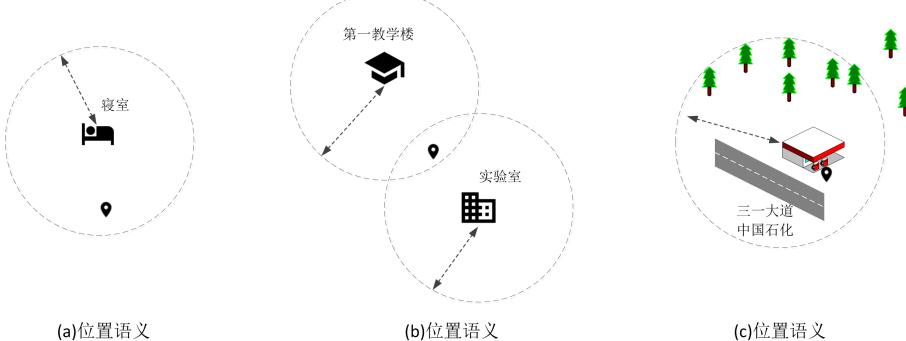


图 4.26 语义标签识别示意图



图 4.27 语义标签结果示意图

在用户轨迹语文化的过程中还存在以下这样的问题: 第一不同用户或者同一个用户访问同一个语义位置 (如食堂、学生宿舍) 等的时候, 由于手机设备不同和 GPS 数据采样的误差, 导致用户当前停留位置的聚类中心并不一定和数据库中记载的完全相同, 因此我们在查询语义数据库的时候, 需要设定一个中心点距离阈值比如说 15 米, 如果当前中心点位置和其中某一个标签的位置距离小于 15 米, 就将该语义标签赋予当前中心点; 如果同时落入了多个语义点的半径范围内, 则将距离最近的语义中心点的标签赋予给当前中心点。

另外一个问题是由于第一个问题引申而来即半径阈值的取值为多少才能给既准确的识别中心点语义标签又能使得临近语义点的距离足够大，这样才能有效的对轨迹停留点进行语义标注，在实验过程中发现基于小样本数据在小范围内（如整个校园内及其周围）的语义标签化得到的数据相对于可靠，可能是由于在小范围内的 GIS 语义标签数据库比较完备，能够有效地识别出用户大部分的停留点。但是如果将数据进一步增加，将活动范围扩大，那么这种方法的可靠性和得到的语义标签的准确性也将受到影响。如果能够得到一个大区域内的 GIS 信息库或者是人工标注好这样的语义标签库，那么就能够应对这样的问题，但是这样显得缺乏可行性，如何针对在大区域内的语义轨迹化也是我们未来工作的研究之一。

### 4.3 用户 GPS 轨迹数据结构化表示

通过前面章节的停留点检测算法以及实践，能够得到用户轨迹中存在的停留点，接下来我们要构建一个用户表示用户轨迹的模型，针对每个用户利用轨迹表示模型来描述和存储该用户所有时间内的轨迹信息。为了便于在下面章节中对基于空间轨迹的用户关系强度度量，我们采用了一种基于三维的层次化的 GPS 轨迹表示模型来描述用户在纵向时间维度内以及横向空间维度中的活动。如图4.28所示，在此抽象表示结构中，其中第一维表示用户维度，第二维的每一行表示用户在当前日期内所有的轨迹集合，第三维中的每一列表示用户在当前日期内，按照时间分割后的时间片中所有的停留点集合，最后得到的结果中  $Gtra_i$  表示用户  $U_i$  所有的轨迹集合。

根据前面章节中获取的用户位置信息多为一个经纬度点的集合或一个几何形状，没有包含任何可以直接识别的信息。通过轨迹语义化，我们基于用户的 GPS 轨迹得到了用户的语义位置轨迹。不同于用户地理轨迹的抽象表示方法，用户的语义轨迹表示更加具有难度和挑战性，根据前面的描述得知：现实生活中同一个用户多次访问同一语义位置所产生的空间位置停留点可能不相同 (GPS 定位存在偏移)，直接对停留点进行语义化并不可行，因此我们在节4.2.3中针对部分停留点进行聚类得到用户的轨迹中停留点的聚类簇，然后用当前的结果替换原有轨迹中对应的停留点。如图4.29所示，我们首先从用户 GPS 轨迹数据中抽象出表示用户空间地理位置的空间轨迹数据结构；然后将聚类后得到的停留点簇按照节4.2.4对停留点添加语义标签得到用户的语义轨迹。现实生活中，人们日常轨迹中的位置移动常常表现为从一个位置移动到另一个位置（如早上从寝室步行到食堂再去图书馆等），在图4.29中我们借鉴这样的想法将用户每天的 GPS 运动轨迹向量序列化后的结果赋值为用户的轨迹向量，再根据时间戳信息结合语义标签构建用户的语义轨迹时间向量，得到用户在一段时间片内的语义轨迹向量，最后将每天的轨

迹向量合并得到用户的语义轨迹序列描述，通过对这种语义轨迹的结构化表示，可以有效地将空间轨迹转换为用户的日常语义轨迹，便于下一步计算用户之间的关系强度。

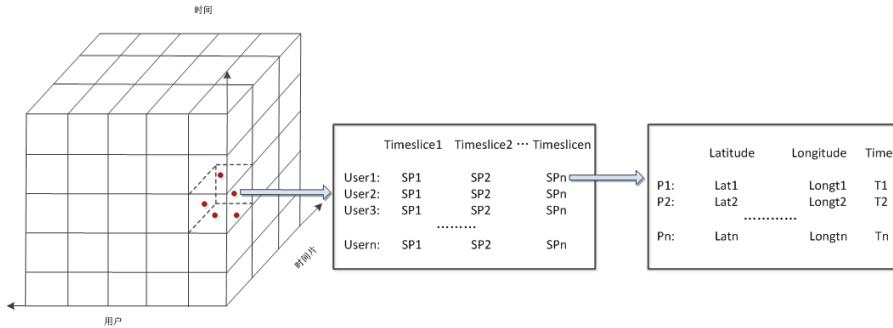


图 4.28 用户轨迹的抽象表示

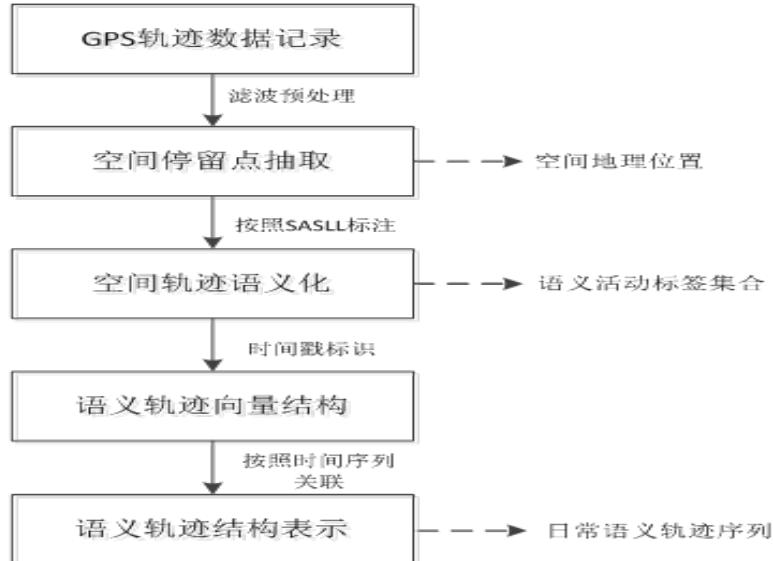


图 4.29 用户语义轨迹描述模型

在获得用户的语义轨迹之后我们将用户的语义轨迹序列按照时间片进行划分，得到形如  $\{Sloc_1, Sloc_2, \dots, Sloc_n\}$  结构的用户语义轨迹序列集合，最终得到的语义轨迹结构如图4.30中的示例描述。

从用户的历史轨迹数据中所挖掘出的频繁模式和序列模式能够反映出用户的日常轨迹运动习惯和行为规律，运动模式在现实生活中表现为用户经常行走的路径序列，是用户轨迹数据规律的抽象表示。用户的日常运动模式在一定程度上代表了用户的个人喜好、意图以及活动模式，例如用户 A 经常下午去操场跑步，B 周末经常去市区逛街等，当从一个更细的粒度甚至能够根据用户的用餐地点推测

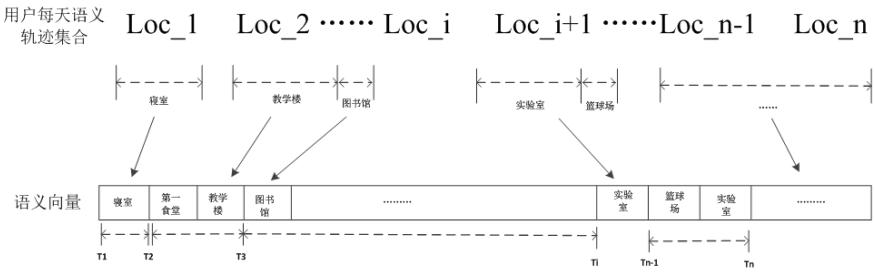


图 4.30 语义轨迹示意图

出用户的口味喜好等。本文首先采用频繁模式和序列模式挖掘算法来探寻用户的日常轨迹运动模式，最后得到用户的运动模式矩阵，矩阵中的行

$$TraFP_{i,j,k} = \{(Sloc_1), (Sloc_2, \dots, Sloc_m) \dots\}$$

表示用户  $u_i$  在时间周期  $k$  内的日常语义轨迹中所挖掘出的频繁模式的第  $j$  项纪录；而

$$TraSQ_{i,j,k} = \{< Sloc_1 >, < Sloc_2, \dots, Sloc_n > \dots\}$$

表示用户  $u_i$  在时间周期  $k$  内的日常语义轨迹中所挖掘出的系列模式的第  $j$  项纪录，这些都将会在下一章中进行详细的描写如何利用用户轨迹运动模式来计算用户之间的关系强度。

#### 4.4 WiFi 感知数据处理计算技术

在日常生活中 WiFi 的感知距离因为受到发射器功率以及周围环境的影响，导致正常 WiFi 的覆盖范围大约在 20 米左右。当用户处在不同的上下文环境中时其周围能够被 StarLog 所感知到的 WiFi 环境也是不同的，不仅如此，对于一个很小的范围内（如教室、图书馆、办公室等）的 WiFi 上下文环境基本是极度相似的甚至相同的，也就是说能够探明检测到的 WiFi 列表应该是相同的，因此我们根据 WiFi 感知数据的这一特性来计算用户之间的关系强度。根据图3.3中的处理流程描述，首选我们将原始 WiFi 感知数据进行处理，从中提取出有用的数据信息（如 WiFi 接入名称、WiFi 强度、WiFi 接入点的 MAC 地址、扫描时间等）；其次在提取出的数据的基础之上，我们根据探测的时间戳信息，将 WiFi 数据序列化。如图4.31所示，我们采用矩阵的方式来表示所有用户在整个数据收集期间的 WiFi 序列集合，其中每一行表示实验数据收集的时间维度从第一天到第 N 天；每一列代表一个独立的用户，其中  $[i, j]$  表示用户  $u_j$  在第  $i$  天的 WiFi 向量数据。每一条 WiFi 向量数据都是由许多切片时间内的 WiFi 上下文（WiFi context）所构成的。

每一次所感知的 WiFi 上下文环境信息是不仅包含了 WiFi 连接名，还包含了接入点 MAC 地址、信号强度、设备类型等复杂的信息，如图4.32所示，WiFi-

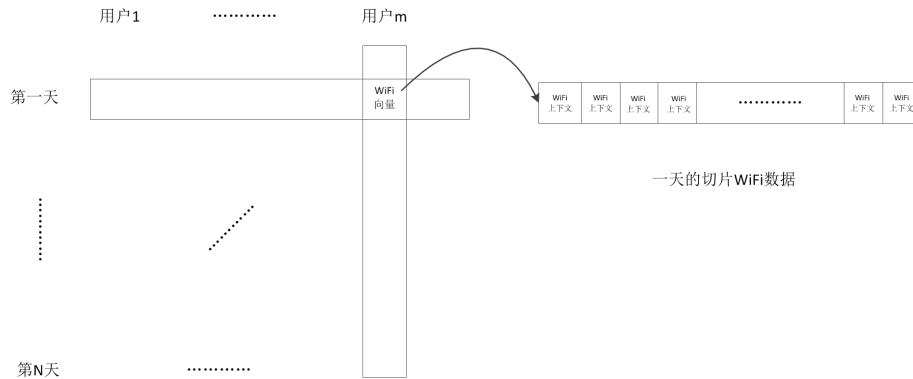


图 4.31 WiFi 数据结构化表示示意图

context 更形象的描述是当前 WiFi 数据收集时刻时，设备周围的感知信息和环境信息，本课题转换视角将 WiFi 感知序列拟物化，看做我们现实中生活的环境，考虑到当用户 1 当前时刻所处的环境和用户 2 所处的环境极度相似或者相同的时候，那么我们就认为二者在同一个环境中也就是产生了现实交互比没有在同一环境中的用户具有更强的关系强度。因此我们将当前时刻的环境用无向带权图来表示，即假设每一个能够探知的 WiFi 源都与设备存在一条边，然后用户在一段时间内的数据就成了一张图如图4.33，最后比较两个用户的在每一个时间切片下的图的相似性推断出用户的关系强度。

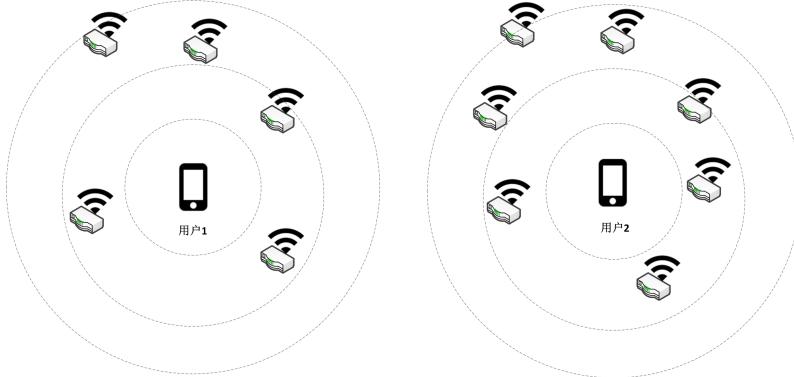


图 4.32 WiFi 上下文抽象化表示示意图

## 4.5 蓝牙感知数据处理计算技术

相比于 WiFi 设备的感知距离，蓝牙设备的感知距离进一步受限缩减为 5 到 10 米左右，在这样短距离的情况下，则可以考虑二者有现实生活中的交互，然而这种方法是有局限的即如果在一时刻内设备能够搜索到多个蓝牙设备而有未和其中任何一个设备进行连接，那么计算用户之间的相似度就不能采用上述的方法。因此我们采用节4.4中相似的数据抽象化方法，将收集到的用户蓝牙感知数据进行按照图3.3中的流程将蓝牙数据进行结构化清洗并结构化存储如图4.34所示，

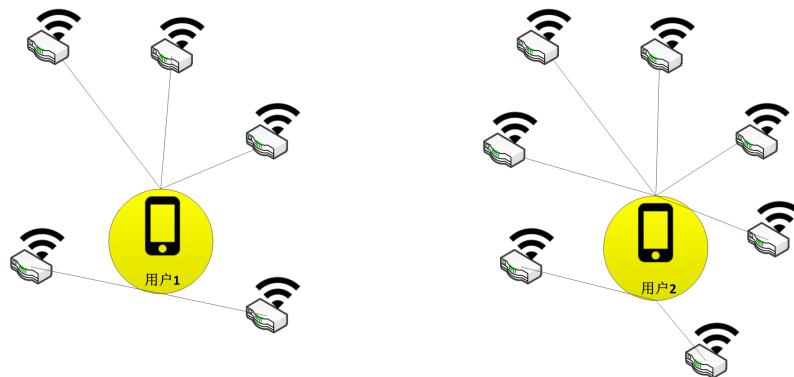


图 4.33 WiFi 图结构抽象化表示

得到蓝牙的上下文感知信息表示如图4.35，最后我们同样采用图的数据结构对蓝牙上下文信息进行存储，如图4.36

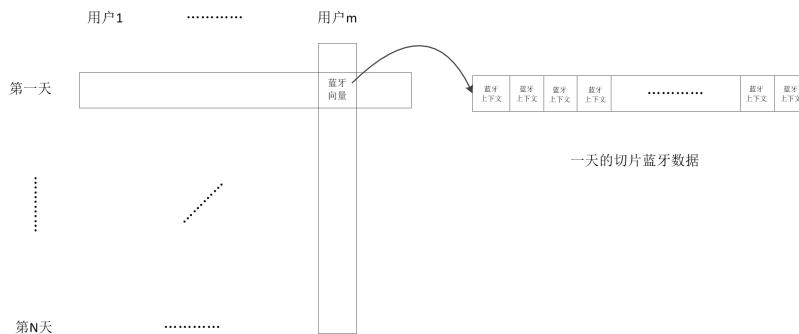


图 4.34 蓝牙数据结构化表示示意图

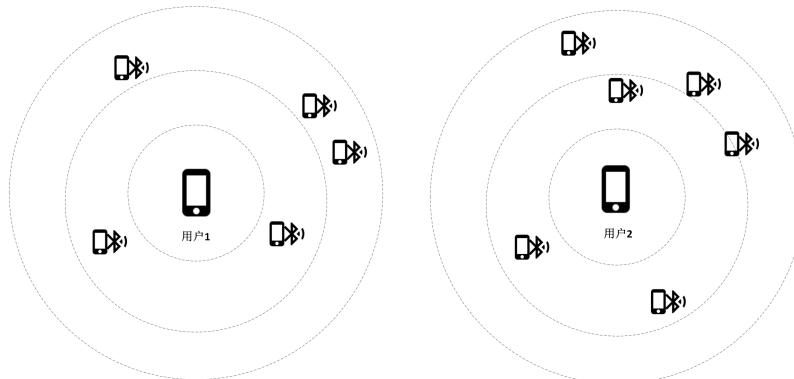


图 4.35 蓝牙数据上下文抽象化表示示意图

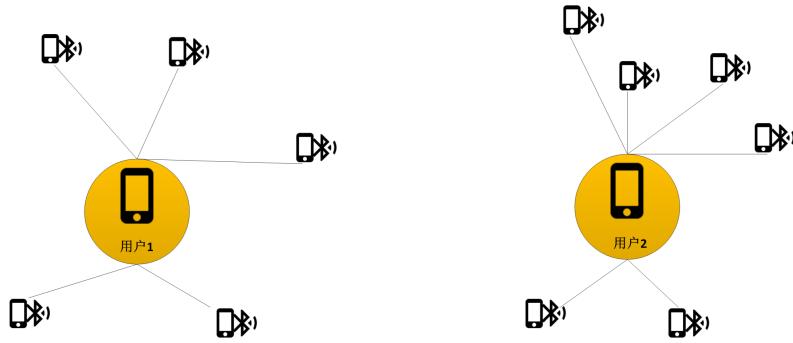


图 4.36 蓝牙数据图结构抽象化表示

## 4.6 小结

本章主要描述了 RSMHD 计算框架中的三个重要组成部分即基于 GPS 轨迹数据的处理和计算、基于 WiFi 感知数据的处理和计算和基于蓝牙感知数据的处理和计算。在本章中首先对 GPS 轨迹数据预处理方法展开了详细的分析，分别对常用的滤波：中值滤波、均值滤波、卡尔曼滤波进行了 GPS 轨迹滤波和噪音点剔除，在经过对实验结果的对比之后发现采用时间片切分的 GPS 轨迹卡尔曼滤波能够得到比较好的效果；然后描述了用户 GPS 轨迹中停留点的检测，通过停留点检测算法识别出用户轨迹中的停留点并为后续工作做好数据准备；接下来介绍了几种常用的聚类算法：K-means 聚类算法、DJ-Cluster 密度聚类算法以及 Science 改进的密度聚类算法对用户 GPS 轨迹数据中的停留点进行聚类分析，得到用户轨迹的聚类簇，并针对每种聚类算法和实验的结果来分析每一种聚类算法在轨迹点聚类问题上的优势和不足之处；在获取到用户轨迹停留点和停留点聚类簇的基础上阐述了停留点的语义化过程，并进行了相对应的结果展示；在本章的最后三章节中，分别描述了本研究中对用户 GPS 轨迹数据的结构化表示存储、WiFi 感知数据的处理和抽象结构化表示方法和蓝牙感知数据的处理和抽象结构化表示方法，并详细的用流程图和模型图进行了更加详细的过程描述。下一章将会详细描述基于本章中划分的模型进行用户关系强度度量的计算方法。

## 第五章 RSMHD 用户关系强度计算方法概述

在第四章中从 RSMHD 用户关系强度计算框架的三个重要组成模块出发来对基于不同的智能手机感知数据源的数据处理、数据结构模型构建和表示等进行了详细的描述介绍，在这一章中将重点介绍基于 RSMHD 模型中每一种非社交感知数据的用户关系强度计算方法以及最后采用集成学习思路来进行结果的融合。

### 5.1 用户关系强度计算方法概述

如图3.1中所示，RSMHD 多维数据源多层次的用户关系度量框架主要从三种不同的感知数据源出发，第一层是用户上下午感知数据收集阶段，这是整个实验研究中数据的来源和信息反馈展示的终端平台。

第二层是基于用户的日常轨迹计算用户之间的关系强度模型，在本模块中，详细的模块框架结构如图3.2所示，在基于用户轨迹数据的计算过程中，我们进一步从用户轨迹数据的三个维度出发：用户的空间轨迹推导用户之间的关系强度、用户的语义轨迹关系推导用户之间的关系强度、用户日常轨迹中挖掘出的运动模式来推导用户之间的关系强度。

第三层度量是基于用户手机的 WiFi 感知数据源，通过对 WiFi 感知数据的处理和结构化建模表示，本研究创新性的采用图结构来还原模拟出符合现实环境中某一时刻设备感知 WiFi 数据的周围的环境信息。因此在计算用户之间 WiFi 数据序列的相似度时，就将问题转换为计算每一个对应时刻内的 WiFi 感知上下文环境信息的相似度，进而来度量用户之间的社交关系强度。

在第四层中用户关系度量是基于用户手机的蓝牙感知数据源，通过对蓝牙感知数据的处理和结构化建模表示，同样采用图结构来还原模拟出符合现实环境中某一时刻设备感知的蓝牙数据的周围的环境信息，在计算用户之间蓝牙数据序列的相似度时，就将问题转换为计算每一个对应时刻内的蓝牙感知上下文环境信息的相似度，进而来度量用户之间的社交关系度。

RSMHD 模型从三种不同的数据源即用户日常轨迹、用户手机 WiFi 感知数据、用户手机蓝牙感知数据抽象出三种不同的层次度量模型，在基于用户轨迹度量层次中，又分解出三种度量维度，三种维度由低到高，由细粒度到粗粒度的度量使这一层次的计算结果更加有效和具有说服力，并最终根据这三层数据源的度量结果，借鉴集成学习的方法进行多层结果的融合。因此 RSMHD 以摒弃了单一的感知数据源采用多层次数据源的用户关系计算并且最终对每层结果进行融合，能够更加有效全面的推理出现实生活中用户之间的社交关系强度。

## 5.2 RSMHD 框架模型的数据输入

在前面的章节中我们详细讲述了每种数据源的数据结构表示以及基于不同数据源的用户关系强度计算，如果将 RSMHD 看作是一个计算用户关系强度的函数的话，那么用户日常轨迹数据、用户手机 WiFi 感知数据、用户手机的蓝牙感知数据就是需要输入的未知变量，最终经过函数计算输出用户之间的社交关系强度结果。因此接下来我们将用部分篇幅来介绍每种数据源的具体清洗提取以及输入样式。

### 5.2.1 轨迹数据的预处理

随着各种定位技术（如全球定位系统 GPS 和 WLAN 和移动网络）的发展和内嵌的定位模块，通过智能手机可以准确的获取用户的位置信息，但是在用户的轨迹信息的采集过程中，因受到地形、气候、GPS 传感器和 SA 干扰误差的影响，会导致得到的用户位置与真实位置之间产生误差，GPS 的位置误差使得用户的轨迹数据中存在大量的噪音数据，影响后续对数据的处理和分析工作。因此，为了保证后续计算的顺利开展和结果的准确性，我们首先要对收集到的用户轨迹数据进行噪音点的剔除处理，然后才能开展下一步工作。

首先假设我们收集的感知数据的用户集合用  $U$  表示，可以得知集合  $U$  的表达式为  $U = \{u_1, u_2, \dots, u_n\}$ ，其中下标  $i$  表示第  $i$  个用户。同时，因为有的志愿者用户可能因为某些原因没有能够在某一天采集数据，我们需要建立一个表  $Date$  来表示采集当前用户感知数据的时间， $Date_i$  代表用户  $u_i$  在整个数据采集周期内的具体日期，表示为  $Date_i = \{d_1, d_2, \dots, d_m\}$ ， $Tra$  表示用户的所收集的 GPS 轨迹数据集，表示为  $Tra = \{Tra_1, Tra_2, \dots, Tra_n\}$ ，在集合中  $Tra_i$  表示用户  $u_i$  在所有天中所收集的 GPS 轨迹数据的集合， $Tra_{i,k}$  则表示用户  $u_i$  在  $d_k$  这一天所收集到的 GPS 轨迹数据集合（其中  $d_k \in Date_i$ ），我们将在接下来的小节中，针对轨迹计算的三种维度不同粒度下的轨迹数据输入进行描述。

### 5.2.2 用户空间 GPS 轨迹数据处理和输入

根据前面章节的需求描述，我们需要对用户采集的每天的 GPS 轨迹数据  $Tra_{i,k}$  进行异常点的剔除，降低因为 GPS 误差所带来的影响。针对用户  $u_i$  每天收集的数据  $Tra_{i,k}$ ，我们将原始轨迹数据按照时间片进行划分（这里我们取时间片为一小时），因此我们得到了  $Tra_{i,k}$  按照时间片分割的每天 24 份的轨迹数据，用  $Sli\_Tra$  表示， $Sli\_Tra_{i,k} = \{Sli\_Tra_{i,k,1}, \dots, Sli\_Tra_{i,k,24}\}$ ，由数据的来源可知  $Sli\_Tra_{i,k,m} \in Tra_{i,k}$ ，接下来根据时间片划分的数据将每段数据采用卡尔曼滤

波，识别空间轨迹中的噪音点，减少 GPS 轨迹中异常点带来的影响，得到了滤波后的用户分片轨迹  $Ntra_{i,k,s}$ ，将众切分的轨迹组合起来得到用户当天 GPS 轨迹滤波后的结果  $Ntra_{i,k}$ ，可知  $Ntra_{i,k} = \{Ntra_{i,k,1}, Ntra_{i,k,1}, \dots, Ntra_{i,k,24}\}$ ，至此我们得到了在基于用户轨迹计算用户社交关系强度这一层的第一维数据的输入。

### 5.2.3 用户语义轨迹数据的处理和输入

当我们得到用户滤波后的 GPS 轨迹  $Ntra_{i,k}$  后，通过停留点检测算法2.1我们对原始轨迹进行停留点检测，得到了  $Ntra_{i,k}$  的停留点集合  $SP\_tra_{i,k}$ ，表示为  $SP\_tra_{i,k} = \{sp_{i,k,1}, sp_{i,k,2}, \dots, sp_{i,k,n}\}$ ，然后在针对每天的轨迹停留点进行聚类得到轨迹的聚类簇，连同  $SP\_tra_{i,k}$  计算得到用户  $u_i$  每天的语义轨迹序列  $Ltra_{i,k}$ ，表示为  $Ltra_{i,k} = \{Loc_1, Loc_2, \dots, Loc_m\}$ ，其中  $Loc_i$  和  $Loc_j$  可能相同也可能不同。按照前面小节中的处理方法，我们同样对用户每天的语义轨迹序列按照时间片进行划分，将  $Ltra_{i,k}$  按照每天的时间划分为 24 段（本研究中采用每小时作为时间片切分刻度，是考虑到本研究的对象中学生的分布情况以及不同学生年级分布中每天课程影响等因素，如果将时间刻度缩小至半小时有可能导致重复切分或者将完整的语义轨迹肢解破坏语义轨迹的上下文推理性）得到每段时间内的用户语义轨迹  $sli\_Ltra_{i,k,m}$ ，用户每天语义轨迹的分段集合表示为  $Sli\_Ltra_{i,k} = \{sli\_Ltra_{i,k,1}, sli\_Ltra_{i,k,2}, \dots, Sep\_ltrace_{i,k,24}\}$ ，在采用快速语义 Hash 算法计算用户语义轨迹之间的相似度的时候，我们在准备用户  $u_i$  和用户  $u_j$  的语义轨迹输入时首先需要按照对应的轨迹数据收集日期进行输入，将相同时期  $d_t$  下的数据  $Sli\_Tra_{i,t}$  和  $Sli\_Tra_{j,t}$  作为当前计算的输入；然后根据语义轨迹的时间片刻度对相应的  $Sli\_Tra_{i,t,s}$  和  $Sli\_Tra_{j,t,s}$  继续建模变换得到当前时间片下的语义轨迹向量  $SliVec\_Tra_{i,t,s}$  和  $SliVec\_Tra_{j,t,s}$ ，其中  $SliVec\_Tra_{i,t} = \bigcup_{s=1}^{24} SliVec\_Tra_{i,t,s}$ ，然后将  $SliVec\_Tra_{i,t}$  作为当前语义轨迹相似度计算的输入，通过时间刻度内语义轨迹向量的计算的到最终的语义轨迹相似度，计算出用户之间的关系强度。

### 5.2.4 用户运动模式的处理和输入

现用户在日常生活中常有规律的访问相似的路径，说明用户的轨迹运动模式能反映出用户之间的相似性。从用户的历史轨迹数据中所挖掘出的频繁模式和序列模式能够反映出用户的日常轨迹运动习惯和行为规律，运动模式在现实生活中表现为用户经常行走的路径序列，是用户轨迹数据规律的抽象表示。根据上文得到的用户  $u_i$  每天的语义轨迹  $Ltra_{i,k}$ ，我们同样按照时间刻度单位对用户每天的语义轨迹集合进行频繁模式挖掘和序列模式挖掘，得到用户  $u_i$  的关于第  $k$  天语义轨迹的频繁模式序列  $TraPF_{i,k}$ ，其中  $TraPF_{i,k} = \{TraPF_{i,k,1}, TraPF_{i,k,2}, \dots, TraPF_{i,k,n}\}$ ，项  $TraPF_{i,k,m}$  表示当前计算出的轨迹频繁

模型的第  $m$  项频繁表达式记为  $TraPF_{i,k,m} = \{(Loc_1), (Loc_1, \dots, (Loc_m), \dots)\}$ ；用户  $u_i$  的关于第  $k$  天语义轨迹的序列模式结果表示为  $TraSQ_{i,k}$ ，其中  $TraSQ_{i,k} = \{TraSQ_{i,k,1}, TraSQ_{i,k,2}, \dots, TraSQ_{i,k,n}\}$ ，项  $TraSQ_{i,k,m}$  表示当前计算出的轨迹频繁模型的第  $m$  项频繁表达式记为  $TraSQ_{i,k,m} = \{<Loc_1>, <Loc_1, \dots, Loc_m>, \dots\}$ ，然后将  $TraFP_i$  和  $TraSQ_i$  作为当前维度计算的数据输入，计算用户基于轨迹运动模式的相似度，将其作为用户基于 GPS 轨迹关系强度计算结果之一输出。

### 5.2.5 用户 WiFi 感知数据的处理和输入

WiFi 和蓝牙作为一种主动对外界感知和探索所得到的感知信息，已经证实能够被用来分析用户之间的社交关系。在日常生活中 WiFi 的感知距离因为受到发射器功率以及周围环境的影响，导致正常 WiFi 的覆盖范围大约在 20 米左右。当用户处在不用的上下文环境中时其周围能够被 StarLog 所感知到的 WiFi 环境也是不同的。我们首先针对用户每天所收集的 WiFi 感知数据进行处理，假设搜集的所有 WiFi 感知上下文数据为  $WiFi\_D$ ，进一步表示为  $WiFi\_D = \{WiFi\_D_1, WiFi\_D_2, \dots, WiFi\_D_n\}$ ， $WiFi\_D_i$  表示用户  $u_i$  在所有时间  $Date_i$  内所采集的 WiFi 上下文感知数据。用户  $u_i$  每天采集的数据我们用  $WiFi\_D_{i,k}$  表示其在日期为  $k$  天时采集的 WiFi 感知数据，对于用户在第  $k$  天的数据，我们将会按照时间片进行切分（这里将会采用更加细粒度的时间，以为计算的可信性保证）将用户每天的 WiFi 感知数据按照 15 分钟的时间刻度划分为 96 份，用  $Sli\_WiFi$  表示， $Sli\_WiFi_{i,k} = \{Sli\_WiFi_{i,k,1}, Sli\_WiFi_{i,k,2}, \dots, Sli\_WiFi_{i,k,96}\}$ ，由数据的定义可知  $Sli\_WiFi_{i,k,m} \in WiFi\_D_{i,k}$ 。对于每一个感知时刻  $t$  得到的 WiFi 上下文数据  $Sli\_WiFi_{i,k,t}$ ，我们采用采用图4.33中的图形数据结构进行处理，将  $Sli\_WiFi$  作为基于 WiFi 感知数据计算用户之间社交关系的数据输入。

### 5.2.6 用户蓝牙感知数据的处理和输入

相比于 WiFi 设备的感知距离，蓝牙设备的感知距离进一步受限缩减为 5 到 10 米左右，在这样短距离的情况下，则可以考虑二者有现实生活中的交互。当用户处在不用的上下文环境中时其周围能够被 StarLog 所感知到的蓝牙环境也是不同的。我们首先针对用户每天所收集的蓝牙感知数据进行处理，假设搜集的所有蓝牙感知上下文数据为  $Bth\_D$ ，进一步表示为  $Bth\_D = \{Bth\_D_1, Bth\_D_2, \dots, Bth\_D_n\}$ ， $Bth\_D_i$  表示用户  $u_i$  在所有时间  $Date_i$  内所采集的蓝牙上下文感知数据。用户  $u_i$  每天采集的数据我们用  $Bth\_D_{i,k}$  表示其在日期为  $k$  天时采集的蓝牙感知数据，对于用户在第  $k$  天的数据，我们将会按照时间片进行切分（这里将会采用更加细粒度的时间，以为计算的可信性保证）将用户每天的蓝牙感知数据按照 15 分钟的时间刻度划分为 96 份，用  $Sli\_Bth$  表示， $Sli\_Bth_{i,k} =$

$\{Sli\_Bth_{i,k,1}, Sli\_Bth_{i,k,2}, \dots, Sli\_Bth_{i,k,96}\}$ , 由数据的定义可知  $Sli\_Bth_{i,k,m} \in Bth\_D_{i,k}$ 。对于每一个感知时刻  $t$  得到的蓝牙上下文数据  $Sli\_Bth_{i,k,t}$ , 我们采用图4.36 中的图形数据结构进行处理, 将  $Sli\_Bth$  作为基于蓝牙感知数据计算用户之间社交关系的数据输入。

在准备好了 RSMHD 模型中各个计算模块的输入数据之后, 接下来我们将根据每个模块中的输入数据详细介绍如何经过 RSMHD 的处理计算输出之间的社交关系强度。

### 5.3 RSMHD 计算用户支之间的关系强度

在前面的节5.2中, 我们针对 RSMHD 框架中的三种计算模型的感知数据输入进行了详细的定义描述, 在本小节中, 将针对每一种感知源数据的用户关系强度计算方法进行描述。

#### 5.3.1 基于用户轨迹数据的关系强度计算

在这里我们需要计算用户集  $U$  中的用户  $u_i$  与用户  $u_j$  之间基于位置轨迹感知数据而得到的关系强度 ( $u_j \in U, u_j \neq u_i$ )。根据图??以及图??中的计算模块的详细流程描述, 我们需要分别从三个维度, 三种数据粒度来计算轨迹数据之间的相似性并得出用户之间的关系强度结果。

##### 5.3.1.1 基于用户 GPS 轨迹数据的关系强度计算

针对采集到的用户 GPS 轨迹数据  $Tra$ , 经过滤波、时间片切分后我们得到了每位用户经过处理后的位置轨迹数据集合  $Ntra_i$ , 我们采用加权的 DTW 算法来计算基于  $Ntra$  的用户间的关系强度。对于需要计算的对象用户  $u_i$  与用户  $u_j$ , 我们分从轨迹数据集  $Tra$  中找出各自所对应的轨迹数据, 并处理得到  $Ntra_i$  和  $Ntra_j$ , 然后以自然日为比较单位, 计算  $Ntra_{i,x}$  和  $Ntra_{j,y}$  之间的相似度, 其中  $xDate_i$ ,  $yDate_j$ , 分别表示用户  $u_i$  与  $u_j$  在时间日期为  $x$  和  $y$  当天收集的位置轨迹数据。为了便于识别用户轨迹数据是否是相同日期, 我们令  $W_{x,y} = 1$  如果当前  $x = y$ , 否则  $W_{x,y} = 0$ , 调用函数  $DTW(Ntra_{i,x}, Ntra_{j,y})$  计算返回用户  $u_i$  在日期为  $x$  和用户  $u_j$  在日期为  $y$  这天的空间轨迹相似度, 具体计算用户之间基于空间轨迹的关系强度的数学公式描述如公式5.1, 最后根据轨迹序列长度对结算结果进行加权处理。

$$RS_{dtw}(u_i, u_j) = \sum_{x,y}^{Date_i, Date_j} W_{x,y} \sum_{s=1}^{24} \frac{DTW(Ntra_{i,x,s}, Ntra_{j,y,s})}{Max(|Ntra_{i,x,s}, Ntra_{j,y,s}|)} \quad (5.1)$$

##### 5.3.1.2 基于用户日常语义轨迹的关系强度计算

基于空间轨迹的度量能够反映出用户在地理空间上的相遇或者相邻, 基于语义轨迹的度量能够进一步得出用户之间的关系强度。通过将用户的空间轨迹语义

化得到语义化轨迹序列  $Ltra$ ,  $Ltra_i$  代表前小节中得到的用户  $u_i$  的语义轨迹序列集合。 $W_{i,j,x,y,s,t}$  表示用户  $u_i$  在日期为  $x$  这天的第  $s$  个时间片下的语义轨迹序列和用户  $u_j$  在日期为  $y$  这天的第  $t$  个时间片下的语义轨迹序列, 由已有知识可知, 当且仅当  $x = y$  且时间片  $s$  和  $t$  下均有数据时,  $W_{i,j,x,y,s,t} = 1$ , 否则  $W_{i,j,x,y,s,t} = 0$ 。我们将用户每段时间内的语义轨迹看做是一条语句, 每一个语义位置标签  $Loc_m$  当做句子分词后的词语, 同时将每个语义标签的出现次数作为此语义标签的权值; 接下来通过 hash 函数计算每一个词语  $Loc_m$  的 n-bit 签名 (n 的值需要参考整篇文档中的词语数量); 然后采用  $w = \text{词语hash} * \text{词语权值} * (-1)^{\text{bit}}$  对每个词语的签名进行加权处理,  $(-1)^{\text{bit}}$  的作用是当 hash 中 bit 位为 0 时结果乘以 -1, 当 bit 为 1 时结果乘以 1; 接下来就将当前时间段内的语义标签进行合并, 将标签 hash 进行相加得到并对相加结果进行逐位扫描 (如果位大于 0 就令当前位为 1, 否则为 0) 最后计算它们的余弦相似度得到结果, 变换流程如图5.1所示, 相似度计算公式如5.2所示。

$$RS_{Ltra}(u_i, u_j) = \sum_{x,y}^{Date_i, Date_j} W_{i,j,x,y,s,t} \sum_{s,t=1}^{24} \cos(\text{quick\_hash}(Ntra_{i,x,s}, Ntra_{j,y,t})) \quad (5.2)$$

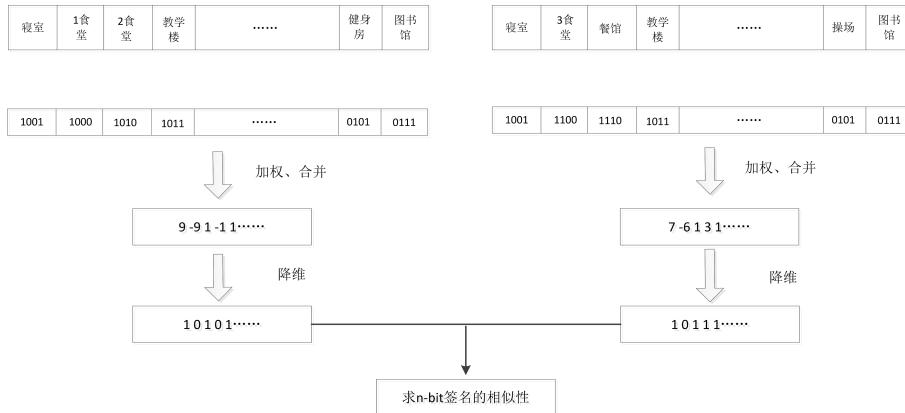


图 5.1 用户语义轨迹快速 hash 变换计算示意图

### 5.3.1.3 基于用户轨迹运动模式的关系强度计算

人们的日常活动具有很强的时间和空间的规律性, 因此用户的轨迹运动模式能反映出用户之间的相似性。我们根据前面小节中得到的用户  $u_i$  语义轨迹的频繁模式  $TraFP_i$  和序列模式  $TraSQ_i$ , 为确保计算结果的准确性, 假设  $W_{i,j,x,y}$  表示用户  $u_i$  与  $u_j$  在时间日期为  $x$  和  $y$  计算得到的语义轨迹频繁模式或序列模式的存在性权值, 当且仅当  $x = y$  且当天得到的轨迹运动模式不为空时取  $W_{i,j,x,y} = 1$  否则取  $W_{i,j,x,y} = 0$ 。在计算过程中, 分别将用户  $u_i$ 、 $u_j$  的轨迹运动模式结果  $TraFP_i$  和  $TraFP_j$ 、 $TraSQ_i$  和  $TraSQ_j$  做比较, 通过比较在同一日期所得到的轨迹运动模

型相似度来推测计算用户之间的关系强度，用户在日常活动轨迹中访问某个位置的次数以及持续时间导致了在最终计算过程中每段位置的贡献度不同，因此本文采用语义位置的持续时间以及频率对结果进行加权，得到最终的加权后的轨迹运动模式， $TraFP$  加权计算算法如算法5.1所示。

---

**算法 5.1 用户轨迹运动模式加权算法**


---

**已知：**用户的语义轨迹  $Ltra_{i,k}$

**求：**加权后的用户轨迹运动模式  $TraFP_{i,k}$

```

1:  $L_1 = get_1FP(Ltra_{i,k})$ 
2:  $k = 2$ 
3: while  $L_k \neq \emptyset$  do
4:    $C_k = apriori(L_k)$ 
5:   for each set in  $SP\_tra_{i,k}$  do
6:      $C_{temp} = C_k \cap set$ 
7:     for each item in  $C_{temp}$  do
8:        $c.num += w_{temp}$ 
9:     end for
10:   end for
11:    $L_k = \{c \in C_k | c.num / \sum_{i=1}^n w_n \geq MinSup\}$ 
12: end while
13:  $TraFP_{i,k} = TraFP_{i,k} \cup L_k$ 
14: return  $TraFP_{i,k}$ 

```

---

其中  $C_{temp}$  代表频繁项集的候选项， $w_{temp}$  则表示语义轨迹  $Ltra_{i,k,temp}$  的权值，即最长语义标签时间除以时间窗口值与记录中不同语义标签数量的倒数乘积，用户关系强度计算公式如5.3。

$$RS_{TraFP}(u_i, u_j) = \sum_{x,y}^{Date_i, Date_j} p_{x,y} W_{i,j,x,y} |TraFP_{i,x} \cap TraFP_{j,y}| \quad (5.3)$$

公式5.3中  $p_{x,y}$  是对计算过程进行加权的权值， $p_{x,y} = 1 - p' \log_2^{p'}$ ， $p'$  是由公式5.4计算得到的。

$$p' = \frac{|TraFP_{i,x} \cap TraFP_{j,y}|}{|TraFP_{i,x} + TraFP_{j,y}|} \quad (5.4)$$

最后我们将每个维度所计算的结果进行合并得到用户  $u_i$  和  $u_j$  基于 GPS 轨迹数据的关系强度计算结果如公式5.5，其中  $\sum_{i=1}^n \frac{1}{w_i} = 1$ ，为了便于计算我们可以取  $\frac{1}{w_i} = \frac{1}{n}$ 。

$$RS_{Tra} = \frac{1}{w_1} RS_{dtw}(u_i, u_j) + \frac{1}{w_2} RS_{Ltra}(u_i, u_j) + \frac{1}{w_3} RS_{TraFP}(u_i, u_j) \quad (5.5)$$

### 5.3.2 基于用户 WiFi 感知数据的关系强度计算

根据节5.2.5中的 WiFi 输入数据  $WiFi\_D$ ，我们对用户  $u_i$  和  $u_j$  分别基于 WiFi 感知数据  $WiFi\_D_{i,x}$  和  $WiFi\_D_{i,y}$  进行关系强度计算， $W_{i,j,x,y,s,t}$  表示用户  $u_i$  在日期为  $x$  这天的第  $s$  个时间片下的 WiFi 感知上下文数据和用户  $u_j$  在日期为  $y$  这天的第  $t$  个时间片下的 WiFi 感知上下文数据，由已有知识可知，当且仅当  $x = y$  且时间片  $s$  和  $t$  下均有数据时， $W_{i,j,x,y,s,t} = 1$ ，否则  $W_{i,j,x,y,s,t} = 0$ 。根据图4.33中所展示的 WiFi 数据的图结构表示，我们可以得到基于 WiFi 感知数据的用户关系强度计算方法如公式5.6所示。

$$RS_{WiFi}(u_i, u_j) = \sum_{x,y}^{Date_i, Date_j} W_{i,j,x,y,s,t} \sum_{s,t=1}^{96} \frac{C}{|\Gamma(a_i)| |\Gamma(b_j)|} \sum_m^{|\Gamma(a_i)|} \sum_n^{|\Gamma(b_j)|} s(\Gamma_m(a_i), \Gamma_n(b_j)) \quad (5.6)$$

公式5.6中  $\Gamma(a_i)$  表示图中所有指向用户  $u_i$  设备的 WiFi 节点的集合， $\Gamma(b_j)$  表示图中所有指向用户  $u_j$  设备的 WiFi 节点的集合。 $s(\Gamma_m(a_i), \Gamma_n(b_j))$  表示两个用户设备当前 WiFi 上下文环境的相似，当  $\Gamma(a_i) = \Gamma(b_j)$  时候， $s(\Gamma_m(a_i), \Gamma_n(b_j)) = 0$ ，当且仅当  $\Gamma(a_i) = \Gamma(b_j)$  时， $s(\Gamma_m(a_i), \Gamma_n(b_j)) = 1$ 。

### 5.3.3 基于用户蓝牙感知数据的关系强度计算

在前面章节中我们详细描述了将用户蓝牙感知数据参照 WiFi 感知数据计算关系强度的方法来计算用户之间的关系强度，根据节5.2.6中的用户蓝牙数据  $Bth\_D$ ，我们对用户  $u_i$  和  $u_j$  分别基于 WiFi 感知数据  $Bth\_D_{i,x}$  和  $Bth\_D_{i,y}$  进行关系强度计算。设  $W_{i,j,x,y,s,t}$  表示用户  $u_i$  在日期为  $x$  这天的第  $s$  个时间片下的蓝牙感知上下文数据和用户  $u_j$  在日期为  $y$  这天的第  $t$  个时间片下的蓝牙感知上下文数据，由已有知识可知，当且仅当  $x = y$  且时间片  $s$  和  $t$  下均有数据时， $W_{i,j,x,y,s,t} = 1$ ，否则  $W_{i,j,x,y,s,t} = 0$ ，详细的计算方法同5.6中描述大同小异，这里为了节约篇幅便不再赘述。

### 5.3.4 关系强度计算结果融合

当得到用户  $u_i$  与  $u_j$  的 GPS 轨迹数据，WiFi 上下文知数据和蓝牙感知数据后，调用函数  $RS_{Tra}(u_i, u_j)$  计算得出用户  $u_i$  与  $u_j$  基于 GPS 轨迹数据的用户之间关

系强度  $rs_1$ ；通过函数  $RS_{WiFi}(u_i, u_j)$  对用户  $u_i$  与  $u_j$  的收集的智能手机的 WiFi 上下文感知数据进行处理计算得出用户之间基于 WiFi 感知数据的强度  $rs_2$ ；最后根据用户采集的蓝牙感知数据通过处理得到  $Bth\_D_i$ 、 $Bth\_D_j$  作为函数  $RS_{Bth}(u_i, u_j)$  的输入，计算得到用户之间基于蓝牙感知上下文环境数据的关系强度  $rs_3$ 。采用集成学习的思想，对每个层次计算得出的用户关系强度结果进行融合，得到最终用户  $u_i$  与  $u_j$  之间的关系强度  $rs(u_i, u_j) = \sum_{i=1}^3 rs_i$ 。

## 5.4 小结

本章就 RSMHD 各层中基于不同移动感知数据源的用户关系强度计算模块进行了细致的分析和讨论，首先对本课题研究提出的 RSMHD 用户关系强度计算框架的计算方法进行了简单概述，通过输入用户的 GPS 位置感知数据、 WiFi 和蓝牙上下文感知数据来计算用户之间的关系强度；其次，我们分别针对 RSMHD 模型中的不同层的源数据处理以及输入进行了完备的描述，为接下来计算做好数据准备；然后我们分别就如何基于用户 GPS 轨迹数据、 用户 WiFi 上下文感知数据、 用户蓝牙上下文感知数据计算用户之间的关系强度的方法进行阐述；最后我们采用集成学习的思想对每层的计算结果进行融合得到用户之间最终的关系强度结果。在下一章中我们将主要介绍本研究实验所采用的数据集，以及 RSMHD 用户关系计算模型所得出的结果等。

## 第六章 实验数据集与实验结果分析

上一章我们首先概括描述了 URSHV 层级模型；然后描述了如何准备数据作为模型的输入；最后从基于原始轨迹数据的用户关系强度计算、基于主题模型的用户关系强度计算以及结果投票三方面重点描述了基于轨迹数据的用户关系强度度量模型。本章我们将描述如何在真实数据集上对第四章提出的模型进行验证以及对实验结果进行分析。

### 6.1 实验数据集介绍

在实验的过程中，我们使用的是由自己基于文献 [51] 中的收集框架改进的一款高效、能耗低、更精确的用于收集手机上下文数据感知收集的框架 StarLog<sup>[52]</sup>。在文章开始我们将研究范围缩小至一个较小范围内进行，在此我们将校园及其周围作为研究的空间范围大小，校园的学生作为研究对象进行了手机上下文信息的采集工作，参与数据采集的学生分布情况以及其它信息如图6.1、6.2所示。每位学生使用搭载 Android 操作系统的智能手机运行 StarLog 程序采集数据，采集的数据主要有：用户手机设备信息、用户的 GPS 位置轨迹信息、WiFi 蓝牙设备使用信息、通话记录信息、社交软件使用信息、手机使用情况信息等。表6.1详细描述了收集的上下文数据的种类和类型。为了保护用户的隐私，我们对部分敏感信息进行了脱敏处理，并且用户在 StarLog 上可以自主的选择所需要收集的传感器数据。

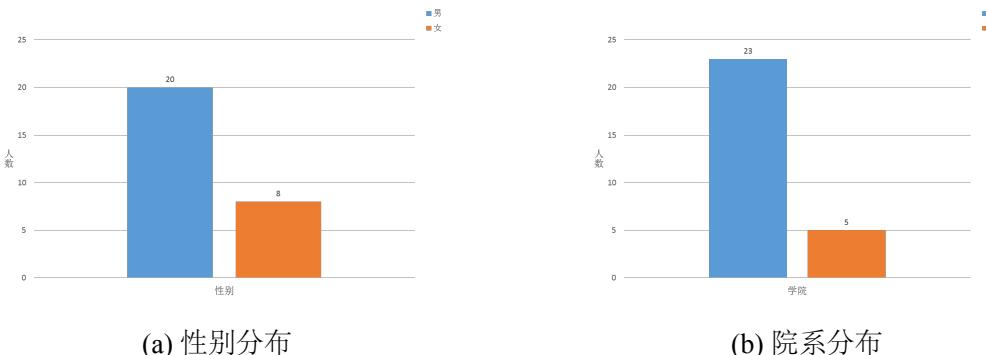


图 6.1 学生志愿者信息 1

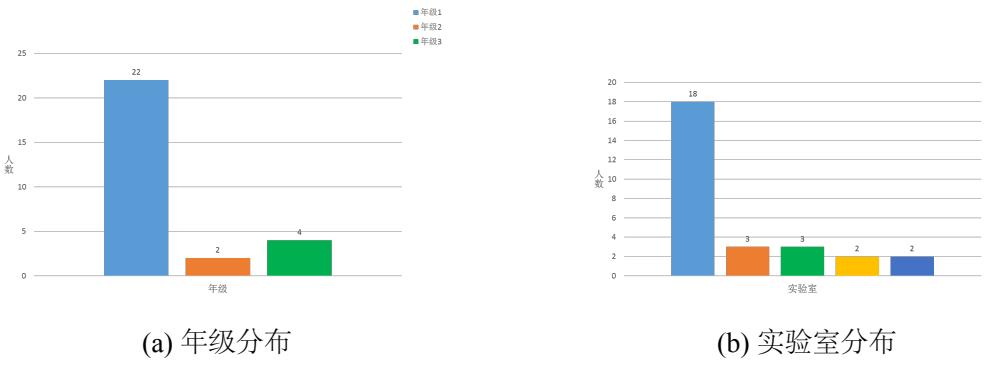


图 6.2 学生志愿者信息 2

表 6.1 StarLog 收集的上下文数据种类和类型

| 数据种类 | 上下文源   | 数据类型    |
|------|--------|---------|
| 位置   | GPS 位置 | 数值型     |
|      | WiFi   | 文本型     |
|      | 蓝牙     | 文本型     |
| 活动   | 加速度    | 数值型     |
|      | 陀螺仪    | 数值型     |
|      | 磁力计    | 数值型     |
| 用户交互 | 手机距离   | 数值型     |
|      | 应用使用   | 文本型     |
|      | 屏幕唤醒   | 文本型     |
| 社交活动 | 拍照     | 文本型     |
|      | 通话记录   | 文本型, 音频 |
|      | 短信记录   | 文本型     |
| 设备状态 | 微信, QQ | 文本型     |
|      | 充电状态   | 布尔型     |
|      | 网络接入状态 | 文本型     |
|      | 设备模式   | 文本型     |

我们通过对 28 位用户采集的数据的分析发现，在为期接近 4 周的数据采集过程中一共产生了 35 天的基于用户智能手机的上下文感知数据，图 6.3 中详细展示了志愿者在整个感知数据采集周期的数据采集贡献度。

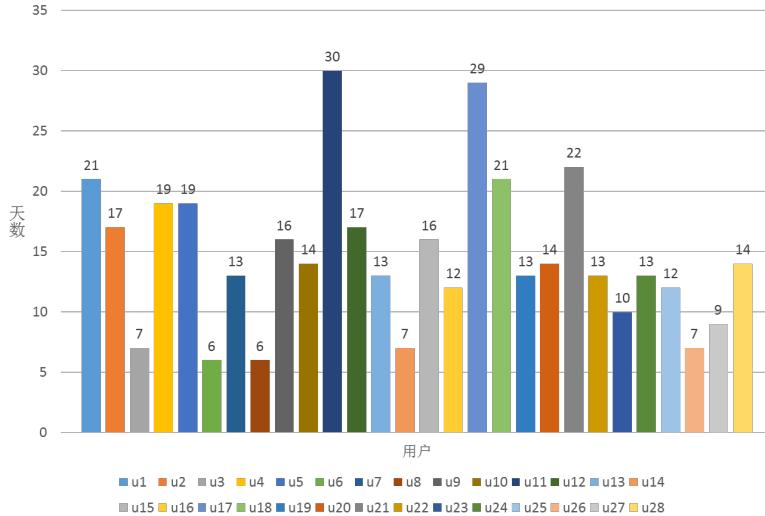


图 6.3 学生志愿者数据采集说明

从采集的学生志愿者的数据信息分析中，我们发现还存在一些不足：首先，从图 6.3 中我们可以看出并不是所有的志愿者都积极的参与到本次研究课题的数据收集环节，存在少部分志愿者没有能够坚持每天使用 StarLog 来收集自己每天的手机感知数据，这将会对用户之间的关系强度计算产生一定的影响；其次，某个用户 A 主观认为和另一个用户 B 是好朋友，而用户 B 不是和用户 A 是好朋友，因此我们提出社交关系满足对称性和反自反性的假设，即在数据集中学生 A 认为和学生 B 是好朋友，那么用学生 A 也是学生 B 的好朋友，即  $rs(u_A, u_B) = rs(u_B, u_A)$  是学生 A 和自身不参与关系强度计算。

## 6.2 实验结果验证方法

在上面的章节中我们主要描述了本研究实验所用到的智能手机上下文感知数据集，鉴于 StarLog 在采集用户手机感知数据时并没有能够直接给出用户之间的关系强度值或者顺序，因此在此之前我们需要得到一个比较符合志愿者真实情况的社会关系强度结果，并用我们提出的顺序比较法来对 RSMHD 模型的结果进行评价。

### 6.2.1 获取用户关系强度结果

根据绪论中所提及到的心理学相关的理论知识，从中证实得到在现实生活中关系亲密的两个用户会更加倾向于一起进行面对面的交流、共同进行社交活动导

致人们更加倾向于喜欢在态度、兴趣、价值观、背景和人格上和其相似的人。由此得知，在日常生活当中兴趣相投的人更容易成为朋友成为朋友。在现实生活中的推理方法通常是借助问卷调查的形式，通过被调查者填写问卷的相似性来作为用户之间的关系强度，但是这样的方法具有一定局限性和不确定性。首先，心理学中根据问卷来判断被调查者的相似性需要大量的符合心理学逻辑的调查题目，而普通问卷的内容受限且缺乏心理学逻辑；其次是志愿者在填写问卷时受到多种因素干扰无法选定最符合自己的选项。综上所述，我们为了简便可靠，在志愿者参加移动数据采集前按照意识将志愿者名单基于关系强度排序（当某志愿者并没有对剩余志愿者进行完全排序时候，我们即采用他已排序的结果作为验证结果），同时在数据采集结束后再将志愿者名单根据关系强度进行排序，得到两张用户关系强度表6.2。

从表中我们可以得出发现志愿者并没有对剩余全部人员给出关系强度排序，往往是对少部分参与者进行排序；同时，有的志愿者可能因为无法对剩余志愿者进行排序而缺少该数据，我们在实验结果验证的时候首先剔除没有给出关系强度排序的志愿者，然后对于每位志愿者我们输出与他们填写的排序长度相同的结果进行比较。

### 6.2.2 实验结果评价方法

在前面的章节介绍中我们通过志愿者根据主观意愿填写的用户关系排序表可以得到志愿者自身和其他志愿者之间的一个关系强度序列  $rsq_t$ ，同时基于 RSMHD 计算模型我们能够得到用户之间关系强度值经过排序后输出一个关系强度序列  $rsq_m$ 。因此，我们对实验结果的评价的关键问题在于如何通过比较  $rsq_t$  和  $rsq_m$  的序列相似性来评价 RSMHD 模型计算结果。

传统的最长公共子串的序列相似计算方法在本研究中因受限于用户问卷关系序列的长度影响，导致计算结果依赖于用户的初始化关系强度序列长度。我们采用基于序列熵的计算序列  $rsq_t$  和序列  $rsq_m$  的全局一致性度量方法。首先我们求出  $rsq_t$  和  $rsq_m$  的最长公共子串  $LCS(rsq_t, rsq_m) = s_1s_2 \dots s_n$ ，假设  $s_r (1 \leq r \leq n)$  与  $rsq_t$  和  $rsq_m$  中对应的用户标识是  $rsq_{tr}$  和  $rsq_{mr}$ ，则评价函数  $F(rsq_t, rsq_m)$  的计算如公式6.1所示。

$$F(rsq_t, rsq_m) = 1 - \frac{1}{\text{Min}(|rsq_t|, |rsq_m|)} \sum_{r=1}^n |H(rsq_{tr}) - H(rsq_{mr})| \quad (6.1)$$

表 6.2 志愿者之间的关系强度问卷表

| 志愿者 $u_i$ | 问卷关系排序 $\langle u_j \rangle$ (数据采集前)                                                               |
|-----------|----------------------------------------------------------------------------------------------------|
| $u_1$     | $\langle u_8, u_{10} \rangle$                                                                      |
| $u_2$     | $\langle u_{10}, u_{17} \rangle$                                                                   |
| $u_3$     | $\langle u_{20}, u_{19}, u_{13}, u_{17}, u_{10}, u_{12}, u_{11}, u_{16}, u_{21} \rangle$           |
| $u_4$     | $\langle u_{11}, u_{19} \rangle$                                                                   |
| $u_5$     | $\langle u_{12}, u_6, u_{21}, u_{11} \rangle$                                                      |
| $u_6$     | $\langle u_{12}, u_5, u_{21}, u_{11} \rangle$                                                      |
| $u_7$     | $\langle u_{10}, u_{17} \rangle$                                                                   |
| $u_8$     | $\langle u_{10}, u_{14}, u_1 \rangle$                                                              |
| $u_9$     | $\langle u_{22} \rangle$                                                                           |
| $u_{10}$  | $\langle u_{17}, u_7, u_{20}, u_{19}, u_{13}, u_{15}, u_8, u_{12}, u_{11}, u_{16}, u_{21} \rangle$ |
| $u_{11}$  | $\langle u_{12}, u_5, u_{21}, u_{16} \rangle$                                                      |
| $u_{12}$  | $\langle u_5, u_{21}, u_{16}, u_6, u_{19} \rangle$                                                 |
| $u_{13}$  | $\langle u_{20}, u_{19}, u_3, u_{17}, u_{10}, u_{16}, u_{11}, u_{12}, u_{21} \rangle$              |
| $u_{14}$  | $\langle u_{10}, u_{17} \rangle$                                                                   |
| $u_{15}$  | $\langle u_{10}, u_{17} \rangle$                                                                   |
| $u_{16}$  | $\langle u_{12}, u_5, u_{21}, u_{17} \rangle$                                                      |
| $u_{17}$  | $\langle u_{12}, u_5, u_{21}, u_{20}, u_{19}, u_{13} \rangle$                                      |
| $u_{18}$  | —                                                                                                  |
| $u_{19}$  | $\langle u_3, u_{20}, u_{13} \rangle$                                                              |
| $u_{20}$  | $\langle u_3, u_{19}, u_{13} \rangle$                                                              |
| $u_{21}$  | $\langle u_{12}, u_5, u_{21}, u_{11} \rangle$                                                      |
| $u_{22}$  | $\langle u_{13}, u_8, u_{22}, u_{23} \rangle$                                                      |
| $u_{23}$  | $\langle u_{22}, u_{13}, u_{23} \rangle$                                                           |
| $u_{24}$  | $\langle u_{23}, u_{18}, u_{22}, u_{13} \rangle$                                                   |
| $u_{25}$  | $\langle u_{26}, u_{28} \rangle$                                                                   |
| $u_{26}$  | $\langle u_{25}, u_{28} \rangle$                                                                   |
| $u_{27}$  | —                                                                                                  |
| $u_{28}$  | $\langle u_{26}, u_{25} \rangle$                                                                   |

### 6.3 实验结果与分析

实验环境采用了 Ubuntu 14.04 x64, 10GB 内存, Intel core i5 3.2GHz 4 核处理器, 使用 Python 编码实现整个计算流程。

### 6.3.1 基于 GPS 轨迹相似性计算用户之间的关系强度

通过利用 StarLog 收集的用户轨迹数据集，我们处理得到的用户 GPS 轨迹数据  $Ntra$ 、用户日常语义轨迹数据  $Ltra$ 、用户轨迹运动模式数据  $TraPF$  和  $TraSQ$ 。我们根据5.3.1中描述的方法计算用户  $u_i$  同其它志愿者  $u_j(j \in volunteers, j \neq i)$  之间的关系强度，即通过加权 DTW 算法计算用户基于日常语义轨迹数据  $Ltra$  的关系强度结果，通过快速 hash 语义变换计算用户基于日常语义轨迹数据  $Ltra$  的关系强度，通过  $TraPF$  和  $TraSQ$  计算用户基于轨迹运动模式的关系强度，最后对三维数据结果进行融合，得到用户之间基于日常轨迹数据的关系强度结果如表6.3所示。

从表6.3中选取部分志愿者之间的关系强度计算结果，得到如表6.4所示的用户关系强度矩阵。

在基于用户空间轨迹相似性、用户日常语义轨迹、用户轨迹运动模式计算用户关系强度时，我们对用户每天数据分片所采用的时间粒度均为小时，图6.4详细展示了不同的时间粒度下评价函数  $F(rsq_{10}, rsq_{12})$  的取值，可以从图中发现不同的时间粒度下得到的评价函数  $F$  值是不同的，当时间粒度等于 100 分钟左右的时候达到峰值主要原因可能  $u_{10}$  和  $u_{12}$  在数据采集期间选修的学校课程相同造成的。学校教学安排为 45 分钟每课时加上课间休息时间，若两个用户选修了相同的课程，那么基于轨迹数据计算得到的用户关系强度在时间粒度接近 100 分钟时将取得较好效果。

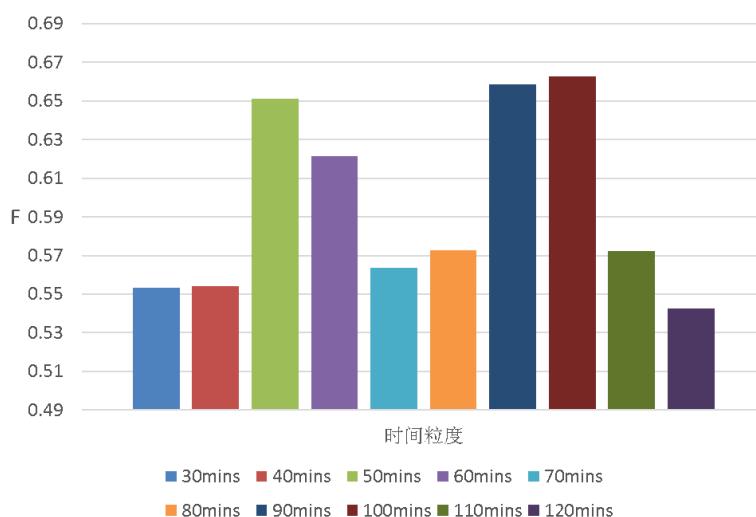


图 6.4 基于不同时间粒度的实验结果

表 6.3 基于 GPS 轨迹计算用户之间关系强度

| 志愿者 $u_i$ | 问卷关系排序 $\langle u_j \rangle$                                                                       | 基于 GPS 轨迹计算用户之间的关系强度                                                                               |
|-----------|----------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------|
| $u_1$     | $\langle u_8, u_{10} \rangle$                                                                      | $\langle u_{10}, u_8 \rangle$                                                                      |
| $u_2$     | $\langle u_{10}, u_{17} \rangle$                                                                   | $\langle u_{10}, u_{17} \rangle$                                                                   |
| $u_3$     | $\langle u_{20}, u_{19}, u_{13}, u_{17}, u_{10}, u_{12}, u_{11}, u_{16}, u_{21} \rangle$           | $\langle u_{20}, u_{17}, u_{19}, u_{13}, u_{10}, u_{16}, u_{11}, u_{12}, u_{21} \rangle$           |
| $u_4$     | $\langle u_{11}, u_{19} \rangle$                                                                   | $\langle u_{11}, u_{19} \rangle$                                                                   |
| $u_5$     | $\langle u_{12}, u_6, u_{21}, u_{11} \rangle$                                                      | $\langle u_{12}, u_6, u_{21}, u_{17} \rangle$                                                      |
| $u_6$     | $\langle u_{12}, u_5, u_{21}, u_{11} \rangle$                                                      | $\langle u_{12}, u_5, u_{27}, u_{11} \rangle$                                                      |
| $u_7$     | $\langle u_{10}, u_{17} \rangle$                                                                   | $\langle u_{10}, u_{17} \rangle$                                                                   |
| $u_8$     | $\langle u_{10}, u_{14}, u_1 \rangle$                                                              | $\langle u_{10}, u_{14}, u_1 \rangle$                                                              |
| $u_{10}$  | $\langle u_{17}, u_7, u_{20}, u_{19}, u_{13}, u_{15}, u_8, u_{12}, u_{11}, u_{16}, u_{21} \rangle$ | $\langle u_{17}, u_7, u_{20}, u_{19}, u_{13}, u_{14}, u_{21}, u_{12}, u_{11}, u_{16}, u_8 \rangle$ |
| $u_{11}$  | $\langle u_{12}, u_5, u_{21}, u_{16} \rangle$                                                      | $\langle u_5, u_{12}, u_{21}, u_{19} \rangle$                                                      |
| $u_{12}$  | $\langle u_5, u_{21}, u_{16}, u_6, u_{19} \rangle$                                                 | $\langle u_5, u_{16}, u_{21}, u_6, u_{19} \rangle$                                                 |
| $u_{13}$  | $\langle u_{20}, u_{19}, u_3, u_{17}, u_{10}, u_{16}, u_{11}, u_{12}, u_{21} \rangle$              | $\langle u_{20}, u_{19}, u_3, u_{22}, u_{24}, u_{23}, u_{11}, u_{12}, u_{21} \rangle$              |
| $u_{14}$  | $\langle u_{10}, u_{17} \rangle$                                                                   | $\langle u_{10}, u_{17} \rangle$                                                                   |
| $u_{15}$  | $\langle u_{10}, u_{17} \rangle$                                                                   | $\langle u_{10}, u_{17} \rangle$                                                                   |
| $u_{16}$  | $\langle u_{12}, u_5, u_{21}, u_{17} \rangle$                                                      | $\langle u_{12}, u_5, u_{21}, u_{17} \rangle$                                                      |
| $u_{17}$  | $\langle u_{12}, u_5, u_{21}, u_{20}, u_{19}, u_{13} \rangle$                                      | $\langle u_{12}, u_6, u_{11}, u_5, u_{19}, u_{13} \rangle$                                         |
| $u_{19}$  | $\langle u_3, u_{20}, u_{13} \rangle$                                                              | $\langle u_3, u_{20}, u_{13} \rangle$                                                              |
| $u_{20}$  | $\langle u_3, u_{19}, u_{13} \rangle$                                                              | $\langle u_3, u_{19}, u_{13} \rangle$                                                              |
| $u_{21}$  | $\langle u_{12}, u_5, u_{21}, u_{11} \rangle$                                                      | $\langle u_{12}, u_5, u_{21}, u_{11} \rangle$                                                      |
| $u_{22}$  | $\langle u_{13}, u_8, u_{22}, u_{23} \rangle$                                                      | $\langle u_{13}, u_8, u_{22}, u_{23} \rangle$                                                      |
| $u_{23}$  | $\langle u_{22}, u_{13}, u_{23} \rangle$                                                           | $\langle u_{24}, u_{13}, u_{23} \rangle$                                                           |
| $u_{24}$  | $\langle u_{23}, u_{18}, u_{22}, u_{13} \rangle$                                                   | $\langle u_{23}, u_{13}, u_{18}, u_{22} \rangle$                                                   |
| $u_{25}$  | $\langle u_{26}, u_{28} \rangle$                                                                   | $\langle u_{26}, u_{28} \rangle$                                                                   |
| $u_{26}$  | $\langle u_{25}, u_{28} \rangle$                                                                   | $\langle u_{25}, u_{28} \rangle$                                                                   |
| $u_{28}$  | $\langle u_{26}, u_{25} \rangle$                                                                   | $\langle u_{26}, u_{25} \rangle$                                                                   |

表 6.4 示例用户关系强度矩阵

|          | $u_3$    | $u_{10}$ | $u_{12}$ | $u_{13}$ | $u_{17}$ |
|----------|----------|----------|----------|----------|----------|
| $u_3$    | $\infty$ | 0.0012   | 0.0013   | 0.0088   | 0.0093   |
| $u_{10}$ | 0.0012   | $\infty$ | 0.0022   | 0.0062   | 0.00773  |
| $u_{12}$ | 0.0013   | 0.0022   | $\infty$ | 0.0129   | 0.0134   |
| $u_{13}$ | 0.0088   | 0.0062   | 0.0129   | $\infty$ | 0.00117  |
| $u_{17}$ | 0.0093   | 0.00773  | 0.0134   | 0.00117  | $\infty$ |

### 6.3.2 基于 WiFi 感知数据计算用户之间的关系强度

通过节5.2.5中输入的用户 WiFi 感知数据  $WiFi\_D$ , 根据时间片划分割预处理后得到  $Sli\_WiFi$ 。我们根据5.3.2中描述的方法, 将用户当前时刻下的 WiFi 感知上下文数据通过图结构化后, 结合公式5.6计算当前时刻下的用户 WiFi 感知环境相似性计算用户  $u_i$  同其它志愿者  $u_j(j \in volunteers, j \neq i)$  之间的关系强度, 得到用户之间基于智能手机 WiFi 感知数据的关系强度结果如表6.5所示,

表 6.5 基于 WiFi 感知数据计算用户之间关系强度

| 志愿者 $u_i$ | 问卷关系排序 $< u_j >$                                                                       | 基于 WiFi 感知数据计算用户之间的关系强度                                                                |
|-----------|----------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------|
| $u_1$     | $< u_8, u_{10} >$                                                                      | $< u_8, u_{10} >$                                                                      |
| $u_2$     | $< u_{10}, u_{17} >$                                                                   |                                                                                        |
| $u_3$     | $< u_{20}, u_{19}, u_{13}, u_{17}, u_{10}, u_{12}, u_{11}, u_{16}, u_{21} >$           | $< u_{10}, u_{17}, u_{14}, u_{13}, u_{10}, u_{16}, u_7, u_{21}, u_{19} >$              |
| $u_4$     | $< u_{11}, u_{19} >$                                                                   | -                                                                                      |
| $u_5$     | $< u_{12}, u_6, u_{21}, u_{11} >$                                                      | $< u_{12}, u_{16}, u_{21}, u_{13} >$                                                   |
| $u_6$     | $< u_{12}, u_5, u_{21}, u_{11} >$                                                      | $< u_{12}, u_5, u_{17}, u_{10} >$                                                      |
| $u_7$     | $< u_{10}, u_{17} >$                                                                   | $< u_{10}, u_{17} >$                                                                   |
| $u_8$     | $< u_{10}, u_{14}, u_1 >$                                                              | $< u_{10}, u_{14}, u_1 >$                                                              |
| $u_{10}$  | $< u_{17}, u_7, u_{20}, u_{19}, u_{13}, u_{15}, u_8, u_{12}, u_{11}, u_{16}, u_{21} >$ | $< u_{17}, u_7, u_{20}, u_{19}, u_{13}, u_{14}, u_{21}, u_{12}, u_{11}, u_{16}, u_8 >$ |
| $u_{11}$  | $< u_{12}, u_5, u_{21}, u_{16} >$                                                      | $< u_5, u_{12}, u_{19}, u_{21} >$                                                      |
| $u_{12}$  | $< u_5, u_{21}, u_{16}, u_6, u_{19} >$                                                 | $< u_5, u_{16}, u_{21}, u_6, u_{19} >$                                                 |
| $u_{13}$  | $< u_{20}, u_{19}, u_3, u_{17}, u_{10}, u_{16}, u_{11}, u_{12}, u_{21} >$              | $< u_{20}, u_{19}, u_3, u_{22}, u_{24}, u_{23}, u_{12}, u_{12}, u_{21} >$              |
| $u_{14}$  | $< u_{10}, u_{17} >$                                                                   | $< u_2, u_{10} >$                                                                      |
| $u_{15}$  | $< u_{10}, u_{17} >$                                                                   | -                                                                                      |
| $u_{16}$  | $< u_{12}, u_5, u_{21}, u_{17} >$                                                      | $< u_{12}, u_5, u_{21}, u_{17} >$                                                      |
| $u_{17}$  | $< u_{12}, u_5, u_{21}, u_{20}, u_{19}, u_{13} >$                                      | $< u_{10}, u_7, u_{12}, u_5, u_{14}, u_{13} >$                                         |
| $u_{19}$  | $< u_3, u_{20}, u_{13} >$                                                              | $< u_3, u_{20}, u_{13} >$                                                              |
| $u_{20}$  | $< u_3, u_{19}, u_{13} >$                                                              | $< u_3, u_{19}, u_{13} >$                                                              |
| $u_{21}$  | $< u_{12}, u_5, u_{21}, u_{11} >$                                                      | $< u_{12}, u_5, u_{21}, u_{11} >$                                                      |
| $u_{22}$  | $< u_{13}, u_8, u_{22}, u_{23} >$                                                      | $< u_{13}, u_8, u_{22}, u_{23} >$                                                      |
| $u_{23}$  | $< u_{22}, u_{13}, u_{23} >$                                                           | $< u_{24}, u_{13}, u_{23} >$                                                           |
| $u_{24}$  | $< u_{23}, u_{18}, u_{22}, u_{13} >$                                                   | $< u_{23}, u_{13}, u_{18}, u_{22} >$                                                   |
| $u_{25}$  | $< u_{26}, u_{28} >$                                                                   | $< u_{26}, u_{28} >$                                                                   |
| $u_{26}$  | $< u_{25}, u_{28} >$                                                                   | $< u_{25}, u_{28} >$                                                                   |
| $u_{28}$  | $< u_{26}, u_{25} >$                                                                   | $< u_{26}, u_{25} >$                                                                   |

### 6.3.3 基于蓝牙感知数据计算用户之间的关系强度

通过节5.2.6中输入的用户 WiFi 感知数据  $Bth\_D$ , 根据时间片划分分割预处理后得到  $Sli\_Bth$ 。我们根据5.3.3中描述的方法计算用户  $u_i$  同其它志愿者  $u_j(j \in volunteers, j \neq i)$  之间的关系强度, 得到用户之间基于蓝牙上下文感知数据的关系强度结果如表6.6所示。

表 6.6 基于蓝牙感知数据计算用户之间关系强度

| 志愿者 $u_i$ | 问卷关系排序 $< u_j >$                                                                       | 基于蓝牙感知数据计算用户之间的关系强度                                                                    |
|-----------|----------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------|
| $u_1$     | $< u_8, u_{10} >$                                                                      | $< u_{10}, u_8 >$                                                                      |
| $u_2$     | $< u_{10}, u_{17} >$                                                                   | $< u_{10}, u_{17} >$                                                                   |
| $u_3$     | $< u_{20}, u_{19}, u_{13}, u_{17}, u_{10}, u_{12}, u_{11}, u_{16}, u_{21} >$           | $< u_{20}, u_{17}, u_{19}, u_{13}, u_{10}, u_{16}, u_{11}, u_{12}, u_{21} >$           |
| $u_4$     | $< u_{11}, u_{19} >$                                                                   | $< u_{11}, u_{19} >$                                                                   |
| $u_5$     | $< u_{12}, u_6, u_{21}, u_{11} >$                                                      | $< u_{12}, u_6, u_{21}, u_{17} >$                                                      |
| $u_6$     | $< u_{12}, u_5, u_{21}, u_{11} >$                                                      | $< u_{12}, u_5, u_{27}, u_{11} >$                                                      |
| $u_7$     | $< u_{10}, u_{17} >$                                                                   | $< u_{10}, u_{17} >$                                                                   |
| $u_8$     | $< u_{10}, u_{14}, u_1 >$                                                              | $< u_{10}, u_{14}, u_1 >$                                                              |
| $u_{10}$  | $< u_{17}, u_7, u_{20}, u_{19}, u_{13}, u_{15}, u_8, u_{12}, u_{11}, u_{16}, u_{21} >$ | $< u_{17}, u_7, u_{20}, u_{19}, u_{13}, u_{14}, u_{21}, u_{12}, u_{11}, u_{16}, u_8 >$ |
| $u_{11}$  | $< u_{12}, u_5, u_{21}, u_{16} >$                                                      | $< u_5, u_{12}, u_{21}, u_{19} >$                                                      |
| $u_{12}$  | $< u_5, u_{21}, u_{16}, u_6, u_{19} >$                                                 | $< u_5, u_{16}, u_{21}, u_6, u_{19} >$                                                 |
| $u_{13}$  | $< u_{20}, u_{19}, u_3, u_{17}, u_{10}, u_{16}, u_{11}, u_{12}, u_{21} >$              | $< u_{20}, u_{19}, u_3, u_{22}, u_{24}, u_{23}, u_{11}, u_{12}, u_{21} >$              |
| $u_{14}$  | $< u_{10}, u_{17} >$                                                                   | $< u_{10}, u_{17} >$                                                                   |
| $u_{15}$  | $< u_{10}, u_{17} >$                                                                   | $< u_{10}, u_{17} >$                                                                   |
| $u_{16}$  | $< u_{12}, u_5, u_{21}, u_{17} >$                                                      | $< u_{12}, u_5, u_{21}, u_{17} >$                                                      |
| $u_{17}$  | $< u_{12}, u_5, u_{21}, u_{20}, u_{19}, u_{13} >$                                      | $< u_{12}, u_6, u_{11}, u_5, u_{19}, u_{13} >$                                         |
| $u_{19}$  | $< u_3, u_{20}, u_{13} >$                                                              | $< u_3, u_{20}, u_{13} >$                                                              |
| $u_{20}$  | $< u_3, u_{19}, u_{13} >$                                                              | $< u_3, u_{19}, u_{13} >$                                                              |
| $u_{21}$  | $< u_{12}, u_5, u_{21}, u_{11} >$                                                      | $< u_{12}, u_5, u_{21}, u_{11} >$                                                      |
| $u_{22}$  | $< u_{13}, u_8, u_{22}, u_{23} >$                                                      | $< u_{13}, u_8, u_{22}, u_{23} >$                                                      |
| $u_{23}$  | $< u_{22}, u_{13}, u_{23} >$                                                           | $< u_{24}, u_{13}, u_{23} >$                                                           |
| $u_{24}$  | $< u_{23}, u_{18}, u_{22}, u_{13} >$                                                   | $< u_{23}, u_{13}, u_{18}, u_{22} >$                                                   |
| $u_{25}$  | $< u_{26}, u_{28} >$                                                                   | $< u_{26}, u_{28} >$                                                                   |
| $u_{26}$  | $< u_{25}, u_{28} >$                                                                   | $< u_{25}, u_{28} >$                                                                   |
| $u_{28}$  | $< u_{26}, u_{25} >$                                                                   | $< u_{26}, u_{25} >$                                                                   |

### 6.3.4 基于 RSMHD 计算用户之间的关系强度

## 6.4 小结

本章我们主要描述了验证我们模型使用的数据集，如何使用数据集构造真实结果，设计适用本课题问题的评估方法以及展示了实验演过并进行了深入分析。

## 第七章 结束语

### 7.1 工作总结

随着硬件的迅速发展，智能手机得以嵌入更多的传感器，且拥有更大的内存，更快的处理器，使得我们可以利用智能手机研究好多以前无法解决的问题。通常情况下，手机都会随身携带，从而手机可以基本完整的记录我们每天的生活轨迹，而生活轨迹又能很大程度上反映人和人之间在真实世界的交互，进而使得使用轨迹数据来度量人和人之间的关系强度成为可能。以前的工作只考虑真实世界人和人的交互次数，实际上，使用手机采集的轨迹数据我们能够得到更多的信息，比如用户基于轨迹的行为模式。因此，我们尝试从更多方面去考虑轨迹数据和用户之间关系强度的关系。本课题针对如何度量日常生活中人们之间的关系强度问题展开研究，提出了一个既可以对 GPS 数据进行处理又可以对基站数据进行处理，从日常轨迹、物理位置以及语义位置三个层次度量人们之间关系强度的层级模型 URSHV。概括起来，主要研究内容和贡献如下：

(1) GPS 数据相较于基站数据更杂乱，因此需要许多额外的处理，我们首先研究了对 GPS 数据的一些额外的处理。原始 GPS 数据采样得到结果有较大误差，且采集数据存在大量用户日常活动在路上行进的点，而我们课题只需要考虑用户停留在宿舍、实验室等语义位置相关的点，故需要对 GPS 数据进行降噪并剔除路上的点。在此基础上，我们需要通过一些方法来发现与 GPS 原始数据对应的如宿舍、实验室等语义位置，进而标记每个语义位置对应的语义标签。在本课题中，我们通过实验各种滤波算法，发现分段卡尔曼滤波具有更好的降噪效果；通过对采集的 GPS 数据进行进一步分析，我们发现路上点的密度远小于用户处于语义位置时的点的密度，因此我们采用基于密度的异常点剔除方法，且该方法可以自动学习参数；当前该领域用来发现语义位置的聚类算法存在一些问题，比如需要预先知道类别的个数或者对参数比较敏感，我们采用最新提出的一个基于密度的聚类算法来发现语义位置，该方法对参数更鲁棒，且不需要预先知道类别个数；在得到语义位置的基础上，我们需要通过一些方法标记语义位置对应的语义标签，目前常用的方法是人工手动标注，我们发现可以通过反地理编码，语义标签推断以及输入自动补全来减少用户和语义标签标注系统的交互。

(2) 我们使用原始轨迹数据的相似度，基于语义位置的用户行为模式的相似度以及基于语义标签的用户行为模式的相似度三方面来度量用户之间的关系强度。如何度量原始轨迹数据的相似度、如何度量用户模式之间的相似度以及如何对三层结果进行融合就是本课题最关键的问题。首先，在计算原始轨迹数据的相

似度时，我们发现使用编辑距离计算得到的相似度效果不是很理想，而 DTW 距离更倾向于序列长度较长的序列。因此，我们对 DTW 计算得到的距离使用三种方法归一化。并且我们发现用户每天活动的多样性不同使得该天轨迹数据的相似度对最终的相似度贡献不同，因此，我们使用用户每天轨迹序列的熵值对用户每天的相似度加权。其次，在计算用户行为模式时，我们发现 LDA 主题模型可以很好的用来发现用户基于轨迹的行为模式，且该模型的推断方法能够帮助我们很好的度量行为模式之间的相似度。最后，在得到三个层次的用户关系强度计算结果后，我们使用集成学习的思想对三个结果进行投票，并且以投票结果作为最终的关系强度。

(3) 以前的工作基于仿真数据集进行验证，真实数据集还是存在一些问题，如何对真实数据集进行处理，构造真实结果，以及如何针对我们的问题提供相应的评估方法以及模型中的各个参数对实验结果究竟有什么影响也是本课题急需解决掉一个重要问题。我们使用第五章第一节提到的数据集，对其朋友关系进行处理，使得该关系满足反自反和对称，用朋友之间调查问卷的相似度作为用户之间真实的关系强度。我们对用户关系强度的度量结果其实是该用户全部好友按与该用户关系强度亲密程度降序排列对应的好友序列，因此我们针对有序序列提出基于逆序对数的一致性评分评估标准来评价我们模型的实验结果。最后我们通过实验证明了各个参数对模型实验结果的影响，并且对结果进行了深入分析。

## 7.2 工作展望

论文针对基于轨迹数据的用户关系强度度量问题展开研究，在对相关技术研究基础上，提出了一个既可以对 GPS 数据进行处理又可以对基站数据进行处理，从日常轨迹、物理位置以及语义位置三个层次度量人们之间关系强度的层级模型 URSHV。虽然取得了一定的结果，但是仍然存在许多问题需要进一步研究和完善。现将这些问题总结如下：

(1) 基于 DTW 的关系强度度量方法虽然能得到一个比较好的度量结果，但是该算法具有比较大的时间复杂度。下一步工作尝试对数据进行一定的优化，使得该算法计算时消耗更少的时间。

(2) 我们只使用了原始轨迹数据的相似度和基于轨迹的行为模式的相似度两方面来度量用户之间的关系强度，根据轨迹数据，我们其实能够得到更多的信息。下一步工作尝试深入理解轨迹数据以及考虑从更多的方面来使用轨迹数据，使得我们对关系强度的度量更加全面。

(3) 虽然轨迹数据和我们在真实世界的交互密切相关。但是，通话记录，短信，蓝牙交互，社交网络交互，这些信息同样能够反映用户之间的关系强度。下

一步工作尝试采集更多的手机传感器数据，分别研究如何基于单个传感器数据度量用户关系强度以及如何综合使用这些传感器数据度量用户之间的关系强度。



## 致 谢

在本文完成和硕士生涯结束之际，我谨向所有给予我指导、关心、支持和帮助的老师、领导、同学和亲人致以衷心的感谢！

衷心感谢我的导师史殿习教授！感谢史老师对我的无私关怀、精心指导和严格要求。在两年多的硕士学习生活中，史老师在百般繁忙之际仍不忘抽出时间关心我的课题进展，找我聊人生、聊理想。不仅对我的课题提出很多很宝贵的建设性意见，而且对我将来人生的发展方向以及职业规划也提出很多至关重要的建议。我本身不是一个很勤奋的人，史老师也未严格要求每天呆在实验室的时间，但是每次有事找史老师时，史老师都在办公室看论文，史老师对移动感知领域论文了解的广度和深度，我望尘莫及，每次看到老师刻苦的身影，暗暗下定决心，一定要多看论文。史老师严谨的治学精神和精益求精的工作作风在不断的鞭笞着我，让我不断前行。在课题方面，我没有做到史老师希望的那么优秀，辜负了史老师对我的期望，在以后的日子里，我会更加努力，做更好的工作，做更好的自己，在此谨向他表示最衷心的感谢和最诚挚的敬意。

衷心感谢尹刚老师、丁博老师、刘慧老师，感谢你们在我课题的开题阶段给我的宝贵的建议和支持，使我更加明确了研究方向和思路，且为后续课题研究进展提供了很多现在看来确实很有远见的指导，衷心感谢窦勇老师和王晓东老师在预审阶段对我论文撰写提出的珍贵意见，使得该论文能够顺利完成。同时也感谢你们在生活上和学习上对我的帮助。

感谢何炫辰学长、李永谋学长、丁涛杰学长、陈富霞学姐和金星学姐，感谢你们在生活和学习上对我的帮助，和何炫辰学长的每次聊天都能使我烦躁的内心平静好多，每次学术上的问题李永谋学长都不厌其烦的帮我解决，陈富霞学姐给了我很多生活上的经验，受用至今。感谢同门好友吴渊、李寒、陈茜、周荣、谭杰夫、樊泽栋。感谢吴渊一直陪我去五楼实验室打乒乓球，经常陪我去东门吃夜宵；感谢李寒经常带我出去玩，看外面的世界；感谢陈茜帮我报账，提供零食当夜宵；感谢周荣给我平时无聊的生活带来乐趣。非常感谢各位同门好友在学习和生活上给我提供的帮助，因为你们使得平时波澜不惊的生活充满了色彩，非常庆幸、非常感谢！感谢王峰、赵邦辉、陈晓鹏、莫晓赟、李中秋、刘帆、童哲航、颜丙政、魏菁、成瑶瑶学弟学妹，和你们一起讨论学术问题，交流人生经验，我自己也受益匪浅。你们都是富有才华的研究人才，祝你们人生之路丰富多彩。感

谢同一个实验室的张飞、王东升，刘冰珣、古崇明经常打扰你们找你们聊天，你们没有怨言还给了我很多学术和生活的建议。

感谢朱涛政委、孙友佳政委和郑永辉政委。我们是科大第一届地方生，很多规矩都不懂，感谢你们在这两年半中给予我的帮助。感谢室友许名广、伍名、谢飞，感谢对面宿舍孙洪雷、吴平杰、吴茂永，在与你们一起度过的快乐的两年半生活中，感谢你们在学习和生活中给予我的帮助。

感谢学院八队的所有同学，与你们在一起的两年半时光是我人生最宝贵的一笔财富。感谢两年以来悉心指导我的每一位老师和前辈，你们的辛勤汗水让我在成长道路上跨步前进。

感谢 NudtPaper，它的存在让我的论文写作轻松自在了许多，让我的论文格式规整漂亮了许多。

最后，深深感谢生我养我的父母，你们的支持是我前进的最大动力，愿你们健康长寿！

## 参考文献

- [1] Starner T. Human-powered wearable computing [J]. IBM systems Journal. 1996, 35 (3.4): 618–629.
- [2] Cheng N, Mohapatra P, Cunche M, et al. Inferring user relationship from hidden information in wlans [C]. In MILCOM 2012-2012 IEEE Military Communications Conference. 2012: 1–6.
- [3] Eagle N, Pentland A. Reality mining: sensing complex social systems [J]. Personal and ubiquitous computing. 2006, 10 (4): 255–268.
- [4] Min J-K, Wiese J, Hong J I, et al. Mining smartphone data to classify life-facets of social relationships [C]. In Proceedings of the 2013 conference on Computer supported cooperative work. 2013: 285–294.
- [5] Pentland A, Eagle N, Lazer D. Inferring social network structure using mobile phone data [J]. Proceedings of the National Academy of Sciences (PNAS). 2009, 106 (36): 15274–15278.
- [6] Zhang D, Guo B, Yu Z. The emergence of social and community intelligence [J]. Computer. 2011, 44 (7): 21–28.
- [7] Miluzzo E, Lane N D, Eisenman S B, et al. CenceMe—injecting sensing presence into social networking applications [M] // Miluzzo E, Lane N D, Eisenman S B, et al. Smart Sensing and Context. Springer, 2007: 2007: 1–28.
- [8] Wang R, Chen F, Chen Z, et al. StudentLife: assessing mental health, academic performance and behavioral trends of college students using smartphones [C]. In Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing. 2014: 3–14.
- [9] Wang R, Harari G, Hao P, et al. SmartGPA: how smartphones can assess and predict academic performance of college students [C]. In Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing. 2015: 295–306.
- [10] Granovetter M S. The strength of weak ties [J]. American journal of sociology. 1973: 1360–1380.
- [11] Haythornthwaite C. Strong, weak, and latent ties and the impact of new media [J]. The information society. 2002, 18 (5): 385–401.

- [12] Gustafson S, Moitra A. Extracting and Measuring Relationship Strength in Social Networks [J]. 2012.
- [13] Khadangi E, Zarean A, Bagheri A, et al. Measuring relationship strength in online social networks based on users' activities and profile information [C]. In Computer and Knowledge Engineering (ICCKE), 2013 3th International eConference on. 2013: 461–465.
- [14] Zhao X, Yuan J, Li G, et al. Relationship strength estimation for online social networks with the study on Facebook [J]. Neurocomputing. 2012, 95: 89–97.
- [15] Zillmann D, Bryant J. Selective exposure to communication [M]. Routledge, 2013.
- [16] Zajonc R B. Attitudinal effects of mere exposure. [J]. Journal of personality and social psychology. 1968, 9 (2p2): 1.
- [17] Zillmann D. Mood management in the context of selective exposure theory [J]. Annals of the International Communication Association. 2000, 23 (1): 103–123.
- [18] Zheng Y, Zhang L, Ma Z, et al. Recommending friends and locations based on individual location history [J]. ACM Transactions on the Web (TWEB). 2011, 5 (1): 5.
- [19] Lee J-G, Han J, Whang K-Y. Trajectory clustering: a partition-and-group framework [C]. In Proceedings of the 2007 ACM SIGMOD international conference on Management of data. 2007: 593–604.
- [20] Li Q, Zheng Y, Xie X, et al. Mining user similarity based on location history [C]. In Proceedings of the 16th ACM SIGSPATIAL international conference on Advances in geographic information systems. 2008: 34.
- [21] Lu E H-C, Tseng V S, Philip S Y. Mining cluster-based temporal mobile sequential patterns in location-based service environments [J]. IEEE Transactions on Knowledge and Data Engineering. 2011, 23 (6): 914–927.
- [22] Burt R S. Structural holes: The social structure of competition [M]. Harvard university press, 2009.
- [23] Carruthers P. The illusion of conscious will [J]. Synthese. 2007, 159 (2): 197–213.
- [24] Burrows R, Nettleton S, Pleace N, et al. Virtual community care? Social policy and the emergence of computer mediated social support [J]. Information, Communication & Society. 2000, 3 (1): 95–121.
- [25] Coles A J, Wing M G, Molyneux P, et al. Monoclonal antibody treatment exposes three mechanisms underlying the clinical course of multiple sclerosis [J]. Annals of neurology. 1999, 46 (3): 296–304.

- [26] Petróczi A, Nepusz T, Bazsó F. Measuring tie-strength in virtual social networks [J]. *Connections*. 2007, 27 (2): 39–52.
- [27] Hsu W-j, Dutta D, Helmy A. Mining behavioral groups in large wireless LANs [C]. In Proceedings of the 13th annual ACM international conference on Mobile computing and networking. 2007: 338–341.
- [28] Eagle N, Pentland A S, Lazer D. Inferring friendship network structure by using mobile phone data [J]. *Proceedings of the national academy of sciences*. 2009, 106 (36): 15274–15278.
- [29] Zheng J, Ni L M. An unsupervised learning approach to social circles detection in ego bluetooth proximity network [C]. In Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing. 2013: 721–724.
- [30] Do T M T, Gatica-Perez D. Groupus: Smartphone proximity data and human interaction type mining [C]. In 2011 15th Annual International Symposium on Wearable Computers. 2011: 21–28.
- [31] Mtibaa A, Chaintreau A, LeBrun J, et al. Are you moved by your social network application? [C]. In Proceedings of the first workshop on Online social networks. 2008: 67–72.
- [32] Ma C, Cao J, Yang L, et al. Effective social relationship measurement based on user trajectory analysis [J]. *Journal of Ambient Intelligence and Humanized Computing*. 2014, 5 (1): 39–50.
- [33] Zheng Y. Trajectory data mining: an overview [J]. *ACM Transactions on Intelligent Systems and Technology (TIST)*. 2015, 6 (3): 29.
- [34] Itakura F, Umezaki T. Distance measure for speech recognition based on the smoothed group delay spectrum [C]. In *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP'87*. 1987: 1257–1260.
- [35] Gonzalez M C, Hidalgo C A, Barabasi A-L. Understanding individual human mobility patterns [J]. *Nature*. 2008, 453 (7196): 779–782.
- [36] Wegmann H. Image orientation by combined (A) AT with GPS and IMU [J]. *INTERNATIONAL ARCHIVES OF PHOTOGRAMMETRY REMOTE SENSING AND SPATIAL INFORMATION SCIENCES*. 2002, 34 (1): 278–283.
- [37] Kalman R E. A new approach to linear filtering and prediction problems [J]. *Journal of Fluids Engineering*. 1960, 82 (1): 35–45.
- [38] Zheng Y, Zhou X. Computing with spatial trajectories [M]. Springer Science & Business Media, 2011.

- 
- [39] MacQueen J, et al. Some methods for classification and analysis of multivariate observations [C]. In Proceedings of the fifth Berkeley symposium on mathematical statistics and probability. 1967: 281–297.
  - [40] Ester M, Kriegel H-P, Sander J, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. [C]. In Kdd. 1996: 226–231.
  - [41] Zhou C, Frankowski D, Ludford P, et al. Discovering personally meaningful places: An interactive clustering approach [J]. ACM Transactions on Information Systems (TOIS). 2007, 25 (3): 12.
  - [42] Rodriguez A, Laio A. Clustering by fast search and find of density peaks [J]. Science. 2014, 344 (6191): 1492–1496.
  - [43] Liu H, Schneider M. Similarity measurement of moving object trajectories [C]. In Proceedings of the Third ACM SIGSPATIAL International Workshop on GeoStreaming. 2012: 19–22.
  - [44] Ratanamahatana C A, Keogh E. Everything you know about dynamic time warping is wrong [C]. In Third Workshop on Mining Temporal and Sequential Data. 2004.
  - [45] Singelis T M. The measurement of independent and interdependent self-construals [J]. Personality and Social Psychology Bulletin. 1994, 20 (5): 580–591.
  - [46] Farrahi K, Gatica-Perez D. What did you do today?: discovering daily routines from large-scale mobile data [C]. In Proceedings of the 16th ACM international conference on Multimedia. 2008: 849–852.
  - [47] 杨若松. 基于轨迹数据的用户关系强度度量方法研究 [D]. [S. l.]: 万方数据资源系统, 2016.
  - [48] Mikolov T, Chen K, Corrado G, et al. Efficient estimation of word representations in vector space [J]. arXiv preprint arXiv:1301.3781. 2013.
  - [49] Blei D M, Ng A Y, Jordan M I. Latent dirichlet allocation [J]. the Journal of machine Learning research. 2003, 3: 993–1022.
  - [50] Manku G S, Jain A, Das Sarma A. Detecting near-duplicates for web crawling [C]. In Proceedings of the 16th international conference on World Wide Web. 2007: 141–150.
  - [51] Rawassizadeh R, Tomitsch M, Wac K, et al. UbiqLog: a generic mobile phone-based life-log framework [J]. Personal and ubiquitous computing. 2013, 17 (4): 621–637.
-

- [52] Xiaoyun Mo R Y H L Z T F W, Dianxi Shi. A Framework of Mobile Sensing Data Collection and Behavior Analysis in an Energy-efficient Way [J]. The 8th IEEE International Conference on Social Computing and Networking(SocialCom 2015). 2015.



## 作者在学期间取得的学术成果

### 发表的学术论文

- [1] Ruosong Yang, Dianxi Shi. SASLL: A System Annotating Semantic Label of Location. The 7th International Symposium on UbiCom Frontiers Innovative Research, Systems and Technologies.(EI 检索)
- [2] Dianxi Shi, Ruosong Yang. Measuring User Relationship Strength Using a Model Based on Hierarchical Voting. 2016 PerCom. (已投稿)

