清 华 大 学

综 合 论 文 训 练

题目：基于 Wi-Fi 定位的伪基站大数据分析

系　　别：数学科学系

专　　业：数学与应用数学

姓　　名：汪芃

指导教师：李振华 助理教授

刘思齐 教授

2016 年 6 月 3 日

关于学位论文使用授权的说明

本人完全了解清华大学有关保留、使用学位论文的规定，即：学校有权保留学位论文的复印件，允许该论文被查阅和借阅；学校可以公布该论文的全部或部分内容，可以采用影印、缩印或其他复制手段保存该论文。

(涉密的学位论文在解密后应遵守此规定)

签　名：_____导师签名：_____日　期：_____

中文摘要

近年来伪基站快速增长，影响手机用户正常使用的同时更滋生违法犯罪行为。传统的伪基站检测方法需使用专业设备、现场检测，或只能用于个人使用、准确率不高。本文用百度提供的伪基站数据集，提出一种基于聚类的大数据中的伪基站检测方法，能够比简单判定规则获得更可靠的结果，并自动发现伪基站密集区域，为执法部门提供线索。

本文解决的核心问题是从噪声很大的无标签数据中，筛选出其中的异常记录。为此，先描述数据并用简单规则初步进行判断，指出这种方法的不足。因此，选择对每条记录的特征，利用改进的 k-means 算法进行聚类，又通过带初始种子的不完整先验信息半监督聚类，用对伪基站的已知信息指导聚类过程。文中回顾了使用的算法，并描述应用于本问题所需的技术细节。

利用数据的一小部分样本，编写程序进行实验。从聚类的内部指标判断，调整参数到较优的状态，得出数据中的疑似伪基站记录。如果将聚类结果用来为更大的数据集进行分类，或在全部数据上利用本文提出的方法处理，可以得到更准确的疑似伪基站集合。可以将观测到这些伪基站的位置在地理位置上进行基于密度的 DBSCAN 聚类，从而得出伪基站频繁出现的地区，为打击伪基站提供依据。

关键词：伪基站；聚类分析；大数据；异常检测；k-means

ABSTRACT

Fake base-stations (FBSs) have been increasing rapidly in recent years. They interfere with the normal operation of mobile phones and give rise to criminal activities. Traditional methods of FBS detection either require on-site process with specialized equipment, or suit for personal use only and suffer from inaccuracy. This research makes use of the FBS big data collected by Baidu, and proposes a FBS detection method in big data based on clustering. It provides better reliability than simple rules and is able to automatically detect areas where FBSs concentrate, thus contributing to the action of the authorities.

The core issue this article addresses is outlier detection in unlabeled data with large noise. Initially, the data set is described and some simple rules are used to select abnormal records. Weakness of this method is discussed. Therefore, we apply an improved k-means clustering algorithm to the features of data. Our current information of FBS gets imposed in the progress through semi-supervised clustering with incomplete prior knowledge and seeding. The article reviews existing methods and includes technical details that are necessary in this specific problem.

Programs are written to conduct experiments on a sample of the big data. With the help of interior indicators, we tune parameters to optimum and they indicate the success of clustering. Suspected FBSs are chosen from the data set. More accurate and larger sets of FBSs can be found if the results of clustering are used to classify larger data sets, or the whole process may be repeated with complete data. Finally, the observed position of these FBSs can be clustered geographically with DBSCAN and the authorities could take legal actions in areas of FBS concentration.

Keywords： fake base-station; cluster analysis; big data; outlier detection; k-means

目 录

第 1 章 引言

1.1 课题背景

近年来，伪基站一直是移动通信安全领域中的一大问题以及研究热点。伪基站利用 GSM 网络通信协议的漏洞，冒充运营商的正常基站，通过其较强的信号，强制用户与其连接，然后可以伪造任意电话号码向用户发送垃圾短信。这不仅给了不法分子实行诈骗等犯罪的机会，伪基站的存在还会干扰附近用户的正常通信，广大用户深受其害。因此，国家明令禁止生产、销售、使用伪基站设备[22]。尽管如此，伪基站的数量在近几年仍较大且快速上升[2]。根据百度发布的《2015 互联网安全白皮书》[23]，来自伪基站的短信数从 2014 年的 11.9 亿条上升为 2015 年的 23.2 亿条。这对发现和查处伪基站提出了更急迫的要求。

然而现有的伪基站检测方法并不令人十分满意。目前管理部门常用的方式是，在疑似有伪基站出现的位置，使用专用设备排查和追踪伪基站设备[24]。然而不法分子常将设备隐藏在机动车当中，并且移动作案，时效性差，难免有“守株待兔”的感觉。另一种措施是利用智能手机上的软件，联网查询手机连接到基站的编号所对应的地理位置，再与手机本身的定位相对比，判断基站是否异常[4][5]。但是，正如本文下一章所分析的情况，因为基站定位数据库本身存在大量的错误和缺失数据，这种方法的正确性难以保证，会出现误报和漏报。此外，这种方法也只能对于个人用户过滤伪基站短信使用，无法对有关部门的监督起到帮助作用。

为了接下来介绍更具体的技术内容，先简单介绍伪基站的工作原理。因为手机没有验证 GSM 基站的机制，它会主动连接任何可用的基站并信任其传输的数据。首先，伪基站用较强的信号诱使手机发生小区重选：每个基站有一个编号，格式为 MCC（移动国家代码，中国为 460）|MNC（移动网络号码，例如中国移动为 00、中国联通为 01）|LAC（位置区域码）|CID（基站编号）。伪基站用同样的 MNC，而随机生成 LAC 和 CID，便可伪装成另一个小区的合法基站。手机会根据扫描到附近几个小区和基站的信号强度，决定连接到哪个基站。简单地说，如果另一个小区的信号明显强于当前小区，则转而连接前者。然后，伪基站将会向手机发送垃圾短信，或者根据获取到的 SIM 卡信息进行其他不法活动。最后，伪基站再次更改自己的 LAC，并拒绝手机此后的连接，从而迫使手机重新连接原来

的真实基站。在此过程中，不仅伪基站可伪造通信内容，用户的手机也无法进行正常通信。虽然此漏洞只存在于 2G 网络，但现今的 3G/4G 手机也会受到影响[28]。

1.2 本课题研究的问题

我们获得了一批可能包含伪基站的通信记录，希望设计合理的方法，从中筛选出来自伪基站的信息条目。具体来说，百度手机卫士[1]会在用户收到来自非通讯录联系人的短信时，记录手机附近的 Wi-Fi AP（无线网络访问点）的信息，以及最近连接的基站信息。这样的数据上报后汇集起来，构成本文研究所使用的伪基站大数据集。每天产生的记录量在几十万条左右。

数据收集机制的设计思路如下：因为 GPS（全球定位系统）的一些限制，记录手机扫描到附近 Wi-Fi AP 的信息，可以用它们的编号来查询它们所处的位置，从而刻画手机此时的位置。同样，可以查询基站编号所对应的位置。如果二者差距过大，则判定该基站异常。

但是，第 2 章中的初步尝试表明，这种机制的效果非常不好。一方面，Wi-Fi 定位的误差可能很大[6][7]，并且基站定位数据库也存在众多错误和数据缺失；另一方面，伪基站随机生成编号的原理决定了大部分伪基站使用的编号无法查询到位置，也就不可能用来计算距离。

因此，本论文研究的主要问题是：怎样从具有很大噪声的大数据中，检测出伪基站产生的异常记录？

第 2 章中，在简单描述大数据样本的特征后，我们试图用一些人工设置的简单规则，来筛选出异常记录。但实验表明这样处理的效果不能令人满意，又没有评价检测效果的指标（因为记录中大部分来自正常基站发送的陌生号码短信，我们不知道哪些记录对应伪基站，即数据没有标签）。所以，在重新分析需求后，选择通过利用记录本身的特征，将记录进行聚类来寻找异常的簇。第 3 章中，将会对所需要的聚类算法进行回顾，即基于原型的 k-means 算法以及基于密度的 DBSCAN 算法。为了将目前我们对伪基站的认识加入此过程，我们指出这是一个不完全先验信息的半监督聚类问题，并且解释在这种情况下，利用初始种子提高聚类质量并实现部分分类的算法。在这之后，第 4 章中可以描述我们设计的大数据中的伪基站检测方法。这里将解释选取的特征以及产生初始种子的方法等技术细节。利用现有知识，还用人工设计的规则筛选出较少而确定的一些正例和反例，有多种用途。在数据集上进行实验，获得了较好的效果。第 5 章总结全文。

1.3 已有研究的情况

利用聚类等方法在大数据中检测伪基站的问题，尽我所知，这是文献中第一次的研究。此前对于伪基站，通常使用专门设备来实地检测其信号，实际当中执法部门多用这种方法[24]；也有利用这种原理的研究论文[25][26]。另一种方法是查询基站编号对应的位置，与手机当前位置进行比对，类似的原理出现在众多文章当中[4][9]，但都只用于个人用户对于自己通信安全的保护。有一些研究还需要对通信协议做额外更改[27]。此外，也有少数运营商，尝试通过发现其基站下用户重连的异常，来大致确定伪基站所在的位置，但这方面的研究还不完善，在现实中也鲜有成功使用的报道。

本文用到多种数据挖掘中的聚类算法。基于原型的无监督聚类中，最为经典的是 k-means 算法[13]。文献[14][15]证明它处理的问题即使在简单情况下也是 NP 难的，因此随后有大量研究来改进其聚类效果。对于簇结构有先验知识的情况下，有人提出基于初始种子的半监督聚类算法 Seeded-KMeans 和 Constrained-KMeans[11]。在先验信息不完整的情况下，FS-KMeans[12]能够较好地处理，而本文对于记录特征的聚类所使用的就是它的一个修改版本。至于基于密度的无监督聚类，DBSCAN[17]是广泛使用的算法，最近的研究文章对其性能有一些深入研究[20]。

本文还考虑了 Wi-Fi 定位的精度问题。这方面长久以来有充分的研究，例如有文章对各种定位方式的精度进行比较[31]；但很多都是针对某种特殊情况、需要额外处理等[29][30]。当然，大部分情况下，Wi-Fi（辅助）定位都是一个动态的、实时的过程，因此可以有更多改进空间。而本研究使用的数据当中，只能获得某一个时刻的 Wi-Fi AP 信息，所以其改进范围有限。有人细致研究了各种环境特征对于多种 Wi-Fi 定位计算策略的影响[6]，本文将其中一些思想应用在若干技术处理当中。

第 2 章 伪基站数据集

我们先对伪基站数据集的来源和内容进行描述，然后用人工设定的简单判断规则对其进行初步筛选。初步分析结果表明，虽然这样能检测到数据中的一部分异常，但还需要用其他技术来提高筛选的准确度。

2.1 数据来源

本课题使用的数据为百度手机卫士[1]在众多移动电话用户的设备上收集所得。对于同意收集相关信息的移动设备，每当收到一条短消息，且该短消息的发件人未在手机通讯录中保存时，就认为这是一条可疑短消息，并且记录相关信息。这些信息包括发件人号码是否为“权威号”、短消息内容是否判断为垃圾消息或诈骗消息，还包括最近连接的基站信息和手机附近的 Wi-Fi 接入点记录等。这些信息在适当时候自动向百度公司上报，后者将其汇集成“伪基站数据集”，以便后期从大数据中挖掘伪基站相关内容。

2.2 数据内容

图 2.1 展示了一条典型的原始数据。为排版美观，删除了部分相似的内容，并加以缩进。数据为 JSON 格式。每条记录当中各字段的含义在表 2.1 当中解释。


```
{
  "line": 23,
  "isAuthority": false,
  "function": "good",
  "wifi": { "tag": true, "lat": 31.27, "lon": 121.51 },
  "wf": [
    { "wifi": "d0:0e:d9:97:c1:fb", "tag": false },
    { "wifi": "8c:21:0a:a5:2c:30", "tag": true, "lat": 31.27, "lon": 121.51, "distance": 45.48 }
  ],
  "base": [
    { "id": "460|00|6305|58241", "tag": true, "lat": 31.280343, "lon": 121.52, "radius": 1500,
```

"cellStrength": -73, "time": 1452863377190, "legal": true },
{"id": "460|00|6305|58289", "tag": true, "lat": 31.280343, "lon": 121.52, "radius": 1500,
"cellStrength": -72, "time": 1452863386117, "legal": true }
]
}

图 2.1 一条原始数据的样例（有部分删节）

表 2.1 数据字段含义

名称	对象中的名称	类型	解释	示例值
line	-	数值	原始记录中的编号	23
isAuthority	-	布尔	是否来自权威号码	false
function	-	字符串	内容检测结果，good：正常 spam：垃圾 cheat：诈骗	good
wifi		对象	百度给出的 Wi-Fi 聚类结果	见下
	tag	数值	是否得到了聚类结果	true
	lat	数值	纬度	31.27
	lon	数值	经度	121.51
wf		数组	此时所有 Wi-Fi 接入点	见下
	Wi-Fi	字符串	Wi-Fi AP 的 MAC 地址	8c:21:0a:a5:2c:30
	tag	布尔	是否查询到位置	true
	lat	数值	纬度	31.27
	lon	数值	经度	121.51
	distance	数值	到百度的聚类中心的距离	45.48
base		数组	最近连接的三个基站	见下
	id	字符串	基站编号	460|00|6305|58241
	tag	布尔	是否查询到位置	true
	lat	数值	纬度	31.28
	lon	数值	经度	121.51
	radius	数值	基站覆盖范围，单位米	1500
	cellStrength	数值	信号强度，单位 dBm	-73
	time	数值	连接到此基站的时间戳	1452863377190
	legal	布尔	id 是否符合格式要求	true

如表中所示，每条记录可以分为三部分。一部分是短消息自身的相关信息。这部分当中，isAuthority 表示发件人号码是否为“权威号”，例如，10086（中国移动客服）、95533（中国建设银行客服）等某些大企业的服务号码。function 表示短信内容的检测结果，包括 good（正常）、spam（垃圾消息）、cheat（诈骗消息）等分类。据报道，百度手机助手对于垃圾短信（以短消息的文本来判断）的识别率可以达到99% [2]。

第二部分是收到短消息的同时，手机能够扫描到的 Wi-Fi AP 的信息。wf 数组当中，每个对象是一个 AP 的记录，包括其 MAC 地址，以及查询到的地理坐标经纬度。此外，百度还用其内部的某种算法，对某些记录给出了 Wi-Fi AP 聚类的结果。百度没有对外公开其计算方法。

第三部分是手机最近连接的三个基站的信息。每个基站有唯一的 id，其格式为 MCC（移动国家代码，中国为 460）|MNC（移动网络号码，例如中国移动为 00、中国联通为 01）|LAC（位置区域码）|CID（基站编号）。通过基站 id，可以从各数据库当中，查询其对应的位置。如果百度从其内部的基站位置数据库中查询到这个位置，则将其记录下来。此外，通过 time 字段，可以知道连接到各个基站的时间顺序。

需要注意的是，这里提到的 Wi-Fi AP 和基站位置，并非其真实位置。绝大多数这样的定位提供商使用被称为“沿街扫描”（wardriving）和“众包”（crowdsourcing）的方法所获得的数据[3]。因此，用二者的编号查询到的位置，很可能包含误差甚至较大的错误。

2.3 数据的初步分析

2.3.1 基本数据

用于实验的样本数据共540304条，文件大小约644MB。function 字段表示的短信内容分类为：正常515268条，垃圾消息18475条，诈骗消息6561条。来自权威号的有201679条。

为什么要在收到短信的同时，记录此时搜索到的 Wi-Fi AP 情况？为了判断伪基站，一种直观的做法为：对现在连接的基站，用其 id 即编号，向“基站定位数据库”搜索其应当所处的位置。将这个位置与手机 GPS（全球定位系统）提供的位置对比，如果误差过大，就判断为伪基站。在上一章中提到过，不法分子通常会为伪基站随意编造一组与目前小区不同的 LAC（位置区域码）和 CID（基站编

号）。那么，例如 GPS 测量出手机位于北京海淀区，而这个编造的 id 对应的位置几乎不可能还在海淀区，而在其他城市，这就产生了两种定位之间距离的异常。

事实上，有一些文章[4][5]提出了这样的在线检测方法。但使用 GPS 定位存在一些问题：非室外环境无法使用、耗电量大、启动慢等[6][7]，以及可能带来的隐私问题。相反，使用手机附近的 Wi-Fi 接入点信息，也可以通过查询其位置来判断手机的位置。因此，百度手机卫士记录了收到短信时扫描到的 Wi-Fi AP，以此作为一条记录的位置信息。

图 2.2 展示的是每条记录当中，Wi-Fi AP 数量的分布。有至少两个 AP 的记录占约87.2%，可以认为大部分记录都能获得比较可靠的定位。除了单独考虑每条记录，我们还可以考察同一个 AP 在所有记录中出现的情况。在通常的误差之外，一些便携的 Wi-Fi 热点位置也可能经常改变，给定位带来困难。[6]如果一个 AP 在若干条记录当中都较好地与基站位置相匹配，那么我们就有较大的把握说它给出的定位是可信的。

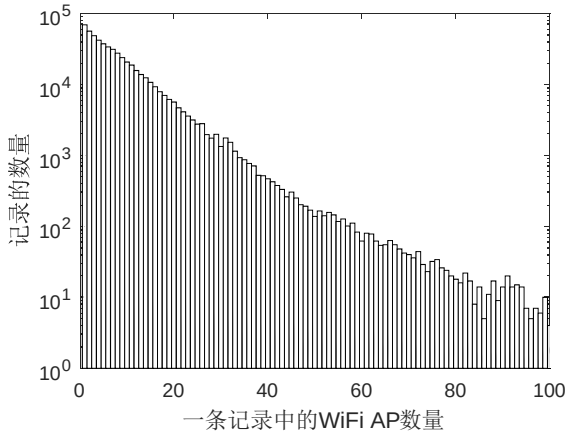

图 2.2 每条记录中 Wi-Fi AP 数量的分布

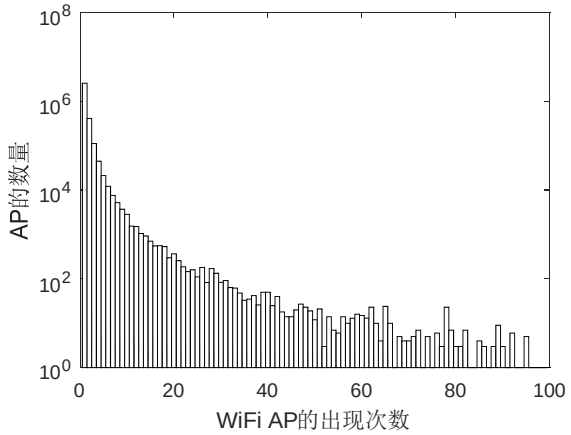

图 2.3 Wi-Fi AP 出现次数的分布

然而，图 2.3 表明在多条记录中重复出现的 AP 只占总数的19.7%（注意对数纵坐标）。基站的重复出现情况也类似。这很可能是因为实验中使用的数据并非百度得到的所有记录。因此，在本次研究中，暂时不在多条记录之间考虑重复出现的 AP 或基站信息。

最后，考虑基站的定位情况。每条记录当中，有基站都能查询到位置的只有49.1%。这有可能导致很大一部分记录无效。同时因为基站位置查询本身包含的误差较大，又使用 Google 提供的地理定位（Geolocation）API，对所有基站记录进行了查询。后者在基本覆盖百度数据的情况下，又增加了376684个基站的信息。

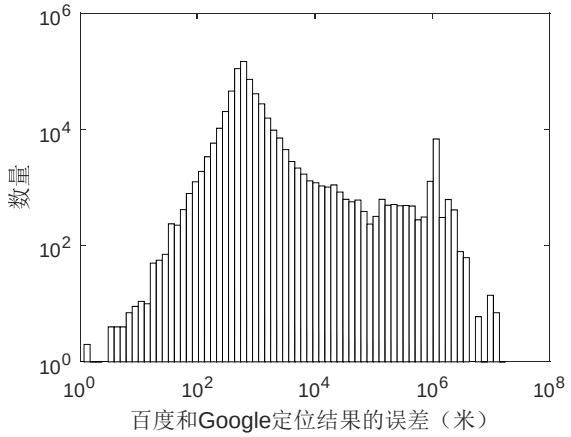

图 2.4 百度和 Google 基站定位的误差分布

如图 2.4 所示，百度和 Google 的基站定位结果中，超过95.4%的误差在10千米以内（注意对数坐标）。因为基站覆盖范围在千米量级，加上测量误差，大部分结果都能互相验证。但是，为了避免剩下的小部分数据对结果造成干扰，在此后的计算中，将同时使用两个来源的基站定位结果。

2.3.2 简单判别规则

如前所述，若只单独考虑每条记录，可以设计一些简单的判别规则，来筛选出可能存在异常的记录。

- 基站定位和 Wi-Fi 定位的距离：这种规则的原理在上一小节当中已经叙述。我们暂时利用百度计算的 Wi-Fi AP 聚类中心。对于百度和 Google 基站定位数据同时存在的情况，取它们到聚类中心距离当中较小的一个。图 2.5 展示了距离的分布。为清晰起见，图 2.5 绘制了距离的密度和累计分布。

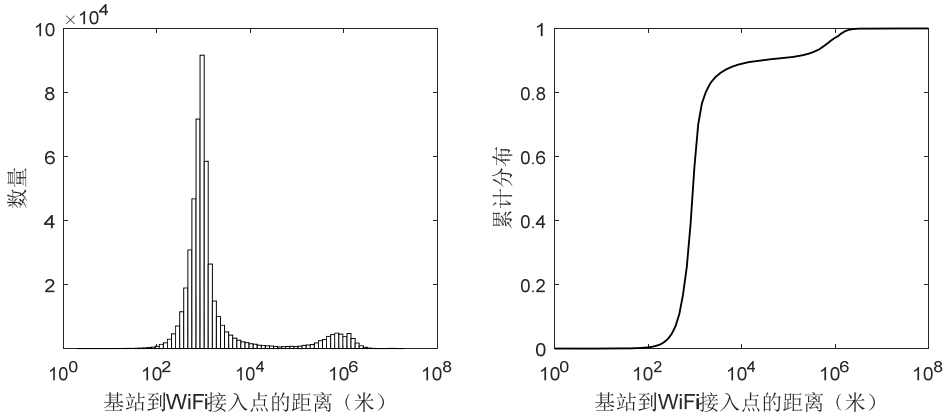

图 2.5 基站定位和 Wi-Fi AP 聚类中心距离的分布

在距离10^1到10^5之间，距离变量取对数似乎符合正态分布或 t 分布。在距离10^5以上的数据占9.05%，这些距离基本能够确定是异常的。因此，截出前一范围的数据（避免10^6附近峰的影响），进行拟合。拟合效果较好的为 t 位置-尺度分布，参数如下（注意，是数据取对数后的结果）：

表 2.2 基站与 Wi-Fi AP 距离的 t 分布拟合结果

参数	估计值	标准差
μ	6.74646	0.000715415
σ	0.376498	0.000871022
ν	2.4692	0.0124145

概率密度函数为

$$f(x|\mu,\sigma,\nu)=\frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\sigma\sqrt{\nu\pi}\Gamma\left(\frac{\nu}{2}\right)}\left[\frac{\nu+\left(\frac{x-\mu}{\sigma}\right)^2}{\nu}\right]^{-\left(\frac{\nu+1}{2}\right)},-\infty<x<\infty.$$

利用这个分布，我们可以选择异常距离的阈值ϵ_1。例如，如果认为距离最大的2%是异常记录，那么阈值可选为5.72×10^3米。此外，在后面的聚类初始种子生成中，还将利用这个模型来生成样本。

- 来自权威号的异常内容：如果 isAuthority 为真，而根据短消息内容判断的分类 function 为垃圾消息或诈骗消息，那么发送这条消息的基站可能是伪基站。

- 基站切换速度：前面解释过伪基站的工作原理，即诱使手机重新选择小区与其连接[9]。而伪基站编号对应的位置通常与实际位置相去甚远，因此，在连接基站的历史记录 base 当中，如果两个基站之间的切换速度（距离除以连接时间）超过某个阈值ϵ_2，则判定为异常。图 2.6 展示切换速度的分布（注意横坐标为对数）。

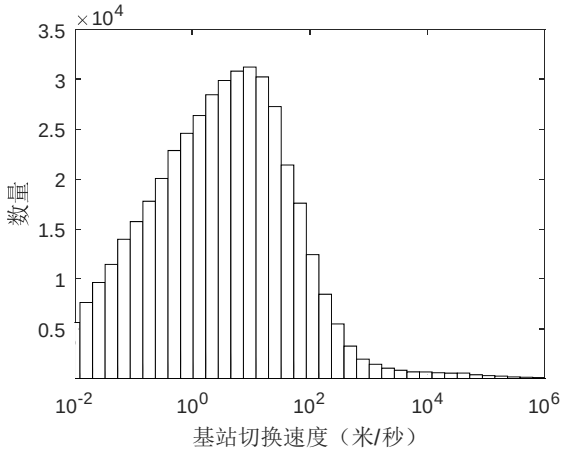

图 2.6 基站切换速度的分布

- 切换到 id 语法错误、查询不到的基站：这也符合伪基站工作原理。

将以上四种简单判别规则进行统计，其中基站定位和 Wi-Fi 定位距离的阈值ϵ_1分别取5000和25000，以及ϵ_2取50和500。在此之外，还考虑筛选同时符合两条规则的记录，它们是来自伪基站的异常记录的可能性更大。表 2.3 列举了筛选出的异常记录的数量。对角线上的数字表示某一种规则筛选的异常记录数量，其他位置是两种规则都判定为异常的记录数量。

表 2.3 简单规则筛选出异常集合交集的记录数量

	定位距离ϵ_1		权威号	切换速度ϵ_2		id 错误
	$5k$	$25k$		50	500	
ϵ_1 $5k$	57928		591	3669	2766	8134
ϵ_1 $25k$		43671	477	3631	2760	6362
权威号			3963	75	60	371
ϵ_2 50				4402		447
ϵ_2 500					3344	319
id 错误						42859

2.4 深入挖掘数据的方向

2.4.1 简单规则的缺点

上一节当中选择的简单判别规则，虽然计算简便直观，但是存在诸多缺点，例如：

- Wi-Fi AP 位置聚类不准确，致使第一种判别规则失效：百度使用的 Wi-Fi 定位数据库中包含不少错误的位置数据，而其使用的空间聚类算法不能很好地判断这种异常情况。

图 2.7 Wi-Fi AP 聚类错误导致误判的例子

（  ：百度/Google 基站定位，  ：Wi-Fi AP，  ：AP 聚类中心）

图 2.7 展示了上面$\epsilon_1 = 25k$判断的异常结果当中，明显是误判的两个例子。因为 Wi-Fi AP 数据库中错误记录过多，很可能给出每个 AP 的分散在较远的地方，而一般的聚类算法得到的基本是它们的几何中心，像图中这样的情况就导致聚类中心没有实际意义。相反，也有 AP 的位置在基站附近，这意味着 AP 和基站的位置实际上是相符的（从而其他 AP 的位置错误）。

- 阈值的选择需要人为干预，并且筛选结果没有很好的评价指标。
- 更多伪基站产生的记录无法用简单规则筛选得到：再次回顾伪基站的工作原理，基站 id 是由使用者随机选取的，那么这个 id 实际中根本不存在、从而无法查询到的概率应当更大。这样第一种规则就失效了。此外，由于

基站定位数据是实地扫描获得的，也有很多真实基站的位置无法查询到，或者数据存在误差和错误。最后，伪基站还有很多潜在的特征，不能由研究者人工发现或手动编码。

以上缺点的存在，导致使用简单判断规则得到很大比例的第一类错误和第二类错误。如果认为 Wi-Fi AP 聚类中心是手机当前所处的位置，并且伪基站在手机附近一千米之内，而将这些异常记录绘制在地图上，并没有形成明显的簇。

图 2.8 是将北京城区（$39°30'\mathrm{N}\sim40°30'\mathrm{N}, 116°\mathrm{E}\sim117°\mathrm{E}$）所有3008条异常记录，即计算四种简单判别规则异常集之并集（$\epsilon_1 = 5k, \epsilon_2 = 50$），并标记在地图上的结果。

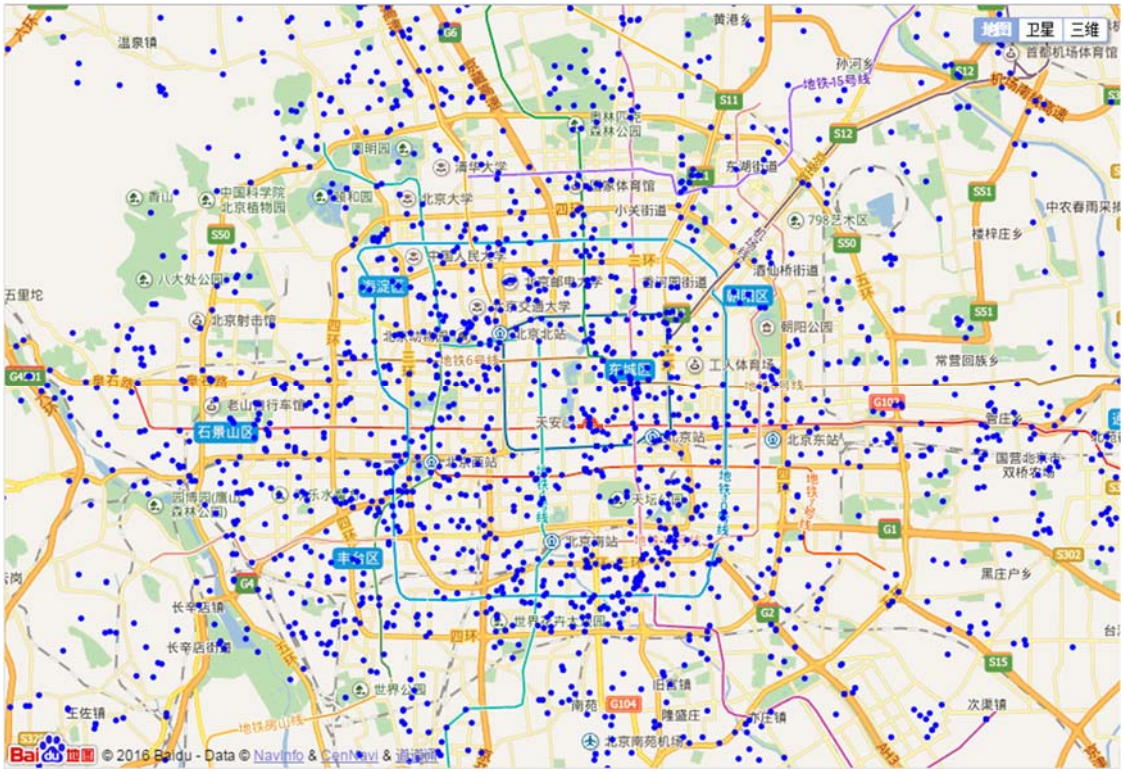

图 2.8 四种简单判别规则筛选出北京城区的伪基站

在地理分布上没有明显的簇，一方面使我们无法对伪基站进一步研究和治理，另一方面更提示得到的结果可能偏差过大。设立伪基站的不法分子往往在某一个地区频繁出现，并且一个伪基站也应当给很多手机发送了短信，从而在地图上形成簇。

2.4.2 研究方向

我们直观上想要设计前面的简单判别规则，最主要的原因是数据没有标注。也就是说，没有人确切地知道某条记录是否对应伪基站。这使得我们无法直接利用伪基站数据集来训练某种分类器，也无法直接对判别结果进行交叉验证等评估。然而本章的尝试已经初步说明，仅仅使用人工编码的规则对于解决问题来说很可能是不够的。

另一方面，如果把每条记录当中的特征看作一个高维空间$\mathcal{F}^n$，我们实际是在这个空间当中选择某些区域里的点$\{x_i\} \subset \mathcal{F}^n$。例如，认为 Wi-Fi AP 与基站定位之间的距离d_1是一个特征，则异常点集是$\{x \in \mathcal{F}^n | d_1(x) \leq \epsilon_1\}$。如果反过来考虑，先将点按照彼此距离的大小划分为一些簇，然后再去判断这个簇是否包含的是异常点。例如从图 2.5 上可以看出按d_1维度有两个明显的簇。比起人工编码的特征来说，这样能够捕捉更多的异常并且具有一定依据。正如一句西方谚语：“如果它走路像鸭子、叫起来也像鸭子，那么它就是鸭子。”[10]

这就是统计学习当中的聚类分析。我们利用它对$\{x_i\}$进行处理，使得相似的点被划分为同样的种类（簇），它还可以提供单个点$\hat{x}_k$来刻画簇k中所有点的共同特征。这样，可以根据簇k当中某些点或聚类中心$\hat{x}_k$来判断整个簇当中点的分类。

但是，传统聚类得到的簇仍然需要人工赋予含义（例如判断是否来自伪基站），并且也没有利用到对伪基站工作原理的已有知识（先验信息）。因此考虑用带随机初始种子的半监督聚类[11]，也就是说，我们模拟伪基站的工作原理，随机产生一些伪基站发送短信所产生的记录以及正常记录，以它们为初始聚类中心，来指导聚类过程。 假如得到的簇中包含一些生成的正例或反例点，那么这个簇的分类就可以确定。此外，因为这些初始种子并不包含所有的可能情况，要允许一些簇中不含有初始种子。这是不完整先验信息的半监督聚类[12]。

第 3 章 聚类与半监督聚类

如第 2 章最后一小节所述，我们接下来的研究方向是利用聚类分析技术，将记录按照特征划分为若干集合，以期从中找出异常记录的某种共性。在本章中，我们先回顾研究中将要使用的几种聚类方法。在传统的无监督聚类之外，针对伪基站大数据的特点，还选择应用了不完整信息的半监督聚类技术。

3.1 基于原型的聚类：k-means

对于特征空间中的点$\{x_1, \dots, x_m\} \subset \mathcal{F}^n$，我们想要将其划分为$k$个集合$C_1, \dots C_k$，使得在某种意义上，同一个集合中的点尽可能地相似，而不同集合中的点不相似。在这方面最经典的是 k-means 算法[13]。在对伪基站大数据记录聚类的处理当中，将会使用它的某种改进版本。因此，首先回顾 k-means 的基本原理。

3.1.1 点划分到质心

每个集合C_i（簇）的中心点称为质心，它是最能代表这个集合的点，而后面我们会看到它正和物理中的质心相关。

对于 k-means 来说，划分簇实际是为簇C_i选择质心c_i的过程。因为一旦确定了质心，把所有点x_j划分到距离它最近的那个簇，是最好的选择。这可以形式化地表示为$C_i = \{x_j | d(x_j, c_i) \le d(x_j, c_t), \forall t\}$（相同距离则任选其一）。

接下来要考虑的就是所选用的距离$d(x, y)$。对于欧氏空间中的点，通常使用欧几里得距离$\mathcal{L}_2$，也可以考虑曼哈顿距离$\mathcal{L}_1$等，取决于更关心各特征的平均差异还是最大差异。此外，还有文档等应用的余弦距离等。

3.1.2 目标函数

聚类的目标是最小化到所属质心距离之和，对于$\mathcal{L}_2$距离记作SSE（误差的平方和），而$\mathcal{L}_1$距离下记为SAE（误差绝对值的和），以此作为衡量聚类好坏的函数。它定义为$SSE = \sum_{i=1}^{k} \sum_{x \in C_i} d(x, c_i)$。

在划分好簇之后，如何选择质心c_i使SSE最小？为简单起见，先从一维情况开始考虑。事实上，多维的情况是一样的，因为只需要对每个维度重复下面的操作即可。

对于$\mathcal{L}_2$距离，通过对SSE求导来求解最好的质心c_k：

$$\frac{\partial}{\partial c_k}SSE = \frac{\partial}{\partial c_k}\sum_{i=1}^{k}\sum_{x\in C_i}(x-c_i)^2$$

$$= \sum_{i=1}^{k}\sum_{x\in C_i}\frac{\partial}{\partial c_k}(x-c_i)^2 = \sum_{x\in C_k}2(x-c_k) = 0$$

$$\Rightarrow |C_k|c_k = \sum_{x\in C_k}x \Rightarrow c_k = \frac{1}{|C_k|}\sum_{x\in C_k}x$$

也就是说，应当选择c_k为C_k所包含点的质心，使得对于此划分，SSE最小。而选取$\mathcal{L}_1$距离时，则稍有不同：

$$\frac{\partial}{\partial c_k}SAE = \frac{\partial}{\partial c_k}\sum_{i=1}^{k}\sum_{x\in C_i}|x-c_i|$$

$$= \sum_{i=1}^{k}\sum_{x\in C_i}|x-c_i| = \sum_{x\in C_k}|x-c_k| = 0$$

$$\Rightarrow \sum_{x\in C_k}sign(x-c_k) = 0 \Rightarrow c_k = median\{x\in C_k\}$$

于是在这种情况下，簇的质心是簇中所有点的中位数。

3.1.3 k-means 算法

文献中已经证明，对于上面所述的$\mathcal{L}_2$距离的 k-means 问题，即使只寻找簇的个数$k=2$时[14]，或者对于一般的k而空间的维数$n=2$时[15]，都是 NP 难的。因此，k-means 算法（也称 Lloyd 算法[13]）通过迭代选择簇中心，来获得某种局部最优解。而以不同的随机初始簇中心来运行 k-means 算法，则在实际中能提高聚类的质量。因为在本研究当中，并不使用完全随机化的初始种子，因此对于初始簇中心的选择不详细叙述。

算法开始时，随机产生k个点c_k，作为初始的簇中心。然后，每一次循环（迭代）当中，先把所有的数据点$\{x_i\}$归于距离其最近的质心所在的簇，这样得到当前选择的质心所对应的簇。接着，根据簇中所有的点，来更新簇的质心。用这些新的质心再进行迭代。

因为计算误差的存在，我们允许算法在迭代一定步数后停止，或者当某次更新的幅度足够小就停止。这包括质心更改的距离，以及两次划分之间的差异等等。图 3.1 形式化地呈现了经典 k-means 算法的过程。

算法：k-means

输入：$\{x_i\} \subset \mathcal{F}^n, k$

输出：$\{C_j\}, \{c_j\}$

过程：

1　（随机）选取$c_1, \dots, c_k$。

2　重复以下过程：

　　2.1　将点分划到最近的质心，即根据$\{c_j\}$选取$\{x_i\}$的最优划分$\{C_j\}$。

　　2.2　重新计算每个簇C_j的质心c_j。

　　2.3　如果循环次数达到预设值，或质心/簇更新足够小，停止。

图 3.1 k-means 算法

易见算法的空间复杂度与输入同阶，而时间复杂度为$O(Ikmn)$，其中I是迭代次数。

此外，也有文献[16]指出，离群点可能会影响聚类的效果，特别是使用$\mathcal{L}_2$距离时（它对大数更敏感）。因此在使用 k-means 时，可能会需要对数据进行预处理。然而，我们感兴趣的恰恰是异常点，所以不能简单地剔除不需要的点。

因为 k-means 的特点，我们还应该使距离度量尽量均匀。例如，如果直接把基站和 Wi-Fi AP 的距离作为特征，那么0米和3000米的差别，肯定比10^6米和$(10^6 + 3000)$米的差别大，因为后者都是同样异常的距离。最简单的处理似乎是对距离取对数。但是这个例子揭示，在聚类之前需要更精细地考虑预处理和异常检测的步骤。

3.1.4 确定簇的个数

对于 k-means 算法，参数k似乎需要人为调整，但是我们可以用一些非监督的评价方法，来自动化这个过程。除了观察SSE的变化趋势，还有以下的统计量可以对每个k产生簇的结果的优劣：

- 轮廓系数[18]：用来衡量每个数据点x_i与其所属的簇C_j中其他点的相似程度，以及与其他簇中点的差异程度。因此，用

$$a_i = mean\{d(x_i, x_t) | x_t \in C_j\}, b_i = \min_{r \neq j} mean\{d(x_i, x_t) | x_t \in C_r\}$$

来表示这两种性能。a_i是x_i与同簇点的平均距离b_i是x_i到最近的另一个簇中点距离的平均距离。而最后，轮廓系数为二者之差的归一化值

$$s_i = \frac{b_i - a_i}{\max\{a_i, b_i\}}$$

它的取值在$[-1,1]$，越接近1越好（此时$a_i = 0$）。

选择不同的k分别进行 k-means 聚类，则平均轮廓系数会呈现出先上升再下降的趋势，并且SSE也会对应地出现拐点（尽管SSE显然随着k是单调下降的）。用这种方法，我们或许能够让此聚类过程无需人为干预，特别是因为在本研究的应用当中，特征空间维数较高，且不一定每一维都有直观含义。

3.2 基于密度的聚类：DBSCAN

DBSCAN[17]是一种基于密度的聚类算法，它能在数据中找到密集程度高的块，并且忽略背景中散布的噪声。我们主要考虑将其应用在伪基站的空间聚类当中。参考图 2.8，如果能通过改进的方法，找出更多（也更准确）的疑似伪基站，那么即使有一定比例的假阳性结果（被标记异常的正常记录），在真正伪基站频繁出现的地区，蓝色标记的密度也会更大，于是 DBSCAN 能够发现这样的密集区域。

3.2.1 按照中心密度将点分类

点x_i的密度定义为其eps半径邻域当中点的个数。又根据一个预先设定的阈值$minPts$，将点分为以下三类：

- 核心点：eps邻域当中，点的个数不小于$minPts$者。
- 边界点：不是核心点，但它在某个核心点的eps邻域中。
- 噪声点：其他的点。

3.2.2 DBSCAN 算法

给定以上定义，DBSCAN 算法的目标是找出所有临近的核心点，形成核心点的簇，然后由核心点来决定其关联的边界点的归属。如果用图论的观点来看，给所有距离小于eps的核心点之间连一条边，再给边界点到它属于的某一个核心点（有多个则任选一个）连边，则 DBSCAN 发现所有非孤立的连通分支。图 3.2 是 DBSCAN 算法的形式化叙述。注意，这里没有考虑算法实现上的某些技巧。

算法：DBSCAN
输入：$\{x_i\} \subset \mathcal{F}^n, eps, minPts$
输出：点的分类$\{t_i\}$，每个簇包含的点$\{C_j\}$

过程：

1 按照上面规则，将每个点x_i分类成核心点、边界点或噪声点，记为t_i。

2 在距离小于eps的核心点之间连一条边。

3 找出非孤立的连通分支，归入簇的集合$\{C_j\}$中。

4 将每个边界点划分到某个与之关联的核心点所在集合C_j中。

图 3.2 形式上简化的 DBSCAN 算法

关于复杂度，在二维情况下，使用 kd 树等数据结构，可以将时间复杂度降低到$O(m \log m)$。但最近的研究[20]表明，对于三维及以上情况，时间复杂度有下界$O(m^{4/3})$。空间复杂度不高于时间复杂度。

3.2.3 参数的确定

为确定参数eps和$minPts$，可以考虑距离每个点最近的第k个点，称为k-距离。如果点应当在某个簇当中，则k应当很小；而对噪声点，k-距离相对而言会比较大。如果对于一个选定的k，绘制k距离的分布图（横轴：k-距离；纵轴：点的个数），会看到在某个k-距离附近，曲线有明显的拐点。这样可以选择此横坐标值为参数eps，而k对应$minPts$。在 DBSCAN 的原始文章[17]中，对于二维情形，推荐的$k = 4$，不过此参数也可以调整。

DBSCAN 的重要假设是，簇的密度变化不太大。否则，如果采用单一的关于距离和点数的阈值（eps和$minPts$），就会漏掉较为稀疏的簇，或者误选中较为稠密的噪声。对于在二维地图上为疑似伪基站聚类的这个应用，需要注意选择期望簇密度较为一致的地区。

例如，同样的伪基站，在闹市的小区里发送短信数比乡镇里的多，从而产生的异常记录更多；噪声（假阳性的记录）也更多。因此，应当对每个范围的区域（例如海淀区）单独进行 DBSCAN 计算。

3.3 半监督聚类及其改进

到目前为止，我们考虑的都是无监督学习问题。前面说过，假如数据中对于每条记录有是否伪基站的标签，就可以使用众多的分类技术来训练一个模型，然后用它对此后观察到的数据点进行分类。这是所谓的监督学习。但是我们没有明确的答案，那么传统的做法是使用聚类或关联分析等无监督学习技术，对于所有

记录，只是按照它们的相似程度作出某种划分。机器对于每个分类的意义是完全不知情的。如果要应用上面 k-means 或 DBSCAN 的聚类结果，需要完全依靠人来给每个簇赋予实际含义。

但是，我们对于伪基站或正常基站产生的记录，有一定的先验知识。例如最简单的，如果 Wi-Fi 聚类中心和基站位置相差在1千米之内，几乎可以确信这是真实基站。如何将这些先验知识应用到聚类当中？利用先验信息产生一些数据点，它们的标记（是否伪基站）是可以确定的，但是它们的质量还不足以单独训练一个分类器。因此，用这些数据点去寻找临近的原始数据，作为下一次聚类的依据，这就是我们使用的半监督聚类的思想。在达到将聚类化为分类这一目的的同时，还可以提高 k-means 聚类的质量，因为前面提到过，原始版本的 k-means 容易陷入局部极值当中。

3.3.1 带有先验初始种子的半监督聚类

基于上面叙述的思想，有两种相似的k-means改进算法被提出[11]，称为Seeded-KMeans 和 Constrained-KMeans。其区别在于：前者的种子此后与原始数据视作等同，而后者的种子在此后每次迭代中不更改所属的簇。

两种算法的共同流程为：由先验知识给出每个簇的初始种子，记为集合$S_1, \dots, S_k$。将这些初始种子的中心作为每个簇的初始中心。然后，按照经典 k-means 的步骤迭代。重复此过程直到循环结束的标准达到。

图 3.3 形式化地表述了两种算法的流程。

算法：Seeded-KMeans 和 Constrained-KMeans

输入：$\{x_i\} \subset \mathcal{F}^n, k, \{S_1, \dots, S_k\}$

输出：$\{C_j\}, \{c_j\}$

过程：

1 令c_j为S_j所包含点的质心。

2 重复以下过程：

2.1 将点$\{x_i\}$分划到最近的质心，对于 Seeded-KMeans 还有各$\{S_j\}$中的点。

2.2 重新计算每个簇C_j的质心c_j。

2.3 如果循环次数达到预设值，或质心/簇更新足够小，停止。

图 3.3 Seeded-KMeans 和 Constrained-KMeans 算法

因为 Constrained-KMeans 算法中，初始种子所属的簇不再改变，所以它更适于初始种子没有噪声的情况。另一方面，即使初始种子也有少量噪声，Seeded-KMeans 允许它们改变所属的簇，所以也不会对聚类结果产生很大的影响。根据算法设计者的实验[11]，使用这样的半监督聚类技术，在大部分数据集上都会有更好的分类效果。

3.3.2 不完整先验信息

在实际应用当中，常见的情况是先验知识并不包括所有簇的信息。这时，上面的两种方法就会遇到困难，因为它们划分簇的个数k是确定的，并且没有给预先不知道的簇预留位置。例如，如果平面上有三个分离的球形簇，而只在两个球中有先验的种子，那么用上一节的方法，只能得到两个簇，它们可能各包括第三个球的一部分。这显然不是我们想要的结果。

而在本研究当中，出现的情况恰好就是这样，我们只能预先对伪基站的工作原理及其可能产生的特征有部分了解，而很难事先为所有情况设定好种子。因此，需要使用不完整先验信息下的半监督聚类。

一种直观的方法是，假如我们事先准备了k个簇的初始种子，又预计有l个簇是目前未知的（这个值也可以用 k-means 确定k的方法来自动调整），那么不妨与 k-means 的初始方式类似，从空间中随机产生l个点，作为未知簇的初始种子。但是有实验表明[12]这样聚类的效果不好。

为了利用准备好的初始种子来为剩下的簇产生初始质心，FS-KMeans 算法[12]被提出。它在产生其余l个初始质心时，每次从目标值最大的m/k个点中，随机抽取p $(p < m/k)$个点，将其中心作为一个初始质心。直观地看，如果一个点距离目前最近的质心还很远，那么它有可能是另一个位置簇当中的点。而求平均值是为了避免少数几个离群点的影响。

基于此，本研究中对这种初始方法略作修改，称为 FAS-KMeans 算法。不同之处在于，在产生第$k+h$个初始点时，希望它距离第$1,\dots,k,k+1,\dots,k+h-1$个点都尽量远。图 3.4 是 FAS-KMeans 算法的描述。

算法：FAS-KMeans
输入：$\{x_i\} \subset \mathcal{F}^n, k, \{S_1,\dots,S_k\}, l, p$
输出：$\{C_j\}, \{c_j\}$
过程：

1 令c_j为S_j所包含点的质心。

2 对于$h = 1, \ldots, l$，依次执行：

2.1 选出目前目标函数最大的m/k个点的集合T_h。

2.2 从T_h中随机选p个点，计算出质心c_{k+h}。

2.3 更新每个点的目标函数值。

3 重复以下过程：

3.1 将点$\{x_i\}$分划到最近的质心。

3.2 重新计算每个簇C_j的质心c_j。

3.3 如果循环次数达到预设值，或质心/簇更新足够小，停止。

图 3.4 FAS-KMeans 算法

注意，图 3.4 的步骤 3.2，仍然可以类似 Seeded-KMeans 和 Constrained-KMeans 的区别，根据需要决定是否对$\{S_j\}$的点也重新划分。

可类似地证明[21]，FAS-KMeans 等以上算法都和经典 k-means 一样收敛到局部最优解。（回忆 k-means 问题是 NP 难的，所以除非$P = NP$，不存在求全局最优解的有效算法。但众多实际应用已经表明，k-means 算法及其改进通常能给出令人较为满意的聚类结果。）

有了这些准备，接下来可以开始针对本研究的具体数据，来设计大数据中的伪基站检测方法。

第 4 章 大数据中的伪基站检测方法

有了之前的准备，现在我们考虑如何用半监督聚类方法，筛选伪基站大数据当中的异常记录。首先，因为每条记录的部分字段不等长（例如可能有多个 Wi-Fi AP 扫描结果），以及我们关心的可能不是各原始变量的线性关系，所以应当从中选取一些与异常检测有关的特征。这些数值要进行仔细的归一化处理。接下来通过第 2 章人工规则类似的方法，选取一部分几乎确定是正常或异常的记录，作为之后评价聚类结果的一种判据。准备好这些，我们描述产生初始种子以及用它们进行半监督聚类的方法，并且对筛选出的异常记录进行空间上的基于密度的聚类。最后，用现有的这份伪基站大数据样本，进行实验并对效果做出评价。

4.1 特征与预处理

4.1.1 特征的选取：有基站定位的情形

从每条有基站定位信息数据当中，我们选择以下一些数据作为该记录的特征 x_i，构成特征集合$\{x_i\} \subset \mathcal{F}^n$。

因为各属性可能有很大差别，而基于 k-means 的聚类算法以及常用的范数都倾向于平等对待特征的每个维度，并发现近似球形的簇，所以需要格外注意标准化处理技巧。

例如，第 2 章中出现的地理距离有可能高达10^7米，而经过尝试，对其取对数能够比较符合简单的概率分布，此后再考虑归一化。又如数据中有一些非数值类型的字段（isAuthority 和 function 等），而它们对于寻找异常数据来说可能同等重要，因此也需要用适当方法将其数值化。

- 基站定位与最近的 Wi-Fi AP 之间的距离：第 2 章的分析表明，Wi-Fi AP 定位的错误是比较普遍的现象，特别是类似图 2.7 所展示的情况，即使 AP 的数量比较多，用通常的聚类算法获得的中心点，与实际位置也可能相去甚远。这样会产生很多假阳性点。

  另一方面，如果使用 Wi-Fi 扫描结果当中，与基站之间距离最小的一个，则在减少第一类错误（H_0：正常记录）的同时，几乎不会产生第二类错误。这是因为，正如前面多次提到的，伪基站工作时随机产生 id，则该 id 对应位置恰好在真实位置附近的几率几乎为零。

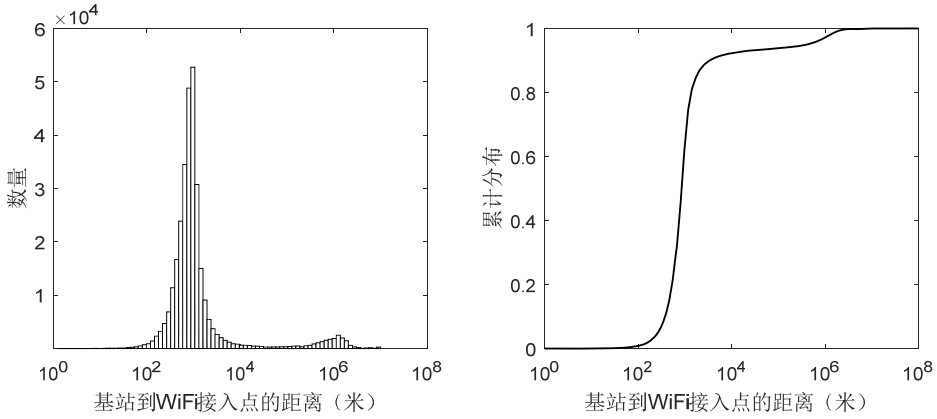

图 4.1 最后连接的基站定位与最近的 Wi-Fi AP 距离的分布

图 4.1 展示了最后连接的那个基站与最近的 Wi-Fi AP 距离的分布，与图 2.8 的含义相似。我们只考虑 base 记录当中，连接时间 time 最晚的那个，因为之前连接的基站可能与现在位置距离较远，从而产生假阳性。

为了将取对数后的距离进行标准化，采用以下函数[6]：

$$x^1 = \frac{1 - \log\ \min d(Base, WiFi_i) / \log \epsilon_1}{1 + \log\ \min d(Base, WiFi_i) / \log \epsilon_1}$$

其中ϵ_1是预设的一个阈值，从第 2 章中，我们知道可以取为10^4~10^5的某个值。此函数在$d = \epsilon_1$时值为0，而距离为0或无穷大时分别为-1和1。

- Wi-Fi 聚类中心与最近 AP 的距离：若$x^1 > 0$，意味着没有一个 Wi-Fi AP 的位置在基站应当在的位置附近。然而这是否因为 AP 过少或定位全部错误（如图 2.8）？对于这种情况，考察

$$x^2 = \begin{cases} 0, 若 x^1 \le 0 \\ \max(0, 1 - (\lg\ \min d(Center, WiFi_i)) / 6), 若 x^1 > 0 \end{cases}$$

即对 Wi-Fi 聚类中心用同样标 1 准来判断其可信程度。如果它能真实地当前实际位置，则附近应当有 Wi-Fi AP，从而$x^2 \approx 1$（更有可能是真实的异常记录）。相反，对于图 2.8 这样分散的情况，x^2将是接近0的数值（更可能是 Wi-Fi 定位错误而产生的x^1异常）。

- 来自权威号的异常消息：这个判据我们在之前已经分析过，通常意味着这条记录异常，此特征记为：

$$x^3 = \begin{cases} 1, 若 isAuthority 且 function \ne good \\ 0, 其他 \end{cases}$$

- 最新两个基站之间的切换速度：回忆图 2.6，如果依次连接的两个基站 id 对应位置间的距离，明显超过这段时间内手机能移动的范围ϵ_2，则可能意味着异常。将此特征记作：

$$x^4 = \min(\frac{d(Base_2, Base_1)}{Time_2 - Time_1}/\epsilon_2, 1)$$

- 切换到 id 不符合语法的基站：有一部分记录中的基站 id 不符合正常的格式（MCC|MNC|LAC|CID），例如 460|00|-1|-1，如果突然切换到了这样的基站，则有可能是伪基站，记为$x^5 = 1$。
- 信号强度：如第 1 章中所述，伪基站通过增大信号强度而诱使手机进行小区重选。因此，如果最近连接的基站信号强度明显比正常值大，则有可能是异常记录。为确定归一化该特征的方法，先对所有记录中基站的信号强度绘制直方图，如图 4.2。

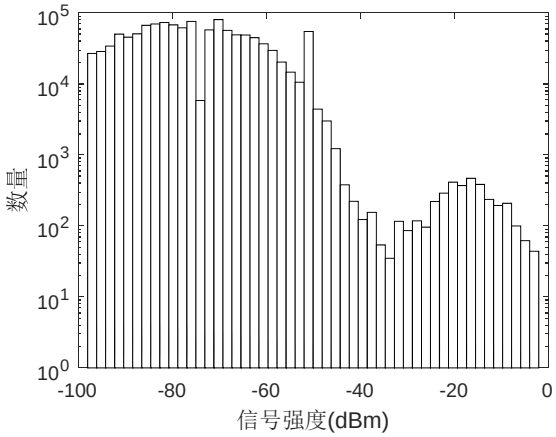

图 4.2 基站信号强度的分布

信号强度分布在$(-100,0)$当中，从图 4.2 中还可以发现（注意对数纵坐标），-40以下的记录占99%以上。因此，记此特征为

$$x^6 = \max(1 + \frac{signal}{40}, 0)$$

4.1.2 特征的选取：无基站定位的情形

在第 2 章的分析中已经提到，如果一个伪基站随机生成 id，则该 id 在基站定位数据库中查询不到的概率更大。因此，除非收集到的数据足够多，我们不能忽略这一部分的记录。但是，因为缺少上一小节当中的某些特征（例如x^1和x^2），需要选取适合这种情况的特征如下：

- 特征x^3, x^4, x^5, x^6：上一小节中的这些特征不依赖于基站定位，从而可以继续使用。
- 从正常定位的基站切换到无法定位的基站：用上次连接基站与 Wi-Fi 定位之间的误差，来刻画从正常工作的状态切换到未知基站的异常程度，用式子表示为：

$$x^7 = \begin{cases} 0, \text{若上个基站没有定位} \\ 1 - \dfrac{1 - \log \min d(Base', WiFi_i) / \log \epsilon_1}{1 + \log \min d(Base', WiFi_i) / \log \epsilon_1}, \text{其他} \end{cases}$$

其中$Base'$表示上一个连接到的基站。

4.1.3 人工规则选取正例与反例

第 2 章中研究的人工设定的规则，并不是完全没有作用。例如，虽然我们不知道基站定位与 Wi-Fi 定位距离过大，是来自伪基站还是定位错误；但如果这个距离足够小，那么它是随机生成 id 时正好对应了附近的基站，这件事的几率几乎为零，于是可以认为这是一条正常记录。

选取的这些正例（认定是伪基站）和反例（认定不是伪基站），由于和第 2 章相似的原因，不能直接用来训练一个分类器，然而可以用它们作为评价聚类质量的一个参考标准。也即，如果我们得到的聚类结论是正确的，那么至少应当满足将这些记录正确分类这一必要条件。此外，在需要时，它们也可以用来帮助判断聚类得到的簇是否为异常簇。

我们将以下两种情况选择为确定的正常记录：基站定位和 Wi-Fi 定位误差小于$\epsilon_3 = 5000$；最新两个基站都有定位且位置之差小于$\epsilon_4 = 10000$。选择以下两种情况为确定的异常记录：来自权威号的异常消息；信号强度大于-30。将这些集合分别记作N_1, N_2和P_1, P_2，表 4.1 记录了各类正例和反例的数量。

表 4.1 人工规则选取的正例和反例数量

集合	计数	占全部记录
N_1	279922	51.8%
N_2	255796	47.3%
$N_1 \cup N_2$	308815	57.2%
P_1	4084	0.76%
P_2	14673	2.72%
$P_1 \cup P_2$	18757	3.47%

反例的数量明显大于正例，这也是意料当中的：大部分记录只是来自非通讯录联系人的正常短消息。这不影响我们用这些数据评价的效果。此外$P_1 \cap P_2 = \emptyset$，说明不同的伪基站可能产生不同的效果。

4.2 带有初始种子的不完整半监督聚类

前面已经介绍了这种半监督聚类的原理，将采用图 3.4 所示的 FAS-KMeans 算法。针对这个具体问题，我们要考虑两个问题：产生怎样的初始种子，以及怎样处理聚类所得的结果。

第一种产生初始种子的方法是从上一小节选出的正例和反例当中，随机选取一定比例的记录$\bar{P}_i$和$\bar{N}_i$，作为初始种子。它的主要问题在于，正例或反例的全体集合本身没有明确的簇划分，不符合改进 k-means 算法的要求。一种解决方法是，先对$\bar{P}_i \cup \bar{N}_i$做经典 k-means 聚类，将其划分为k个簇$\{S_1, \dots, S_k\}$，然后用这些簇初始化不完全信息的类 k-means 聚类。注意在用正例和反例集检验聚类效果时，应当去掉这些初始种子（从而实现交叉检验）。这种方法的缺点是选出的初始种子类型比较单一，有可能会产生很多没有初始种子的簇，即算法中的h应当较大。

第二种方法是根据前面观察到的各数值分布，来模拟正常基站和伪基站的工作情况，产生对应记录。用户的状态分为两种：静止和移动，对于后者，按照图 2.6 所示的速度分布，产生一条移动轨迹。

如果要产生一个正常基站，就在所需点的附近选择一个位置，其距离服从图 2.5 所示的分布。它的信号强度小于-30，且与距离成正比。这个基站的 id 是合法的，并且查询到了其所处的位置，不会产生来自权威号的伪基站。也产生一部分正常基站，它们的位置在数据库中查找不到或查找错误。将这样的很多条记录，按照其不同的产生条件，划分为代表正常基站的初始簇$S_{N_1}, \dots S_{N_s}$。

如果要产生一个伪基站，则随机生成一个 id，然后用 Google 基站定位数据库查询其对应位置，作为该基站的查询结果。注意伪基站只可能是最近连接到的基站，而其余两个是正常基站。该伪基站有可能产生异常短消息。其信号强度至多为0，最小为$C_2 = Signal_{now} - ACCMIN + CRO$，这代表手机发生小区重选时，来自伪基站的信号应当具有的最小信号强度[9]。式中$Signal_{now}$为手机当前接收到的信号强度；$ACCMIN$是最小接入电平门限，一般为$-110 \sim -100$；CRO表示小区重选偏移量。伪基站正是用较强的信号来诱使手机进行小区重选的。我们产生的记录当中，伪基站的信号强度也应当明显高于上一个正常基站。这样，我们同样得

到了伪基站所在的初始簇$S_{P_1}, \dots S_{P_t}$。

聚类的过程不再赘述，其中参数的选取在实验部分解释。当我们得到了最终给出的聚类结果$S_1, \dots S_{k+l}$，可以依据簇$S_1, \dots S_k$的初始分类，来确定其中点的种类；而其余的簇，可以根据与前k个簇的距离等特征，人工决定其类别。

4.3 疑似伪基站的空间聚类

在第 3 章中，我们回顾了基于密度的聚类算法 DBSCAN。用上一节的方法筛选出疑似伪基站时，就可以进一步将其标示在地图上，进行空间上的聚类。我们假设百度计算的 Wi-Fi 定位结果是准确的，可以表示此时手机实际所处的位置。这样，一条伪基站记录说明在手机附近存在伪基站，从第 1 章中我们知道通常不超过一千米左右，那么用手机此时的位置就代表伪基站位置的期望。

从 DBSCAN 的算法流程我们可以看出，选择核心点的密度参数eps和$minPts$是固定的，那么如果考虑的空间范围过大，会出现某些地区的噪声密度比另一些区域的核心点密度更高的情况，从而导致该算法失效。因此，应当针对某个较小的感兴趣的区域单独进行这一步操作。

4.4 实验

用 C#和 MATLAB 实现了上述算法，来对数据中的异常记录进行检测。首先，综合 4.2 节所描述的两种方法产生初始种子。然后，考虑 k-means 系列算法的主要问题——选择簇的个数k。

在 3.1.4 节中，我们描述了通过聚类的内部特征，来确定恰当k的两种指标。对于$k \in [2,20]$，分别运行 FAS-KMeans 聚类，得到其平均距离和平均轮廓函数的变化曲线如图 4.3。

从图中可以看出，SSE（左图）在$k \in [2,5]$时快速下降，大约$k > 8$时已经趋于平稳。而平均轮廓系数（右图）的变化则更为明显，在$k \in [2,7]$上升，然后一旦$k \geq 8$，就呈现出明显的下降。根据前面对轮廓系数含义的描述，数值越接近1，说明点与当前簇的相似度越大，同时与其他簇的相似度越小。当$k \in [2,7]$，增加簇的数量能将本来就不同的簇分开，使轮廓系数上升；$k \geq 8$时则有些簇被不必要地划分了，从而使一些点处于中间状态，轮廓系数降低。最后，平均轮廓系数最高能达到0.9，说明此时聚类的效果良好。

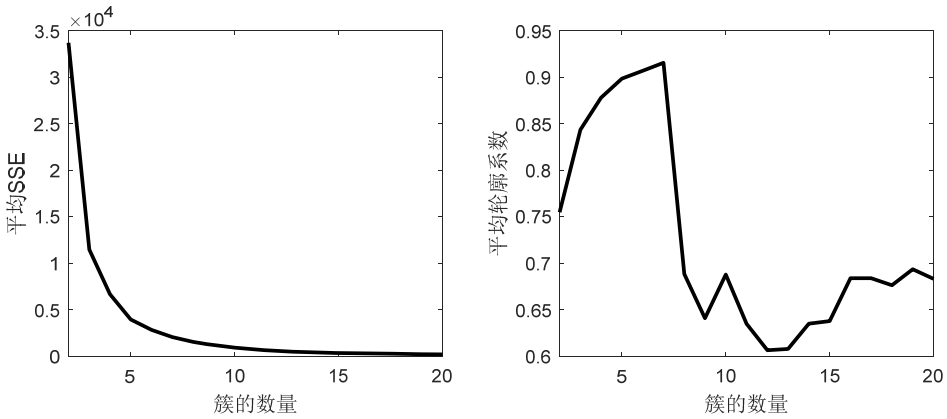

图 4.3 平均到簇中心距离和平均轮廓系数随簇的数量的变化

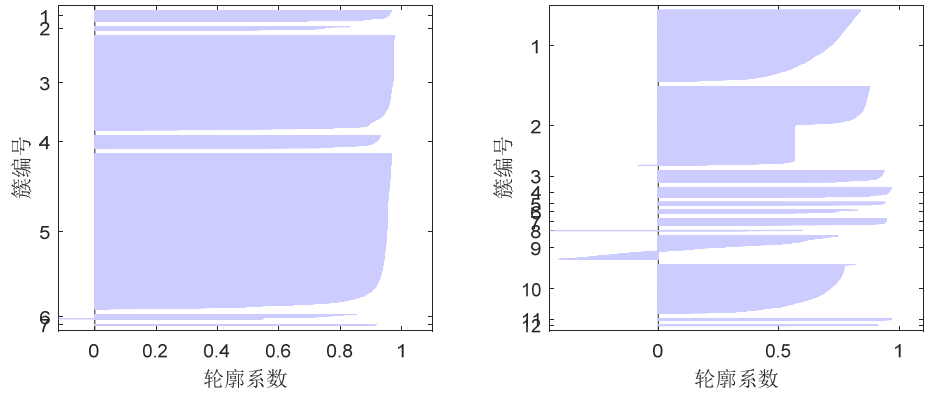

图 4.4 正确的簇个数（7，左图）和错误的簇个数（12，右图）对应的轮廓系数分布

为了进一步确认，可以观察对于某个指定的簇个数，轮廓系数在各簇当中的分布情况。图 4.4 的左图显示簇的数量正确时，每个簇中的大部分点都有接近于1的轮廓系数，显示它们的分类正确性很高。而右图是簇个数错误时的情况，轮廓系数明显更低，甚至还有一些为负，意味着它们有可能被归入了不正确的簇。最后，观察左图，可见簇的大小差异很大，例如编号 2、6、7 的簇相比而言很小，这也符合我们的期待，即异常记录只占总数的一小部分。

接下来，我们就使用图 4.4 左图所表示的 7 个簇，研究它们的分类情况。前面挑选出了正例（认为一定是异常记录）的集合P_1, P_2以及反例（认为一定是正常记录）的集合N_1, N_2。为了考察聚类效果，同时鉴别某些簇的分类，在产生初始种子时没有使用这些集合，而是用 4.2 节描述的第二种方法。因此，使用这些正反例来做初步验证是合理的。

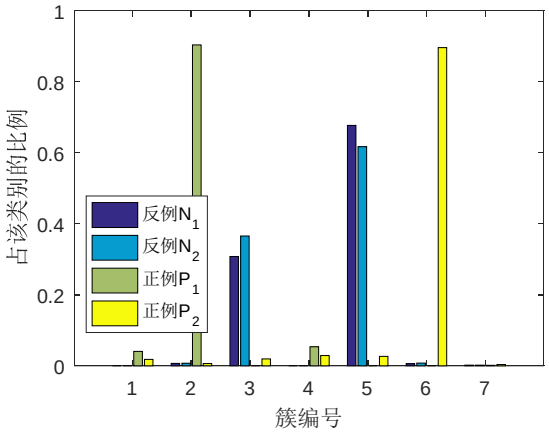

图 4.5 各簇中正例和反例的比例

因为这几个集合的元素数量不同，图 4.5 中展示了各簇当中包含的P_1, P_2和N_1, N_2占该集合的比例。从图中可以很明显地看出，簇 2、6 分别包含正例P_1和P_2而几乎不含反例，所以异常情况比较明显；簇 3 和 5 包含大部分反例，可以认为是正常簇。而剩下三个簇还需要进一步检查。

为此，列出各个簇的质心，如表 4.2。

表 4.2 各簇的质心

簇	x_1	x_2	x_3	x_4	x_5	x_6	x_7
1	0	0.0156	0.0089	0	1	0.0115	0.0707
2	0.0812	0.4603	1	0.0052	0.002	0.0206	0.0006
3	0.0618	0.0154	0	0.0025	0	0.0005	0.0003
4	0	0.8292	0.0082	0	1	0.0125	0.1244
5	0.1053	0.8301	0	0.0023	0	0.0005	0.0006
6	0.0197	0.5095	0	0.0748	0.6747	0.9581	0.0005
7	0.0872	0.9819	0.0052	0.3614	0.0960	0.0376	0.1340

簇 4 和 7 的特征x_2较大，说明它们记录中的 Wi-Fi 定位信息不可靠，因此在与地理位置相关的特征上不可靠。即使这些记录确实来自伪基站，因为我们用 Wi-Fi 定位在地图上标注伪基站可能出现的区域，它们也不能提供很多有用的信息。所以，这里不断言它们的分类，但不加入疑似伪基站的集合中。而簇 1 当中点主要异常的特征是x_5，可以将这些点记为疑似伪基站，在下一步使用。

表 4.3 记录了人工选取的正例和反例在筛选结果中的判定情况。可以看出，第一类错误和第二类错误的比例都在2%以下。考虑到数据中存在的噪声，以及选取的正例和反例当中很可能仍然存在小部分点是不属于该类的，可以认为筛选结果与人工选取的正例和反例符合良好。至于是否在全部数据上都有较高的准确率，是暂时无法检验的。

表 4.3 正例和反例的判定情况

判定结果	$P_1 \cup P_2$	$N_1 \cup N_2$
异常基站	18274 (98.1%)	7323 (1.4%)
正常基站	362 (1.9%)	518602 (98.6%)

按照上面的规则，从所有记录中筛选出35564条疑似伪基站记录，图 4.6 是绘制的北京城区的伪基站热力图。在一个区域中记录越多、越密集，则在热力图上表现为颜色深、范围大。因为数据有限，这些记录在选定的地区只有500条左右，暂时没有进行空间上的聚类。若在更全的数据集上使用这些簇为记录分类或重新进行上述步骤的聚类，应当能够得到比第 2 章的简单规则更好的结果。

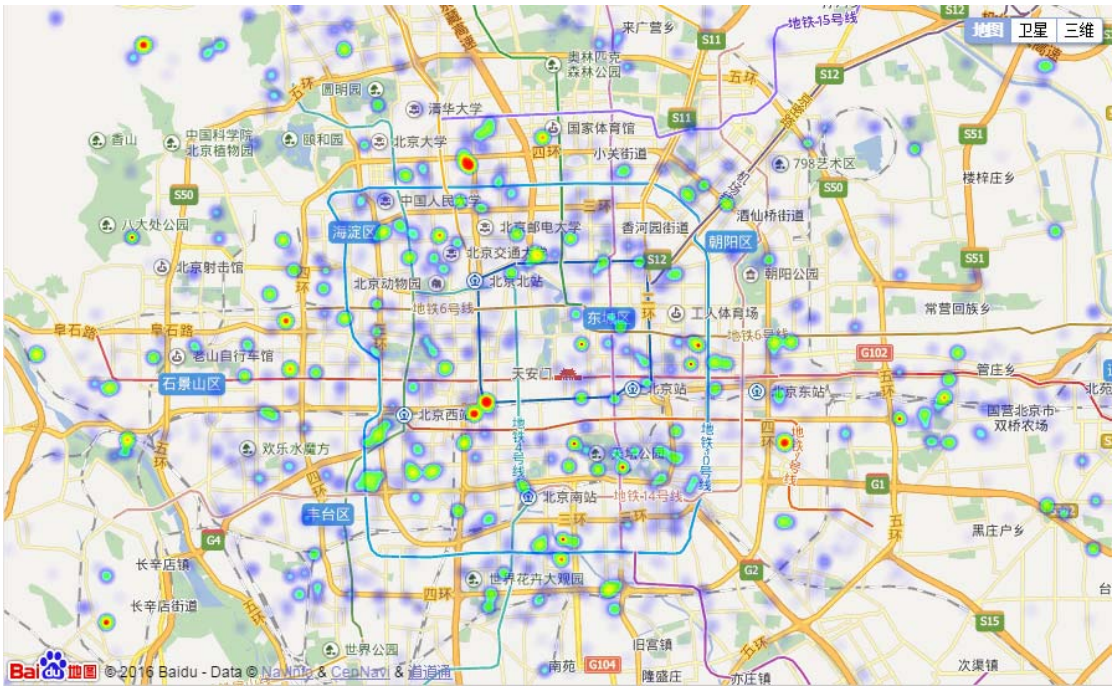

图 4.6 北京城区数据中包含的疑似伪基站的热力图

第 5 章 结论

5.1 所做工作的情况和价值

为从百度提供的伪基站数据集中检测出代表伪基站的异常记录，我们首先对数据进行描述，以及用人工设计的简单判别规则进行筛选。但是这种方法有若干不足，既有很多伪基站产生的记录无法被检测到，也没有合理的评价指标。

这是因为我们拥有的数据没有标注，即无法用常规的分类技术处理它。因此，尝试通过对每条记录的某些特征进行聚类，从而将数据分成若干相似的类，以便发现当中的某种共性。经典算法 k-means 可能得到局部解，而我们也想用对伪基站的一些现有知识来指导聚类，因此将此问题看作带有不完整先验信息、使用特定初始种子的半监督聚类问题。文章回顾了使用的改进 FAS-KMeans 算法以及相关技术细节。

此后，我们提出了大数据中的伪基站检测方法，包括提取特征、初始种子的生成等具体操作上的处理，直到最终选取疑似伪基站集合的完整方法。在样本数据集上进行实验，从 k-means 的若干内部指标找到了合适的聚类参数，它也确认了聚类具有较好的效果。利用一些原始记录中能够确认是或不是伪基站的记录，能够确定作为疑似伪基站进行下一步操作的集合。

由于样本数据量相对于全国的范围来说还比较稀疏，我们暂时没有在地图上详细地查找伪基站频繁出没的区域。如果将更大规模的数据利用本文建立的模型进行分类，或者直接用全部数据重复实验的过程，应当能对指定的区域使用 DBSCAN 算法来找到伪基站的地理分布。这样操作后，最终得到的信息能够帮助执法部门更好地检测和打击利用伪基站实施的违法犯罪行为。

5.2 存在的问题和改进方向

本研究中所使用的数据，在收集时设想的最主要依据是检查 Wi-Fi 定位和基站定位是否相符。但是，定位数据不准确、不完整，以及大多数伪基站的 id 无法查到位置，这两个重要的缺陷使上述机制无法筛选出大部分伪基站记录。本文试图利用充分记录中包含的其他几种信息，来尽量多地找出伪基站对应的异常记录。

如果百度手机卫士能够从根源上对数据包含的内容进行改进，则我们能够从数据中发现更确切的证据。例如，根据伪基站通用的工作原理，捕捉“切换到信

号强的基站——基站 id 查询不到或定位错误——无法正常联网，收到异常内容的短信——再次发生小区重选”这样的过程，比目前数据只记录“某一个时刻的状态”更有用。在这种情况下，在人工编写某些判别规则之外，我们仍然可以利用本文中描述的方法，选择某些特征、产生初始种子，对记录进行半监督聚类，从而实现异常检测。

具体到本文所使用的聚类技术，也有可以改进之处。在选择特征时，设计了多种类型的特征，使用到几乎所有的信息；但是特征的选取仍然包含设计者的一些倾向性，例如认为基站切换过快或来自权威号的异常短信等情况，是异常记录的可能性更大。同时，在将各特征归一化的过程中，将数值或逻辑变量都映射到[0,1]等区间上，对距离度量的影响不一定十分恰当。这些问题，可以在以后的工作中，调研在类似的研究中的常用处理方法等，进行更细致的调整。

此外，本文的结果目前只能通过聚类的内部指标以及事先选定的正例和反例来验证，又由于只获得了百度大数据的一个样本，无法在地图上检验伪基站的地理、时间维度分布情况。一方面，今后可以使用完整数据来进一步评价本文结果的有效性；另一方面，如果能够获得一些有标注的数据，则我们可以得到更可信的外部评价结果。为产生确定的伪基站记录，可以与有关部门合作，使用缴获的伪基站设备，获得真实的伪基站数据，用来测试本文分类的准确性。

插图索引

表格索引

参考文献

[1] 百度时代网络技术(北京)有限公司. 百度手机卫士-原安卓优化大师[EB/OL]. [2016-04-06]. http://shoujiweishi.baidu.com/.

[2] 公安部重点实验室与百度安全合作打击伪基站诈骗[J]. 互联网天地, 2015, 3: 40.

[3] Federal Communications Commission. In the Matter of Google Inc.: Notice of Apparent Liability for Forfeiture [R]. 2012-04-13, File No. EB-10-IH-4055.

[4] 尚青为. 面向移动通信安全的伪基站识别机制研究[D]. 北京邮电大学, 2015.

[5] 陈强, 刘亮. 基于智能手机的伪基站检测方法[J]. 信息安全与通信保密, 2014 (12): 131-134.

[6] Tsuda Y, Kong Q, Maekawa T. Detecting and correcting Wi-Fi positioning errors[C]//Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing. ACM, 2013: 777-786.

[7] Zandbergen P A. Accuracy of iPhone locations: A comparison of assisted GPS, Wi-Fi and cellular positioning[J]. Transactions in GIS, 2009, 13(s1): 5-25.

[8] Google Inc. Google Maps Geolocation API [EB/OL]. [2016-04-08]. https://developers.google.com/maps/documentation/geolocation.

[9] 刘锦旭, 翟晖. 浅析伪基站的主动识别与主动防御[J]. 广东通信技术, 2014, 34(2): 61-64.

[10] Exploring Indiana Highways: Trip Trivia[M]. Exploring America's Highway, 2007.

[11] Basu S, Banerjee A, Mooney R. Semi-supervised clustering by seeding[C]//In Proceedings of 19th International Conference on Machine Learning (ICML-2002).

[12] Wang C, Chen W, Yin P, et al. Semi-supervised Clustering Using Incomplete Prior Knowledge[C]//Proceedings of the 7th international conference on Computational Science, Part I: ICCS 2007. Springer-Verlag, 2007: 192-195.

[13] Lloyd S P. Least squares quantization in PCM[J]. Information Theory, IEEE Transactions on, 1982, 28(2): 129-137.

[14] Aloise D, Deshpande A, Hansen P, et al. NP-hardness of Euclidean sum-of-squares clustering[J]. Machine learning, 2009, 75(2): 245-248.

[15] Mahajan M, Nimbhorkar P, Varadarajan K. The planar k-means problem is NP-hard[M]//WALCOM: Algorithms and Computation. Springer Berlin Heidelberg, 2009: 274-285.

[16] Tan P N, Steinbach M, Kumar V. Introduction to data mining[M]. Boston: Pearson Addison Wesley, 2006.

[17] Ester M, Kriegel H P, Sander J, et al. A density-based algorithm for discovering clusters in large spatial databases with noise[C]//Kdd. 1996, 96(34): 226-231.

[18] Rousseeuw P J. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis[J]. Journal of computational and applied mathematics, 1987, 20: 53-65.

[19] Banerjee A, Davé R N. Validating clusters using the Hopkins statistic[C]//Fuzzy systems, 2004. Proceedings. 2004 IEEE international conference on. IEEE, 2004, 1: 149-153.

[20] Gan J, Tao Y. DBSCAN Revisited: Mis-Claim, Un-Fixability, and Approximation[C] //Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data. ACM, 2015: 519-530.

[21] Selim S Z, Ismail M A. K-means-type algorithms: a generalized convergence theorem and characterization of local optimality[J]. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 1984 (1): 81-87.

[22] 中华人民共和国国家工商行政管理总局. 禁止非法生产销售使用窃听窃照专用器材和“伪基站”设备的规定[Z]. 2014-12-23.

[23] 百度时代网络技术(北京)有限公司. 2015 中国互联网安全白皮书[EB/OL]. [2016-05-19]. http://shoujiweishi.baidu.com/safety.html.

[24] 王凯. 移动通信伪基站案例分析与探究[J]. 中国无线电, 2013 (6): 34-36.

[25] 孙鹏飞. 快速查找伪基站的方法研究[J]. 电信技术, 2015 (5): 31-33.

[26] 田华锋, 方莉, 杨雅玲. 基于现场实测定位的无线网络安全卫士[J]. 信息通信, 2015 (9): 228-229.

[27] 尚青为, 魏更宇. 基于数字签名的伪基站垃圾短信识别研究[J]. 软件, 2014 (12): 108-112.

[28] 黄桂泉. 3G/4G 网络能否避免“伪基站”的危害[J]. 移动通信, 2014 (z2).

[29] Fang Y, Deng Z, Xue C, et al. Application of an improved K nearest neighbor algorithm in WiFi indoor positioning[C]//China Satellite Navigation Conference (CSNC) 2015 Proceedings: Volume III. Springer Berlin Heidelberg, 2015: 517-524.

[30] Ledlie J, Park J, Curtis D, et al. Molé: a scalable, user-generated WiFi positioning engine[J]. Journal of Location Based Services, 2012, 6(2): 55-80.

[31] Zandbergen P A. Accuracy of iPhone locations: A comparison of assisted GPS, WiFi and cellular positioning[J]. Transactions in GIS, 2009, 13(s1): 5-25.

致 谢

感谢软件学院李振华老师对我毕业论文的指导。李老师为我选择了一个有趣、新颖、值得深入研究的课题，以及提供来自业界的真实数据，同时耐心聆听我每次的进展汇报，并提出指导意见。本论文的题目虽然有具体的计算机学科背景，但其中更重要的是对数据的统计建模与挖掘，而大数据背景下的数据分析，又和传统的统计学有所不同，这使得这篇论文能够兼顾我所学的专业知识与兴趣所在。我还想借此机会感谢李老师过去对我们计算机方面论文的悉心指导与修改。

感谢副导师刘思齐老师，刘老师也是我的班主任。在四年时间里，我不但在刘老师的专业实践课上收获颇多，更在诸多生活及事务上得到刘老师的帮助、照顾与关心。

我还要感谢软件学院本科生陈键同学，与我探讨本文相关数据的各方面研究内容。感谢计算机系博士生张朝昆和鄂金龙，在与他们合作科研的过程中，我学到了很多完成这篇论文需要的技能和方法。

声 明

本人郑重声明：所呈交的学位论文，是本人在导师指导下，独立进行研究工作所取得的成果。尽我所知，除文中已经注明引用的内容外，本学位论文的研究成果不包含任何他人享有著作权的内容。对本论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确方式标明。

签 名：_____ 日 期：_____

附录 A 外文资料的书面翻译

检测和改正 Wi-Fi 定位错误

摘要

最近，移动电话上的基于 GPS 和 Wi-Fi 的定位技术有所发展，这引发了很多基于位置的服务（LBS）的出现。然而，GPS 定位对手机电池的消耗很大，也不能在室内使用。另一方面，Wi-Fi 定位能实现节能的室内和室外定位，其准确率也可接受。但是，Wi-Fi 定位有时产生很大的错误，这有多种原因，例如 Wi-Fi 接入点（AP）参考点的移动。在本论文中，我们通过在时间序列中进行异常点检测，尝试自动检测和纠正这样的错误。我们将用户在时间T的测量值，与用她过去坐标的记录而预测的时间T的坐标进行比较，通过计算这两个位置的距离，来判断当前位置测量是否正确，从而解决此问题。然而，用单一的预测方法（预测器）来精确地预测用户坐标，这是困难的，因为用户所处的环境（例如移动速度，和过去坐标的分散度）在很大程度上影响预测器的性能。因此，我们通过使用一组各有优劣的预测器，来设计一种环境感知的错误检测方法。

A1 引言

最近，包括全球定位系统（GPS）和基于 Wi-Fi 的定位在内的定位技术快速发展，这引发了移动电话上很多基于位置的应用的出现。这些应用包括游客信息系统、与朋友共享位置、生命记录。我们也可以将测量的坐标作为元数据（metadata）嵌入到图片或微博（例如推文，tweet）当中。虽然 GPS 在室外提供很高精度（大约10米误差）的定位，但它有如下缺点：(1) GPS 会快速耗尽手机的电量。这对电量很少的移动电话来说是个严重问题。(2) 因为 GPS 使用 GPS 卫星，它不能在室内使用。

另一方面，虽然 Wi-Fi 定位比 GPS 稍稍不准确一些（20~40米误差），这种方法有如下优点：(1) 它的耗电量比 GPS 低很多。图 A-1 展示了手机电池的电量随时间的消耗情况。它告诉我们，GPS 比 Wi-Fi 定位消耗的电量多很多。(2) 因为 Wi-Fi 定位利用 Wi-Fi 接入点（AP）发射的信号，即使手机在室内也可以测量坐

标。(3) 这种方法能测量不带有 GPS 接收器的设备的坐标，例如笔记本电脑和某些平板电脑。最近，HTML5 中的 W3C 地理位置 API 已经确立，基于 Wi-Fi 的位置信息也在笔记本电脑的网络浏览中广泛使用。

基于以上事实，我们可以通过主要采取 Wi-Fi 定位，来减少手机的电池消耗。同时，无论手机在室内外，我们都可以获取手机的位置。这里，Wi-Fi 定位是基于指纹法技术而发展的。指纹法利用一个训练阶段，这个阶段在已知坐标测量 Wi-Fi 信号（即 AP 的唯一 MAC 地址以及来自 AP 的信号强度）。一组 AP 及其信号强度成为一个指纹，指纹唯一对应某个坐标。通过沿街扫描收集这样的指纹，然后存储在 Wi-Fi 定位数据库中。在定位阶段，在未知坐标观察到的 Wi-Fi 信号与之前存储的指纹对比，来确定最接近的匹配。

如上所述，通常 Wi-Fi 定位的误差为大约20至40米。然而，Wi-Fi 定位的误差有时比通常情况大很多（从几百米到十几千米）。这类错误通常由以下原因造成：(1) Wi-Fi AP 在不同时间的移动造成定位错误。这是因为人们带着 AP 进出公寓、家庭、办公室。同时，最近 Wi-Fi 移动热点装置激增，这些设备带有 Wi-Fi 接入点以及移动互联网接入功能，有 Wi-Fi 功能的设备可以通过这些装置连接到互联网。另外，一些高速火车也带有 Wi-Fi AP。（在下文中，我们称这些接入点为反常接入点）。因为 Wi-Fi 定位基于事先收集的指纹来确定手机的坐标，当反常设备移动时，它们会对这种定位方法有负面影响。(2) Wi-Fi 定位的精度也取决于手机附近 AP 的密度。当 AP 的数量较少时，估测的位置可能不准确。(3) 当手机附近指纹的密度较低，估测的位置可能错误，虽然这一点取决于 Wi-Fi 定位提供商的覆盖区域。

Wi-Fi 定位错误并不经常发生（在我们的实验中，当实验者在城市中行走，是测量数据中的9.8%）。但是，因为人们白天大多数时间在室内，他们的坐标通常是用 Wi-Fi 定位测量的。因此，这种错误在我们日常生活中的影响是显著的。也就是说，我们认为处理 Wi-Fi 定位错误是很重要的。在本文中，我们试图自动判断一个用 Wi-Fi 测量的坐标是否正确（即检测定位错误）。这也使我们能够实现节能的混合定位功能，即手机主要采用 Wi-Fi 定位，仅当估计 Wi-Fi 测量的坐标有很大误差时，才自动切换到 GPS。

我们可以简单地把这个问题（检测定位错误）看作时间序列当中的异常点检测问题。也就是说，通过过去的测量结果来发现一个异常测量值。我们可以利用以前的测量结果来估计当前测量值，将其与当前实测值对比，来解决此问题。当估测值与实测值偏差很大时，判断实测值为异常（错误）。假设我们通过过去的坐

标点（轨迹），用 Kalman 滤波器（它常被用来做这件事）来预测当前的坐标点。也即，我们比较滤波器预测的坐标与 Wi-Fi 定位实际测量的坐标，判断实测值是否正确。当有足够多的历史测量时，这种方法可能效果很好。然而，这种预测器有时对真实 Wi-Fi 定位数据的处理较差。例如，因为 Wi-Fi 定位不是在所有地方都能测量手机坐标的（例如在隧道中），我们有时只有很稀疏的轨迹。在这种情况下，预测值的可靠性可能很低。在下面描述的我们的实验中，我们发现超过5分钟不能测量手机坐标这种情况，每小时平均发生1.9次。同时，一些应用场景不满足连续定位的假设。比如，假设一个用户在商店里拍了一张照片，她那时的坐标嵌入到照片中。在此案例当中，她过去的轨迹无法获得，因为她手机的坐标只有在发照片或发推文时才被测量。

这里我们假设另一个预测器简单地用 Wi-Fi 扫描数据，通过向另一个 Wi-Fi 定位提供商查询，来输出坐标。即使我们没有轨迹时，这个预测器也可能输出准确的坐标。但是，当手机周围的 Wi-Fi AP 很少时，此预测器的可靠性可能较低。这种预测器也受异常 AP 的影响。如上所述，预测器的可靠性取决于手机所处的环境，例如轨迹的稀疏程度以及手机周围的 AP 数目。（在我们的实验中，我们确认不能用一个预测器在多种情况中都准确地检测 Wi-Fi 定位错误。）这就是说，我们应该设计一种环境感知的错误检测方法，它考虑环境信息。在本论文中，我们提出一种环境感知的 Wi-Fi 定位错误检测方法，它使用手机的环境数据，并结合几种不同特征的位置预测器，例如一个即使没有历史记录（过去的坐标和扫描记录）也能很好工作的预测器，以及一个当附近 AP 很少时也能表现较好的预测器。也即，我们的方法用环境感知的方法来检测异常点。用这种方法，我们可以主要使用 Wi-Fi 定位，仅当认为 Wi-Fi 测量的坐标有误时再切换到 GPS，以此节省手机的电量。但是，就像上面说的，人们日常生活中只有短暂的时间有 GPS 覆盖。此外，也有很多设备不带 GPS 接收器。在本文中，我们也设计一种环境感知的方法，它通过使用上述预测器的组合，不用 GPS，来预测正确的坐标。

本文接下来的内容中，我们首先介绍我们的数据集以及它提供的新发现。然后我们提出检测 Wi-Fi 定位错误以及估计正确坐标的方法。在那之后，我们用这个数据集来评价我们的方法。据我们所知，这是第一份试图用环境感知方法来检测和改正 Wi-Fi 定位错误的工作。

A2 数据集

A2.1 概览

若干参与者在几种情况下携带手机（Google Nexus One），手机每分钟收集 Wi-Fi 定位的坐标点（维度、精度、时间戳）。手机简单地使用原生的 Android API 来测量它们的坐标。（此 API 结合 Wi-Fi 和 GSM 数据测量坐标。）同时，手机储存一次 Wi-Fi 扫描的结果（收到的信号强度以及 AP 的 MAC 地址）。手机也连续测量 GPS 坐标点，这个数据将在实验一节当中用作真实数据（ground truth）。（我们用此数据判断一个 Wi-Fi 坐标点是否正确。）注意 GPS 坐标无法在室内获得。为了得到室内的真实数据，我们采用经验采样法（experience sampling method，ESM）。ESM 原本用来故意打断一个受试者，让她记录她当前的状况。在本实验中，当手机获得了 Wi-Fi 坐标点但不能同时获得 GPS 坐标时，每个手机上的应用程序会要求实验者在 Google 地图界面上输入她的当前位置。我们用这种方法收集了以下三种轨迹数据：

- 旅途数据集：参与者坐火车、公共汽车、和/或步行来上下班/上下学时收集这部分数据。此数据集还包括参与者出门做事时收集的轨迹。我们共获得五名参与者的11条轨迹。
- 城市数据集：一名参与者在城市中行走。这名参与者经常穿过楼房和车站，并走进商店。
- 地下数据集：一名参与者在城市的地下部分行走。

注意这些数据集中不包括两条相同的轨迹，即其中轨迹的起点和终点不重合。此外，在实验中，我们要求实验参与者随意携带手机。很多情况下，男性参与者把手机装在他们的裤兜里，女性参与者将手机放在包里。另外，当手机不能获得 GPS 坐标，我们采用 ESM。无法获得 GPS 的时间约为49%，因为我们的数据集包括“地下”数据，也就是说49%的真实数据来自 ESM。

表 A-1 提供以上数据集的概览。旅途数据集的错误率比其他集更高，因为当用户在火车或城市轨道交通上时，手机不能很好地捕捉 Wi-Fi 信号。图 A-2 页展示了 Wi-Fi 定位的误差。误差是指同时获取的 Wi-Fi 坐标和 GPS 坐标（或 ESM 回答）之间的距离。另外，图 A-3 展示了 Wi-Fi 定位的累积分布函数（cumulative distribution function，CDF）。小的误差（小于约150米）可能对应基本的 Wi-Fi 定位误差，我们希望检测比这更大的错误。（在我们的实验中，确定了我们的方法能成功检测的最小错误。）上面几幅图展示了大于150米误差的分布很广泛。此外，小错误（150到5000米）可能难以和正确测量相区分，它们占所有测量的13%。

A2.2 数据的特征

这里我们研究手机的环境怎样影响 Wi-Fi 定位误差。图 A-4 展示了当手机的坐标用 Wi-Fi 定位测量时，手机检测到的 AP 数量与不正确测量比例之间的关系。（这里我们认为误差大于150米的测量是错误的。）如图 A-4 所示，当手机周围的 Wi-Fi AP 数量较少时，Wi-Fi 定位不能精确地测量手机的坐标。

图 A-5 展示了用户（手机）的移动速度以及不正确测量比例之间的关系。（GPS 测量值包括手机的速度。）如图所示，当手机快速移动，Wi-Fi 定位方法不能测量出手机准确的坐标。这有可能是因为手机快速移动时，有时不能成功地接收来自 AP 的信号。

如简介一节中提到的，移动 Wi-Fi 热点造成 Wi-Fi 定位错误。比如，若带有移动热点的人与一个用户在相同的公共汽车或火车车厢中，热点可能会影响用户的 Wi-Fi 定位结果。在这种情况中，用户的手机连续捕捉到来自移动热点的信号。另一方面，因为手机移动很快，它不能连续捕捉来自某个固定 AP（例如放在房子或办公室当中的 AP）的信号。我们考虑五分钟的 Wi-Fi 扫描数据，记录每个 AP 在其中出现的次数。这个数字可能对移动 AP 较大，对固定的 AP 较小。图 A-6 展示了这个数字的方差以及不正确测量比例之间的关系。当方差大的时候，可能用户靠近一个移动热点。于是，错误测量的比例增加。

A3 方法：检测 Wi-Fi 定位错误

我们试图检测有上述特征的 Wi-Fi 定位错误。如简介一节提到的，我们通过使用一组具有不同特点的预测器，来尝试实现环境感知的错误检测。为了实现环境感知，我们也利用手机的环境数据，这些数据可能影响预测器以及 Wi-Fi 定位本身的表现（例如手机周围的 AP 数量）。

A3.1 问题描述

我们利用有监督的机器学习技术来检测 Wi-Fi 定位错误。假设在当前时间T，获得了手机的 Wi-Fi 坐标(x_T, y_T)，同时（时间T）手机进行一次 Wi-Fi 扫描ws_T。手机还有 Wi-Fi 坐标的历史记录$(x_1, y_1), \cdots, (x_{T-1}, y_{T-1})$，以及在每个时间片获得的 Wi-Fi 扫描记录$ws_1, \cdots ws_{T-1}$。用时间1到$T$的坐标和 Wi-Fi 扫描记录，我们将

坐标点(x_T, y_T)分到合适的类别中，即“正确”或“错误”类。注意，当我们想要使用有监督的机器学习技术时，我们需要准备对每个度量（measurement）准备一个训练信号（teaching signal）。我们通过计算 Wi-Fi 坐标和真实数据之间的距离，来从真实数据（GPS 坐标点或 ESM 回答）中获取训练信号。当距离大于给定的阈值ϵ，我们认为 Wi-Fi 坐标不正确。我们在实验一节研究阈值的大小。注意我们假设 Wi-Fi 定位提供商只提供这样一种服务，它对于一次查询（Wi-Fi 扫描）只返回(x_T, y_T)。正如现有的提供商，我们不被允许随意浏览它们非常昂贵的 Wi-Fi 指纹数据库。

A3.2 整合的错误检测方法

假设一个预测器（例如 Kalman 滤波器）估测了在时间T的坐标(xn_T, yn_T)。由此我们可以构造一个弱的错误检测器（weak error detector），它判断(x_T, y_T)是否是错误的测量，即(x_T, y_T)和(xn_T, yn_T)的距离是否大于阈值ϵ，通过以下式子：

$$h_n(p_T, pn_T) = \frac{1 - d(p_T, pn_T)/\epsilon}{1 + d(p_T, pn_t)/\epsilon}$$

其中p_T为(x_T, y_T)，pn_T为(xn_T, yn_T)，而$d(p_T, pn_T)$表示(x_T, y_T)和(xn_T, yn_T)的距离。我们展示一些例子：当距离为零，$h_n()$值为1；当距离为ϵ，$h_n()$值为0。而当距离趋于无穷，$h_n()$值趋于-1。我们可以用$C_n^{WiFi} = \text{sign}[h_n(p_T, pn_T)]$来判断$(x_T, y_T)$是否为错误的测量。也即，当$C_n^{WiFi}$为$-1$，认为$(x_T, y_T)$是错误的测量；而$C_n^{WiFi}$为$+1$时，认为$(x_T, y_T)$是正确的测量。注意，当我们想要在上面式子里用较小的阈值ϵ时，我们证实很多错误检测器总是输出-1。实际当中，在上式里用$\alpha\epsilon$（$\alpha > 1$。在我们的实现中，$\alpha = 3$而$\epsilon = 150$）代替ϵ效果好，因为很多错误比阈值大很多。

在本论文中，我们提出一种整合的错误检测架构，它能够如图 A-7 所示，考虑手机的环境。我们假设此方法在服务器计算机上运行，这一点将在相关工作一节中讨论。整合错误检测定义为以下式子：

$$H(p_T, s_T) = \text{sign}\left[\sum_{n=1}^{N} \alpha_n p(C^n = 1|s_T) h_n(p_T, pn_T)\right]$$

其中s_T为从$ws_1, \cdots ws_T$和$(x_1, y_1), \cdots, (x_T, y_T)$计算得出的手机环境特征（features），$N$为弱的错误检测器的数量，而$C^n \in \{-1, +1\}$表示第$n$个预测器的测量结果（$pn_T$）是否正确。此外，$\alpha_n$是第$n$个检测器的权重。此权重对应第$n$个检测器的错误检测

准确率，这个准确率是从训练数据用交叉验证（cross validation）方法计算得到的。另外，用$p(C^n=1|s_T)$，我们可以将手机的环境加入到第n个错误检测器中。注意$p(C^n=1|s_T)$是当给定手机的环境特征向量s_T给定时，第n个预测器给出的估计值(xn_T, yn_T)正确的概率。也就是说，我们用手机的环境来估计第n个预测器的可靠性，并把手机的环境加入到我们的整合的错误检测方法中。用贝叶斯（Bayes）定理，我们可以计算此概率如下：

$$p(C^n=1|s_T)=\frac{p(s_T|C^n=1)p(C^n=1)}{p(s_T|C^n=1)p(C^n=1)+p(s_T|C^n=-1)p(C^n=-1)}$$

其中$p(C^n)$是第n个检测器可靠性的先验密度，由训练数据中它的预测器的估计以及 GPS 真实数据计算而来；$p(s_T)$是手机的环境数据s_T（证据）的先验密度；$p(s_T|C^n)$是C^n对于s_T的可能性。我们用混合高斯模型（Gaussian Mixture Model，GMM）来对C^n建模。也即，我们准备两个模型（一个对应$C^n=1$，另一个对应$C^n=-1$），每个模型展示出对应它所在类别（1或-1）的环境特征s_T的分布。为构造此模型，我们用最大期望（expectation maximization，EM）来估计高斯模型的参数。我们可以如下计算 GMM 对于s_T的$C^n=1$（或$C^n=-1$）的概率如下：

$$p(s_T|C^n)=\sum_{\mathrm{i}}\pi_i\mathcal{N}(s_T,\mu_i,\Sigma_i)$$

其中π_i是 GMM 的第i个多变量高斯分布的混合权重；μ_i和Σ_i对应是高斯分布的均值向量以及协方差阵。

如上所述，我们融合各个弱的错误检测器的结果，结合从环境中计算出它们的可靠性，来检测 Wi-Fi 定位的错误。当$H(p_T,s_T)$为-1，我们可以说p_T是错误的测量（“错误”类）。这里我们考虑s_T（特别是ws_T）也直接与p_T直接相关。因此，在下一小节中描述的普通错误检测器之外，我们准备了一个特别的错误检测器，它利用以下式子，并且我们把它加入整合的错误检测器$H(p_T,s_T)$，如同普通的检测器一样：

$$h_{WiFi}(p_t,s_t)=2p\left(C^{WiFi}=1\middle|s_T\right)-1$$

其中$C^{WiFi}\in\{-1,+1\}$表示 Wi-Fi 测量p_T是否正确。我们基于 GMM 方法估计$p\left(C^{WiFi}=1\middle|s_T\right)$，用和计算$p(C^n=1|s_T)$的相同方法。（当我们训练 GMM 时，我们可以从 GPS 的真实数据知道训练集中的C^{WiFi}是1或-1。）这个弱的检测器让我们能够仅从手机的环境来估计p_T的可靠性。

接下来，我们描述准备的（正常）预测器，以及环境特征。

A3.3 预测器

我们准备了若干具有不同特征的预测器，来应对手机的各种环境。我们介绍我们在本研究中准备的预测器。

A3.3.1 Kalman 滤波器

此预测器采用 Kalman 滤波器，它通常用来追踪运动物体的轨迹。Kalman 滤波器基于线性系统，它的算法以两步过程工作：预测和更正。在预测阶段，滤波器基于前一时间步（时间$T-1$）的状态变量，来估计当前的状态变量（在我们的情况中是(xn_T, yn_T)）。在本次实现中，我们简单地假设手机的运动速度与前一时间步相同。在更正阶段，用实际测量值(x_T, y_T)来更新估计得变量。我们采用预测阶段所估计的变量(xn_T, yn_T)作为此预测器所估测的坐标点。当手机有足够多的历史坐标时，此预测器可能工作效果好。

A3.3.2 改进的 Kalman 滤波器

我们的数据有时包括很大的误差，也就是说，测量值和实际坐标点之间的距离可能很大。因为 Kalman 滤波器以递归方式工作，即预测和改正这两个过程在每个时间步重复，一个较大的误差可能很大地影响之后的预测。我们通过简单地忽略那些距离上个更正后的坐标点太远的测量值，来应对此问题。也即，这个滤波器在改正其预测值时，不考虑这样的测量结果。

A3.3.3 粒子滤波器（particle filter）

假设一个人走出汽车并步行。当她离开车子时，她的移动速度可能突然改变。为追踪这样的轨迹，我们采用粒子滤波器，它通常用来估测非线性系统的状态。它的算法以三步过程工作：采样、计算权重、重采样。在采样过程中，新粒子从前一个时间片（$T-1$）的粒子产生，并基于一个动态模型移动。产生的粒子显示对于时间T坐标的估计。在计算权重过程中，基于重要度函数和时间T的测量值来计算粒子的权重。基本上，靠近测量值并且与重要度函数匹配的粒子有较大的权重。在重采样过程中，权重小的粒子被删除掉。这三个过程迭代重复。此预测器输出采样过程中粒子的平均值，作为估计的坐标点。在采样过程中，我们采用一个双变量的高斯分布作为动态模型，它的平均值对应在时间T外推得到的坐标点，这个坐标点的来自每个粒子在时间$T-1$的速度和坐标。同时，它的标准差对应时间时间$T-1$的平均坐标点和粒子坐标之间的距离，协方差为零。用这个滤波器，

我们试图追踪速度突然改变的手机的运动。

A3.3.4 第二意见

此预测器简单地用ws_T查询另一个 Wi-Fi 位置提供商。在我们的实现中使用 Skyhook 位置服务。即使手机没有过去的坐标时，这个追踪器也能工作良好。然而，当手机周围 AP 数少时，它可能不会输出准确的坐标。此外，它受异常 AP 的影响。

A3.3.5 查询修改

当 Wi-Fi 扫描结果包括来自移动 AP 的信号，它们可能对 Wi-Fi 定位的位置估计有负面影响。此预测器试图从 Wi-Fi 扫描数据ws_T中移除移动 AP 的信号，然后用修改过的扫描数据向 Wi-Fi 定位服务（Google 位置服务）查询。此预测器基于这样的假设设计，即一个 Wi-Fi 扫描当中包括少数移动 AP。它包括四个步骤。我们用例子来解释这个方法，算法也在算法 1 中给出。假设 Wi-Fi 扫描ws_T包括来自六个 Wi-Fi AP 的信号，其中一个是移动 AP。

1) 此方法将六个 AP 随机划分到g个小组中。（g是偶数。本例中，我们假设g为3。注意一个小组中的元素数量不超过$\left\lceil \frac{ws_T\text{中}AP\text{数}}{g} \right\rceil$。
2) 此方法用每个小组中的 Wi-Fi 扫描数据，来向 Wi-Fi 定位提供商（Google Geolocation API）查询。也就是说，比如当 AP1 和 AP2 包含在一个小组中，该方法用 AP1 和 AP2 的信号扫描数据来查询。
3) 从三个小组中获取三个数据点（坐标）。因为我们假设ws_T中有一个移动 AP，三个点当中有一个可能有很大的误差。（此数据点来自那个包含移动 AP 的小组。）也即它可能远离另外两点。此方法简单地用凝聚层次聚类（自底向上聚类）（agglomerative hierarchical clustering，bottom-up clustering）。那么，我们首先将每个数据点（坐标）放在一个簇（cluster）里，然后迭代地合并两个最近的簇。为定义簇之间的相似性，我们使用最短距离法（single-linkage method）：

$$d(c_1, c_2) = \min_{(x_1,y_1)\in c_1,(x_2,y_2)\in c_2} d((x_1, y_1), (x_2, y_2))$$

其中c_1和c_2对应簇，(x_1, y_1)和(x_2, y_2)对应数据点，$d((x_1, y_1), (x_2, y_2))$表示两个数据点之间的欧几里得距离。注意，当最大簇中的数据点数量超过$\lfloor g/2 \rfloor$时停止合并簇。通过这样做，我们可以将彼此较近的固定 AP 组合

在一起。（也就是说，最大的簇不包含移动 AP。）

4) 此方法关注最大的簇，并用此簇当中的 AP 的 Wi-Fi 扫描数据，来向 Wi-Fi 定位提供商查询。该预测器输出这个查询得到的数据点（坐标点）。

注意在我们实现的步骤 2 和 4 当中使用的 Google 位置服务（Google Geolocation API），它不接受只包含一个 AP 的 Wi-Fi 扫描数据。那么，例如当g为三，而ws_T当中包含的 AP 数量小于六，第一步中创建的若干个小组可能只包括一个 AP。在这种情况下，我们应该增加小组里的 AP 数量。也就是说，我们添加从ws_T中随机选择的 AP（的信号数据）到每组中，直到组中都有两个 AP。注意我们不选择组里已有的 AP。此外，当ws_T里的 AP 数量小于四时，此预测器不输出任何结果。当ws_T中包含数量很多的 AP 时，即使ws_T中有小部分移动 AP，此预测器也可能工作得很好。然而，当ws_T中的 AP 数量较小时，它的效果可能不好。我们假设g的值是3和5，并对每个g准备一个这样的预测器。

A3.4 环境特征

我们准备以下环境特征，来捕捉手机的环境信息，并构造包含特征数值的向量（s_T）。

- 时间间隔：此特征对应ws_{T-1}和ws_T之间的时间间隔。当手机很长时间不能获得它的坐标时，这个时间间隔会很大。那么，某几个预测器（Kalman 和粒子滤波器）的表现会变差。
- 移动速度和距离：我们采用由(x_{T-1}, y_{T-1})和(x_T, y_T)计算的手机移动速度作为一个特征。我们也采用(x_{T-1}, y_{T-1})和(x_T, y_T)的距离。当这些数字很大，时间T的坐标点可能不正确。
- 速度的方差和均值：我们关注最后的n个坐标$(x_{T-n+1}, y_{T-n+1}), \cdots, (x_T, y_T)$。我们用每对相邻的坐标点（例如$(x_t, y_t)$和$(x_{t+1}, y_{t+1})$）来计算手机的速度，采用速度的方差作为一个特征。当手机的移动速度变化时，这个数值变大。此时，基于 Kalman 滤波器的预测器的表现会变坏。我们也采用最后n个坐标的平均速度。
- AP 的数量：此特征对应ws_T包含的 AP 数量。几个预测器（“第二意见”、“查询修改”以及使用 Google Android API 的“Wi-Fi 定位”）与这个特征的数值有关。
- 信号强度的均值和方差：我们采用ws_T中所包含 AP 的平均信号强度作为

一个特征。这个特征可能与手机周围的信号质量相关。我们也采用信号强度的方差。

- 和移动热点相关的特征：例如，若一个带着移动热点的人与手机用户乘坐同一公共汽车或火车车厢，热点可能影响手机的 Wi-Fi 定位结果。在这种情况下，手机连续捕捉来自移动热点的信号。另一方面，因为手机很快地移动，它无法检测来自静态 AP（即放置在房屋或办公室中的 AP）在时间T的信号，这个 AP 在时间$T-1$曾被检测到。我们计算在时间$T-1$和T都检测到的 AP 数，比上时间$T-1$和T检测到的 AP 总数。我们采用计算得到的数值作为一个特征。我们还从$ws_{T-n-1},\cdots ws_T$计算一个与移动热点有关的特征。对每个 AP，数包含该 AP 的扫描的数目。我们采用这个数字的方差作为一个特征。当带移动热点的人与手机用户同乘公共汽车或火车车厢时，该值变大，如图 A-6 所示。

在我们关于环境特征提取的实现中，设置n为五。注意 Wi-Fi 位置提供商也提供估计的定位误差。然而，因为它在我们的研究中不可靠，我们不把它用作特征。

A4 方法：更正 Wi-Fi 坐标

当我们判断一个手机的坐标点不正确，如果手机在户外，我们可以打开 GPS 接收器。当手机在室内时，我们应该不用 GPS 来估测它正确的坐标。这里简述我们的错误更正方法，因为这篇论文主要关注检测 Wi-Fi 定位错误。图 A-8 展示了我们方法的概览。这种方法中，我们采用为错误检测方法而准备的那些预测器。我们也为每个预测器准备一个错误估计器。估计器通过使用环境特征小节中计算出的环境特征，来估测对应预测器的误差（实际坐标点与预测器估计的坐标之间的距离），因为我们认为预测器的可靠性与手机的环境有关。我们用序列最小优化（sequential minimal optimization，SMO）进行回归，来构造每个错误估计器。如同错误检测那样，我们用 GPS 和 ESM 真实数据来训练估计器。注意在训练时，误差最大设为1000米，以防止回归模型对于大的误差发生过拟合（overfitting）。

每个预测器输出手机的坐标(xn_T, yn_T)。对应的估计器输出时间T的坐标的误差。我们假设误差的逆rn_T表示预测坐标的可靠性。利用预测的坐标和它们的可靠性，我们计算坐标的加权平均为

$$(\hat{x}_T, \hat{y}_T) = \frac{\sum_{n=1}^{N} w_n rn_T(xn_T, yn_T)}{\sum_{n=1}^{N} w_n rn_T}$$

注意w_n是第n个预测器的权重，它对应于第n个预测器的的平均准确度（150米内的精度），从训练数据计算得来。计算得到的$(\hat{x}_T, \hat{y}_T)$对应此方法的输出。

A5 评价

A5.1 评价方法：检测 Wi-Fi 定位错误

我们采用弃一交叉验证（leave-one-session-out cross validation），来评估错误检测方法。（一次运行对应在数据集一节当中获得的一条轨迹。）也就是说，我们用其他三十条轨迹训练的错误检测模型（分类器），来检测一条轨迹当中的错误。为研究我们提出的方法的有效性，在我们的方法之外测试了以下简单的方法：

- Kalman 滤波器：此方法简单地构造一个分类器（C4.5 决策树，decision tree），它使用$h_n()$作为一个特征，此值是由 Kalman 滤波器估计的(x_T, y_T)和在时间T的坐标计算得来。它将特征分入正确或错误类别。以下的方法也使用在预测器一节中准备的预测器。
- 改进的 Kalman 滤波器
- 粒子滤波器
- 第二意见
- 修改的查询（$g = 3$或5）
- 环境：此方法简单地用环境特征构造一个 C4.5 决策树分类器。

我们在每个时间片，用分类结果计算得到的精度（precision）、召回率（recall）、F-度量（F-measure），来评估这些方法的分类效果。（$F\_measure = \frac{2 \cdot precision \cdot recall}{precision + recall}$）

A5.2 结果：检测 Wi-Fi 定位错误

A5.2.1 简单方法的结果

图 A-9 展示了每个方法获得的正确和错误结果的平均 F-度量。（三个数据集的平均值。）这里我们假设误差大于150米的 Wi-Fi 坐标是错误的。我们首先关注 Kalman 滤波器、改进的 Kalman 滤波器、粒子滤波器，它们使用过去的轨迹。用

Kalman 滤波器方法的结果非常差（44.7%）。这可能是因为“不正确”的点比“正确”的少，所以分类器可能会忽略“错误”的实例。为解决这一问题，我们采取局部权重学习（locally weighted learning，LWL），其中错误的实例比正确的实例权重大三倍。结果也在图 A-9 中展现。此外，图 A-10 展示每个数据集的分类准确率。如图所示，在三种方法当中，粒子滤波器最好，并在旅途数据集结果更好。这可能是因为即使当速度经常改变，这种方法也能追踪手机。（当一个人乘坐火车、公共汽车或地铁时，她的速度经常改变。）这里我们研究为什么这些方法不能在旅途以及地下数据集上达到很好的精度。我们发现有时候这些方法不能准确地追踪参与者，例如因为她在隧道或地下轨道交通中，而坐标长时间不能测量之时。在这种情况下，她真实的坐标与估测坐标之间的距离变大（几百米）。因为我们假设错误的 Wi-Fi 测量的误差大于150米，因此这三种方法很难区分真的和假的错误测量。

第二意见方法在图 A-9 中取得了不错的准确率。但是，当参与者穿过大的区域时，例如在旅途和城市数据集中的大学、公园、车站，这种方法通常不能准确地预测她的坐标。在第二意见方法中，我们使用 Skyhook 位置服务。因为它的 Wi-Fi 定位数据库中的指纹可能用沿街扫描技术收集而来，远离道路的坐标估计不是很准。因此，在这样的大的区域当中的识别结果不好。另一方面，当参与者在火车上，这种方法获得了好的精度。

与我们的期待相反，对于旅途数据集，修改查询方法获得的结果不好。这是因为 Wi-Fi 定位错误通常发生在ws_T中的 AP 数量较小时，如图 A-4 所示。这种情形在旅途数据集中频繁出现。修改查询方法在没有足够数量的 AP 时不能准确地预测坐标。但是，我们确认了当ws_T中的 AP 数量较大，这种方法能够准确地预测坐标。这可能反映了地下数据集相关的结果，修改查询方法在相关结果中达到较好的精度。我们进行实验的城市地下空间很拥挤，人们携带的移动 AP 可能影响 Wi-Fi 定位。另一方面，地下商店、咖啡店、餐馆可能有很多 AP，它们可以被修改查询方法利用

此外，如同环境方法的结果显示的那样，很难只用环境数据来检测定位错误。注意我们证实了当误差阈值为500米时，环境方法达到了70%的准确率。因此，我们可以大体上只用环境数据来检测定位误差。

A5.2.2 我们的方法的结果

图 A-9 和 A10 展示了我们的方法获得的结果，它比其他方法表现好很多（不

用 LWL 时为86.0%）。我们方法的平均 F-度量大约比粒子滤波器好15%，尽管后者使用目前最好的追踪技术。像上面提到的，每个简单方法有各自的优缺点，它们取决于手机的环境。在我们的方法中，准备的预测器可能根据收集的环境互相弥补弱点。例如，如上所述，当一个参与者走过大的区域时，第二意见方法不能很好地工作。另一方面，Kalman 滤波器和粒子滤波器方法能成功地在这种情况中追踪参与者。因为我们的方法能通过参与者的环境特征（例如移动速度）知道她是否在行走，所以它能适应地采用各预测器的意见。我们认为这在图 A-10 的结果中得到反映，其中我们的方法对于每个数据集都得到好的结果。此外，我们的方法在 GPS 不可用的地下数据集中也达到了好的精度。这个结果表明我们可以在无法使用 GPS 查询的地方检测 Wi-Fi 定位错误。

这里我们研究在 Wi-Fi 错误检测当中，加入环境数据的效果。也就是说，我们计算在我们的方法中不使用 GMM 时的分类准确率，GMM 是用来处理环境数据的。此时准确率是79.3%，比我们的标准方法差了约7%。特别地，与城市和地下数据集相关的准确率下降了。我们使用基于信息增益的特征分析（information gain based feature analysis），发现环境数据中的移动速度特征对分类任务贡献很大。因此，我们考虑，因为数据集中的参与者几乎以固定的速度移动，此特征能容易地捕捉到异常情况（定位错误）。

此外，我们根据错误测量的距离误差，研究我们怎样正确地检测它们。图 A-11 展示了错误测量的误差距离对于分类准确率的累积值。尽管只用 Wi-Fi 相关的数据可能很难找到发现小的误差（150~500米），我们的方法能够以约75%的准确率检测（分类）这些错误。

A5.2.3 历史记录较少时的分类

上面的结果是使用所有可用的历史记录（一次运行中所有过去的坐标和 Wi-Fi 扫描结果）计算得到的。这里我们研究为检测错误，需要历史记录的程度。图 A-12 展示了当我们更改使用的历史记录数量时（20个坐标点到1个），平均 F-度量的变化。当我们只有一项历史记录，精度很差。这可能是因为只用一个历史测量值时，我们不能用若干个预测器（Kalman 和粒子滤波器）和环境特征。另一方面，当我们用超过一个坐标点，可以达到好的精度，因为我们可以应用强大的预测器。注意，当历史记录少的时候，精度不稳定，因为预测器的估计值也不稳定。

A5.2.4 错误检测的敏感性

与错误类关联的 F-度量（77.0%）比正确类的（92.4%）差，这可能是因为在训练数据中，错误的实例少于正确的实例。另一方面，一个二类分类器（binary classifier）计算一个实例的分数，这分数显示出这个实例属于某一类的程度。我们可以对这个分数使用阈值，来做出二类选择（binary decision）。通过改变这个阈值，我们获得了图 A-13 所示的接收者操作特征曲线（ROC curve）。真阳性率（true positive rate）对应所有错误实例当中被正确分类为错误实例的比例，而假阳性率（false positive rate）是所有正确实例中被错误标记的正确实例的比例。通过改变阈值，我们可以控制错误检测的敏感性。如图所示，当我们想要以约95%的真阳性率寻找错误的实例，假阳性率大约为30%。

A5.3 评价方法：更正 Wi-Fi 坐标

和上面的评价一样，我们采用弃一交叉验证来评价我们的错误更正方法。我们将我们的方法估计的坐标，与 Wi-Fi 定位获得的原始坐标相比较。注意我们对于误差大于150米的 Wi-Fi 坐标应用此方法。

A5.4 结果：更正 Wi-Fi 定位错误

图 A-14 展示了原始 Wi-Fi 定位和我们方法的累积分布函数（CDF）。因为我们只关注误差大于150米的 Wi-Fi 坐标，原始 Wi-Fi 定位的误差小于150米部分的累积概率为零。我们的方法比原始 Wi-Fi 定位好很多，在150米之内达到47%的精度。此外，我们的方法在163米达到50%精度。另一方面，原始的 Wi-Fi 定位在26786米达到50%精度。图 A-14 也展示了粒子滤波器估计器的 CDF，它在6733米达到50%精度。当观察到连续的 Wi-Fi 定位错误，或者定位错误发生在一次运行（轨迹）的开始时，粒子滤波器输出非常大的误差。

在用我们的方法获得的结果中，改进的 Kalman 滤波器和粒子滤波器预测器贡献很大（特别在城市数据集中）。另一方面，我们发现 Kalman 滤波器通常不贡献，因为如预测器一节中所说，此预测器受错误测量的影响很大。同时，我们发现第二意见预测器也有贡献，它是使用 Skyhook 位置提供商的。我们认为 Skyhook 和 Google 覆盖的区域有些不同，前者可以弥补后者。但是，我们发现在很多情况中，这个预测器因为覆盖区域问题而不输出结果。进一步，我们发现修改查询预测器的错误估计器有时表现较差。当 Google 的 Wi-Fi 定位数据库中包含的ws_T中

的 AP 数量较小时，此预测器（和预测器中使用的 Google 位置服务）似乎输出误差很大的坐标。然而，很难用我们准备的环境特征来检测这种情形。

A6 结论

在本论文中，我们提出检测和改正 Wi-Fi 定位错误的方法。因为 Wi-Fi 定位错误在多种情形中发生，我们通过使用一组不同特征的位置预测器来应对它们。在我们用真实 Wi-Fi 坐标数据的实验评价中，我们的方法实现了很准确的错误检测，并且比简单方法的表现好很多。

书面翻译对应的原文索引

[1] Tsuda Y, Kong Q, Maekawa T. Detecting and correcting Wi-Fi positioning errors[C]//Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing. ACM, 2013: 777-786.