

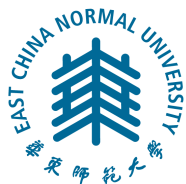
2018 届博士学位论文

分类号: \_\_\_\_\_

学校代码: 10269

密 级: \_\_\_\_\_

学 号: 52141500013



華東師範大學

East China Normal University

博 士 学 位 论 文

DOCTORAL DISSERTATION

论文题目: 分布式 k 近邻轨迹查询

院 系: 数据科学与工程学院

专 业 名 称: 软件工程

研 究 方 向: 基于位置的服务

指 导 教 师: 周傲英 教授

学位申请人: 章志刚

2018 年 05 月



Dissertation for doctor degree in 2018

University Code: 10269

Student ID: 52141500013

EAST CHINA NORMAL UNIVERSITY

# **$k$ Nearest Neighbour Query over Distributed Trajectories**

|                     |  |
|---------------------|--|
| Department:         | School of Data Science and Engineering |
|                     | Software Engineering                   |
| Major:              | Location Based Service                 |
| Research direction: | Prof. Aoying Zhou                      |
| Supervisor:         | Zhigang Zhang                          |
| Candidate:          |  |

2017.05



## 华东师范大学学位论文原创性声明

郑重声明：本人呈交的学位论文《基于矩阵分解的个性化推荐系统》，是在华东师范大学攻读硕士/博士（请勾选）学位期间，在导师的指导下进行的研究工作及取得的研究成果。除文中已经注明引用的内容外，本论文不包含其他个人已经发表或撰写过的研究成果。对本文的研究做出重要贡献的个人和集体，均已在文中作了明确说明并表示谢意。

作者签名：\_\_\_\_\_

日期： 年 月 日

## 华东师范大学学位论文著作权使用声明

《基于矩阵分解的个性化推荐系统》系本人在华东师范大学攻读学位期间在导师指导下完成的硕士/博士（请勾选）学位论文，本论文的研究成果归华东师范大学所有。本人同意华东师范大学根据相关规定保留和使用此学位论文，并向主管部门和相关机构如国家图书馆、中信所和“知网”送交学位论文的印刷版和电子版；允许学位论文进入华东师范大学图书馆及数据库被查阅、借阅；同意学校将学位论文加入全国博士、硕士学位论文共建单位数据库进行检索，将学位论文的标题和摘要汇编出版，采用影印、缩印或者其它方式合理复制学位论文。

本学位论文属于（请勾选）

- ☐ 1. 经华东师范大学相关部门审查核定的“内部”或“涉密”学位论文\*，于年月日解密，解密后适用上述授权。
- ☐ 2. 不保密，适用上述授权。

导师签名：\_\_\_\_\_

本人签名：\_\_\_\_\_

年 月 日

\*“涉密”学位论文应是已经华东师范大学学位评定委员会办公室或保密委员会审定过的学位论文（需附获批的《华东师范大学研究生申请学位论文“涉密”审批表》方为有效），未经上述部门审定的学位论文均为公开学位论文。此声明栏不填写的，默认为公开学位论文，均适用上述授权）。



章志刚 博士学位论文答辩委员会成员名单

| 姓名  | 职称   | 单位     | 备注 |
|-----|------|--------|----|
| 吴悦  | 教授   | 上海大学   | 主席 |
| 阚海斌 | 教授   | 复旦大学   |    |
| 张娅  | 研究员  | 上海交通大学 |    |
| 郑凯  | 特聘教授 | 苏州大学   |    |
| 何晓丰 | 研究员  | 华东师范大学 |    |





## 摘 要

随着信息技术和互联网的发展,网民用户和网络产品数量成爆炸式增长,用户从信息匮乏时代进入信息过载时代。个性化推荐系统对用户行为和企业商品特性数据建模,为用户提供满足他们兴趣和需求的信息,同时为企业推广提供目标客户。现代互联网服务提供商,例如淘宝等在线购物网站、爱奇艺等在线视频网站、大众点评等生活信息服务网站,提供大量商品给用户消费,让用户给商品评分以及使用标签描述商品。针对以上用户行为数据,本文以矩阵分解相关理论为基础,建立模型以解决推荐系统中的三个典型任务:(1)在隐式反馈数据上构建商品推荐模型,推荐用户感兴趣的商品;(2)利用显式评分数据建立评分预测模型,预测用户对商品的喜好值;(3)利用显式标签数据建立标签推荐模型,方便用户输入来描述商品属性,帮助推荐系统良性循环。本文的研究问题和技术贡献总结如下:

1. 基于加权局部矩阵分解的商品推荐:现有的针对隐式反馈数据的矩阵分解模型往往只从数据的全局信息出发,忽略了数据之中的局部信息。为了利用隐式反馈数据的局部信息,本文提出了一种加权局部矩阵分解模型进行商品推荐,并为该模型设计了高效的子矩阵选择算法和改进的交替最小二乘参数优化算法,对用户和商品的局部特性建模,同时缓解了数据稀疏性问题。真实数据上的实验结果表明该模型有较优的推荐效果,并验证了考虑隐式反馈数据的局部信息有助于商品推荐。
2. 基于多主题矩阵分解的评分预测:为克服现有工作中针对显式评分数据局部信息建模的不可解释性和目标函数的不一致性,本文提出了多主题矩阵分解模型。它结合主题模型和概率矩阵分解模型,利用主题模型建模数据局部信息和矩阵分解来刻画用户和商品的局部内在特征。本文使用贝叶斯方法建模主题矩阵分解模型,使得模型只需少量的经验设置参数以得到更高的推荐准确率。实验结果说明该模型在评分预测中优于其他局部矩阵分解模型,并对局部建模信息具有一定的可解释性。
3. 时间感知的标签推荐:为了利用标签数据中用户标注标签的时间信息,本文提出了时间感知的张量分解模型。该模型利用 Hawkes 时间点过程对用户使用标签的时间信息建模,并利用指数函数将 Hawkes 过程中的叠加形式转化为递归形式,使得计算用户当前时间对标签的喜好值只跟上次使用标签的时间有关,减少了大量的计算时间,并将其以权重的方式加入到逐对排序张量

分解模型。实验结果表明该模型能够有效地利用时间信息，提高标签准确度，同时在冷启动问题上也有较好的表现，并且具有可接受的推荐新颖性。

**关键词:** 矩阵分解，局部信息，主题模型，商品推荐，评分预测，标签推荐

## ABSTRACT

With the development of information technology, the numbers of Internet users and network products show explosive growth. So the world has transformed from the time of lack of information to the information overload era. Personalized recommendation system builds models utilizing users' behavior data and characterization data of enterprise product, then provides target information for users and offers target customers for business promotion. Modern Internet service providers, such as online shopping site Taobao, online video site iQIYI, and life information service website Dianping, provide users with a large number of goods, allowing users to rate goods and describe goods with tags. Focusing on the above user behavior data, this thesis aims at tackling three typical recommendation tasks based on the matrix factorization techniques: (1) building *item recommendation* model on implicit feedback datasets to provide target items for users; (2) building *rating prediction* model on explicit rating datasets to predict the user's preference for the item; (3) building *tag recommendation* model on explicit tag datasets to make tagging input convenient for users to describe items, which helps the positive cycle of recommendation system. The research questions and technical contributions in this thesis can be summarized as follows,

1. Local Weighted Matrix Factorization for *item recommendation*: the existing matrix factorization models on implicit feedback datasets only consider the global property of data and ignore the local property. To utilize the local property in implicit feedback datasets, this thesis proposes Local Weighted Matrix Factorization to recommend items. To learn the parameters, this thesis designs an efficient sub-matrix selection algorithm and an improved Alternating Least Square optimization algorithm. LWMF models the local property of users and items and relieves the data sparsity problem. The experimental results on real datasets show that LWMF has relatively good recommendation performance and verify that considering local property of

implicit feedback is helpful for item recommendation.

2. Probabilistic Multi-Topic Matrix Factorization for *rating prediction*: there are two main weak points in previous studies. One is the non-interpretability of the models built on the local information in explicit rating data. Another is the inconsistency of the objective functions. To overcome these problems, this thesis presents a Probabilistic Multi-Topic Matrix Factorization model. This model combines topic model with probabilistic matrix factorization model. Topic model is used to capture local information of data and matrix factorization models the local inner property of users and items. Furthermore, this thesis extends a Bayesian formulation of probabilistic multi-topic matrix factorization model. It requires fewer efforts in parameter selection and can achieve higher recommendation accuracy. Extensive experiments demonstrate the effectiveness of the proposed model compared with several competitive baselines and the interpretability to local modelling information.
3. Time aware *tag recommendation*: To utilize the time when users use tags, this thesis presents a time aware tensor factorization model. This model utilizes Hawkes process to model the temporal information in users' tagging behavior. Specifically, exponential function is used to transform accumulation to recursion form for Hawkes process, which makes that target user's preference to tags at current time only depends on the last time this user used tag. Then this model incorporates temporal point process into pairwise tensor factorization model by modeling it as weight. The experimental results show that this model can utilize time information effectively and can improve the accuracy of tag recommendation. In addition, this model performs well in the cold start situation and can achieve good quality to recommend new tags.

**Keywords:** *Matrix Factorization; Local Information; Topic Model; Item Recommendation; Rating Prediction; Tag Recommendation.*

# 目录

|                                |    |
|--------------------------------|----|
| 第一章 绪论 . . . . .               | 1  |
| 1.1 研究背景 . . . . .             | 1  |
| 1.2 研究内容与挑战 . . . . .          | 4  |
| 1.3 主要贡献 . . . . .             | 5  |
| 1.4 章节安排 . . . . .             | 6  |
| 第二章 问题定义及研究现状 . . . . .        | 9  |
| 2.1 问题定义 . . . . .             | 9  |
| 2.1.1 轨迹数据 . . . . .           | 9  |
| 2.1.2 分布式 $k$ 近邻轨迹查询 . . . . . | 10 |
| 2.2 轨迹索引 . . . . .             | 11 |
| 2.3 轨迹降维 . . . . .             | 11 |
| 2.4 轨迹相似度度量 . . . . .          | 11 |
| 2.5 时间序列 top- $k$ 查询 . . . . . | 11 |
| 第三章 查询处理框架 . . . . .           | 13 |
| 3.1 引言 . . . . .               | 13 |
| 3.2 接口函数 . . . . .             | 14 |
| 3.3 FTB 框架 . . . . .           | 15 |
| 3.3.1 FTB 框架设计原理 . . . . .     | 15 |
| 3.3.2 FTB 框架实现 . . . . .       | 17 |
| 3.4 FLB 框架 . . . . .           | 20 |
| 3.4.1 FLB 框架设计原理 . . . . .     | 20 |
| 3.4.2 FLB 框架实现 . . . . .       | 22 |
| 3.5 本章小结 . . . . .             | 24 |

|       |                       |    |
|-------|-----------------------|----|
| 第四章   | FTB 框架的应用             | 26 |
| 4.1   | 基于欧式距离的概要数据           | 26 |
| 4.1.1 | 基于欧式距离的轨迹相似度度量        | 26 |
| 4.1.2 | 基于哈尔小波的轨迹概要数据抽取       | 27 |
| 4.2   | 轨迹欧式距离上下界             | 28 |
| 4.2.1 | 基于哈尔小波的欧式距离表示         | 28 |
| 4.2.2 | 基于哈尔小波的欧式距离上下界        | 29 |
| 4.3   | 基于欧式距离的查询算法:ED-FTB    | 31 |
| 4.3.1 | ED-FTB 算法实现           | 31 |
| 4.3.2 | ED-FTB 算法性能分析         | 34 |
| 4.4   | 实验分析                  | 35 |
| 4.4.1 | 实验设置                  | 35 |
| 4.4.2 | 算法有效性                 | 36 |
| 4.4.3 | 算法可扩展性                | 38 |
| 4.5   | 本章小结                  | 39 |
| 4.6   | 附件                    | 40 |
| 第五章   | FLB 框架的应用             | 44 |
| 5.1   | 基于动态时间卷曲距离的概要数据       | 44 |
| 5.1.1 | 基于动态时间卷曲距离的轨迹相似度度量    | 44 |
| 5.1.2 | 基于包围信封的概要数据           | 46 |
| 5.2   | 动态时间卷曲距离的下界           | 47 |
| 5.2.1 | 满足 DTW 约束的包围信封及下界     | 47 |
| 5.2.2 | 基于多粒度包围信封的下界          | 48 |
| 5.3   | 基于动态时间距离的查询算法:DTW-FLB | 49 |
| 5.3.1 | DTW-FLB 算法实现          | 49 |
| 5.3.2 | DTW-FLB 算法性能分析        | 51 |
| 5.4   | 实验分析                  | 52 |
| 5.4.1 | 实验设置                  | 52 |

|           |        |    |
|-----------|--------|----|
| 5.4.2     | 算法有效性  | 53 |
| 5.4.3     | 算法可扩展性 | 55 |
| 5.5       | 本章小结   | 56 |
| 5.6       | 附件     | 57 |
| 参考文献      |        | 63 |
| 附录        |        | 67 |
| 致谢        |        | 69 |
| 发表论文和科研情况 |        | 70 |

## 插图

|       |  |    |
|-------|--|----|
| 图 3.1 | 逐步剪枝策略 . . . . .                                   | 14 |
| 图 3.2 | 多粒度概要数据逼近与某候选轨迹的真实距离 . . . . .                     | 16 |
| 图 3.3 | 剪枝示例 . . . . .                                     | 17 |
| 图 3.4 | 利用下界和全局阈值的剪枝过程 . . . . .                           | 20 |
| 图 4.1 | 哈尔小波变换 . . . . .                                   | 27 |
| 图 4.2 | ED-FTB 剪枝效果图 . . . . .                             | 36 |
| 图 4.3 | 下界对剪枝结果的影响 . . . . .                               | 37 |
| 图 4.4 | $n, k$ 和 $M$ 对 ED-FTB 算法通信开销的影响 . . . . .          | 37 |
| 图 4.5 | Communication cost comparison on T-Small . . . . . | 38 |
| 图 4.6 | Scalability of ED-FTB on T-Big . . . . .           | 39 |
| 图 5.1 | 带约束的动态时间卷曲 . . . . .                               | 45 |
| 图 5.2 | 包围信封 . . . . .                                     | 46 |
| 图 5.3 | DTW-FLB 剪枝效果 . . . . .                             | 53 |
| 图 5.4 | DTW-FLB 通信开销与 $n, M$ 和 $k$ 之间的关系 . . . . .         | 54 |
| 图 5.5 | DTW-FLB 性能与 $\delta$ 间的关系 . . . . .                | 55 |
| 图 5.6 | The scalability of DTW-FLB . . . . .               | 56 |



## 表格

|  |    |
|--|----|
| 表 5.1 时间序列 $\mathcal{S}$ 的多粒度包围盒 . . . . . | 47 |
|--|----|

## 第一章 绪论

轨迹大数据是移动物联网时代的产物，记录了移动对象的行为信息。通过轨迹数据分析技术可以挖掘人类活动规律与行为特征、城市车辆的移动模式、大气环境变化规律等信息。这些挖掘成果为研究城市发展、社会变迁和自然环境演变提供重要的参考价值。本文重点研究了分布式  $\text{top-}k$  相似移动轨迹查询问题。本章中，章节 1.1 阐述了分布式轨迹数据相似性查询的研究现状；章节 1.2 详述了本文的具体研究内容以及所遇到的挑战；章节 1.3 简述了本文的主要研究方法和研究贡献。

### 1.1 研究背景

随着卫星定位导航、无线通信和普适计算等高新技术的不断发展，带有定位功能的移动智能设备已被广泛应用，并成为社会生产、生活的重要组成部分。移动对象（含人、动物和车辆等）在主动或被动使用这些设备的同时，产生了大量记录其移动历史行为的轨迹数据（Trajectory Data）。例如：滴滴出行，全球最大的一站式多元化出行平台。2017 年 10 月，滴滴出行发布的《第三季度全国重点城市交通出行报告》称其每天新增的定位轨迹数据超过 70TB<sup>1</sup>。轨迹数据是地理空间在时间轴上形成的多维空间中的一条曲线，表示了移动对象在一段时间内时空信息等移动行为的变化。每条轨迹可看作一条时空采样点构成的序列，其中每个采样点记录了时间、位置、速度、方向等信息。从微观角度将，轨迹数据蕴含了移动对象的移动模式与规律，例如从轨迹数据种我们可以发现市民的居住地、工作地和消费娱乐场所。从宏观角度来讲，海量的轨迹大数据蕴含了群体的移动迁徙和社会发展的变化，如城市的发展、交通的演化以及社会的变迁。通过轨迹分析等手段进行知识发现，并将它们运用在各种交通和服务应用系统中，包括交通导航、交通智能指挥、车辆监控、物流配送、城市规划、军事调度等。

---

<sup>1</sup><http://www.xiaojukeji.com/index/index>

海量的轨迹数据具有重要的社会和应用价值，不仅为解决拥堵、改善交通服务、缓解能源紧缺和降低大气污染等社会问题提供了新的机遇，而且对认知人民的社会活动、优化公共资源配置、为建立新型共享经济有着特殊的意义。2017年12月8日，习近平总书记在中共中央政治局第二次集团学习时强调：实施国家大数据战略，加快建设数字中国。因此，轨迹大数据称为政府和企业的重要资源财富并得到广泛重视。在此背景下，轨迹大数据的分析和挖掘已经被学术和工业界大量研究并成为数据挖掘领域的重要的新兴分支。在工业界，百度地图能根据实时轨迹数据进行路径规划。摩拜进行了基于自行车骑行目的地预测的研究。美团和饿了么等外卖公司设计了基于轨迹数据的智能派单系统。上海电信根据手机信令数据研究了人口流动分析。学术界也出现一些针对轨迹数挖掘的热点研究工作，包括可伸缩的快速轨迹聚类 [1-4]、轨迹流的连续查询 [5]、路径规划及路径发现 [6]、汇集模式发现 [7]、旅伴模式发现 [8, 9] 以及实时共享乘车 [10] 等。

轨迹相似性度量是轨迹挖掘的核心内容。现有的轨迹数据间的相似性通常通过它们之间的距离来度量，即两条轨迹数据之间的距离越小，则认为它们之间月相似。根据这一准则，研究者们将应用于时间序列分析的距离度量函数应用到轨迹数据中，这些度量准则包括：欧式距离 (Euclidean Distance, ED)[11]，动态时间弯曲 (Dynamic Time Warping, DTW)[12]，最长公共子序列 (Longest Common Subsequence, LCSS)[13, 14]，实值序列上的编辑距离 (Edit Distance on Real sequence, EDR) [15]。部分研究者针对轨迹数据设计了霍斯托夫距离 (Hausdorff Distance, HD)[4]、弗雷歇距离 (Fréchet Distance, FD) [16, 17]、一路距离 (One Way Distance, OWD) [18] 等。此外，还有部分学者针对轨迹的语义特性设计了用于计算轨迹语义相似度的距离 [19-21]。尽管这些距离度量函数可以较好地捕捉到轨迹之间的相似度，但它们的计算复杂度高，要想应用到大规模轨迹数据集上需要进一步研究如何提高计算效率。此外，如何针对特定问题选择合适的相似度度量函数也是重要的研究内容 [22, 23]。

此外，轨迹数据由于来源广、数据量大等特性，往往以分布式形式采集和存储

在各结点中。这类结点既可以是管理着大量轨迹数据的高性能服务器也可以是管理单一移动对象的个人智能手机。此外，结点间可能相互独立，不能互相访问。因此，如何对分布在具有不同存储和计算能力的各个结点上的轨迹数据进行统一的存储、分析和挖掘服务是合理利用迹数据的首要问题。现有的分布式解决方案可分为以下三类：(i) 基于对等系统（Peer-To-Peer, P2P）的架构。该架构中各个结点能互相访问，因此不适用与轨迹大数据。(ii) 基于主-从式（Master-Slave）的分布式集群架构。该架构中由一个 Master 结点和多个 Slave 结点构成，其中 Slave 结点存放数据并负责具体任务的执行，Master 结点存放数据的元数据并负责任务的分发和调度。基于该类架构典型的系统有 Hadoop 和 Spark。该架构往往要求所有物理结点在同一个集群内部，结点间通过高速网络连通。因此，也不符合轨迹大数据的要求。(iii) 基于协调者-远程结点的（Coordinator-Remote Site）的分布式架构。该架构中存在一个协调者和多个远程节点，其中协调者结点不存储任何数据只负责跟远程结点通信，而远程结点负责存放本地数据，且各远程结点间互不通信。第三种方案可看做第二种方案的特例，首先它允许所有结点物理上互相远离，不要求它们位于同一集群内部。其次，远程结点间做到完全独立且互不知晓。以上两点能有效地保证数据拥有者的隐私和数据存储、计算的分布性。

目前，分布式轨迹相似度研究已经得到广泛的重视。文献 [24, 25] 研究了主-从式架构下轨迹数据的 join 查询，其关注重点是如何降低从节点间数据交换量过高的问题。文献 [26] 研究了基于协调者-远程节点架构下的 top- $k$  相似轨迹查询研究。其研究的是用户手机连接基站的轨迹数据并将每个基站看做一个远程节点。由于手机在用户移动过程中会连接到不同的基站，故轨迹数据被切割成若干数据片段存放在不同的基站中。在该研究中，其研究目标是如何提高查询的效率而未关注如何降低通信开销。类似的，Smart Trace[14] 和 Smart Trace<sup>+</sup> [13] 在相同的系统架构中研究了相同的问题，但在其研究内容中，将每个智能手机看做远程结点且每个结点仅保存一条轨迹数据。在这两项工作中，计算效率和通信开销同时得到了考虑。与以上不同的是，本文研究了更加抽象的分布式场景，即每条轨迹完整的保

存在一台远程结点上，且每个远程结点可存储多条轨迹数据。

## 1.2 研究内容与挑战

本文选用基于协调者-远程节点的分布式架构进行管理和分析轨迹大数据，并对轨迹大数据上的  $\text{top-}k$  查询进行了研究。该查询可以表示为给定待查询轨迹  $Q$ ，从分布式轨迹数据集  $\mathcal{D}$  中找出与  $Q$  最相似的  $k$  条轨迹。针对该查询和给定的系统架构，一个直观的解决方案就是用户将  $Q$  提交给协调者结点后，该结点直接将  $Q$  发送给所有的远程结点。远程结点接收到查询对象后，计算其与本地所存储的轨迹数据的距离并将相似度值返回给协调者结点。最终，协调者结点从接收到的距离值中选择最小的  $k$  个作为结果返回给用户。这样的解决方案，设计简单且易于实现，但也存在着通信量过大的问题，尤其是当待查询轨迹数据量较大或远程结点较多时。因此，如何降低通信开销是我们需要解决的首要问题。此外，对于相似度计算往往开销较大，我们需要考虑如何提高查询的执行效率。最后，如上节所介绍，相似度距离计算准则很多，现有的工作只针对某一具体距离来进行通信或效率优化。因此，缺乏统一的框架或方法来处理。综上所述，本文研究面临的挑战主要表现为以下 3 点：

- **首先，针对时间序列上已有的  $\text{top-}k$  查询技术往往只针对某一相似度准则设计，缺乏通用性。**相似度度量准则是轨迹数据挖掘分析的核心内容，选用不同的度量函数，导致的挖掘结果以及对结果的理解往往也大不相同。现有的工作都是针对某一具体研究问题，精心挑选或自定义一相似度度量准则以满足研究的需要。因此，所提出的解决方案具有较高的局限性，不能推广到其他类似的问题中。为此，需要提出些通用或对一类问题使用的解决方案。
- **其次，现有分布式  $\text{top-}k$  查询算法中仍然存在着通信开销过大的问题。**直接将原始查询轨迹发送到所有远程结点的方式，其通信开销随轨迹长度和远程结点的个数的增加而快速增大。而现有的数据降维方案如 Douglas-Peucker 算法、奇异值分解 (Singular Value Decomposition, SVD)、离散傅里叶变换 (Discrete

Fourier Transform,DFT)、HAAR 小波变换 (Haaar Wavelet Transform,HWT)、分段聚合近似 (Piecewise Aggregate Approximation, PAA)、自使用分段常量近似 (Adaptive Piecewise Constant Approximati, APCA) 等方法虽然降低了数据的维度,但也造成了数据信息损失,使得数据不再直观且不能结果的准确性。此外,这些方法大都针对一维时间序列数据,而轨迹是天然的多维时间序列数据。为此,需要提出新的轨迹数据降维方法,在保证查询结果正确性的同时,降低通信开销。

- 最后,现有的分布式  $\text{top-}k$  查询时间效率需要得到提升。轨迹相似度的计算除欧式距离是一次的计算复杂度,其他都需要二次的计算复杂度。在每个远程结点进行查询轨迹与局部保存的轨迹进行两两相似度计算的方案不可行。索引技术是加快查询速度有效工具,但现有的索引技术只能针对原始轨迹数据,无法适用于降维后的数据。为此,需要设计新的方案来提高查询效率。

### 1.3 主要贡献

本文围绕分布式  $\text{top-}k$  相似轨迹查询这一问题,系统性的提出了解决方案。首先,针对轨迹相似度度量的多样性,提出了两种查询实现框架。这两个框架分别针对不同类别的相似度准则,在保证查询结果正确性的同时能有效地降低通信开销。接着,我们将两个常用相似度距离函数分别嵌入到两个框架中,并提出了对应的算法。最后,使用真实轨迹数据集验证了算法的性能。因此,本文的主要贡献分为以下 3 点:

- 针对通信开销较高问题,提出了两个通用查询实现框架 **FTB (Framework with Two Bounds)** 和 **FLB (Framework with Lower Bound)**。FTB 框架中要求能根据降维后的概要数据计算出相似度距离的上下界,并能保证概当细粒度的概要数据获取后,所计算出来的上下界越来越紧并最终能收敛。FLB 框架中仅要求能根据降维后的概要数据计算出相似度距离的下界,且能保证当

细粒度的概要数据获取后，所计算出来的下界越来越紧。这两个框架由于仅需要传递查询轨迹的概要数据，因此能大大降低通信开销。

- **针对 FTB 框架，提出了基于欧式距离的 FTB-ED 算法。**为验证 FTB 框架的有效性，本文研究了如何将具体欧式距离嵌入到该框架中。为此，本文首先利用 Haar 小波变换以得到不同粒度的轨迹概要数据，并证明了全部概要数据的欧式距离等于原始轨迹数据的欧式距离。接着，利用部分概要数据，本文提出了基于欧式距离的轨迹相似度上、下界，并理论证明了该相似度的正确性。然后，我们将该上、下界应用到 FTB 框架中，并提出了 FTB-ED 算法。在该算法中，我们引入了索引等机制以提高查询效率。最后，我们通过大量实验验证了 FTB-ED 算法的有效性和可扩展性。
- **针对 FLB 框架，提出了基于动态时间卷曲距离的 FLB-DTW 算法。**为验证 FLB 框架的有效性，本文研究了如何将具体 DTW 距离嵌入到该框架中。为此，本文首先针对查询轨迹使用不同粒度的包围信封（Bounding Envelope）来表示轨迹的概要数据。接着提出了基于动态时间卷曲距离的下界，并理论证明了该下界的正确性。最后将该下界应用到 FLB-DTW 框架中，并提出了 FLB-DTW 算法。在该算法中我们引入了多种机制以提高查询效率。最后，我们通过大量实验验证了 FLB-DTW 算法的有效性和可扩展性。

## 1.4 章节安排

本文一共分为六章，章节安排如图 ?? 所示：

- 第二章从轨迹数据存储、轨迹降维以及轨迹数据相似性查询三个方面介绍了研究背景知识和研究现状。
- 第三章介绍了两个通用查询处理框架 FTB 和 FLB 以分别应对能同时提供上、下界的和仅能提供下界的相似度准则。
- 第四章介绍了如何将欧式距离嵌入到 FTB 框架中，并提供高效的查询结果。

- 第五章介绍了如何将 DTW 距离嵌入到 FLB 框架中，并提供高效的查询结果。
- 第六章对上述已有工作进行了总结，并展望了未来的研究方向和内容。





## 第二章 问题定义及研究现状

本章第一节首先对数据模型和要解决的查询问题进行了定义，然后从以下方面介绍了相关的研究工作：轨迹索引、轨迹降维、轨迹相似度度量、时间序列数据 top- $k$  查询，最后对本章内容进行小结。

### 2.1 问题定义

本文的研究目标是给定一条查询轨迹，从存储在若干远程结点中的轨迹中找出  $k$  条距离最短或相似度最高的轨迹。为此，本节首先介绍了轨迹数据模型，接着介绍了查询的定义。

#### 2.1.1 轨迹数据

轨迹是描述移动对象移动行为的数据。通常来说，一条轨迹  $T$  可看做包含  $n$  个元素（即轨迹点）的有序序列。每个轨迹点  $p$  包含了时间、位置等维度的信息。因此轨迹可被形式化定义为如下：

**定义 2.1.1 (轨迹).** 轨迹形式化表示为:  $T = \{p_0, p_1, \dots, p_{n-1}\}$ 。

$$T = \{p_0, p_1, \dots, p_{n-1}\} \quad (2.1)$$

其中  $|T| = n$  代表轨迹所包含的点数，即轨迹的长度。每个轨迹点  $p$  包含时间 ( $t$ )、位置 ( $l$ ) 等维度的信息。因而  $|p| = d$  称为轨迹的维度。此外轨迹中的点严格按时间升序排列，即  $\forall i, j, 0 \leq i \leq j < n$  则  $p_i.t \leq p_j.t$ 。

轨迹数据的来源多样且复杂。根据移动对象的划分可分为如下几类：

- **人类活动轨迹数据:** 该类数据分为主动式和被动式。主动式数据是人们主动利用移动定位设备分享或汇报自己的位置等信息。典型的有社交网络中的数

据, 用户提交位置获得服务的数据。被动式数据是人们无意间使用各种服务时所产生的轨迹数据。典型的有公交刷卡轨迹和手机的信令轨迹数据。

- **交通工具轨迹数据:** 这类数据主要是交通工具使用车载 GPS 设备所产生的移动轨迹数据。例如, 出租车、公交车的活动轨迹数据。
- **动物活动轨迹数据:** 这类数据是为了研究动物生活、迁徙等行为和习惯而捕获的数据。
- **自然现象活动轨迹数据:** 这类数据典型的有台风、冰山、海洋事件等的轨迹数据, 用以探索自然现象的活动规律。

轨迹数据符合大数据时代的 3V 特征, 即量大、实时、多样。轨迹数据采样由于受设备、采样频率等因素影响, 数据质量较低且各个轨迹的采样间隔差异显著。这些问题导致原始轨迹数据的可用性较低。因此, 我们在进行轨迹数据分析前往往需要经过数据清理 (data cleaning)、地图匹配 (map matching)、轨迹分段 (trajectory segmentation) 等预处理方式化为校准轨迹。校准轨迹数据能够通过数据管理技术进行轨迹索引以便有效地存取。因此, 本文所处理的轨迹数据为预处理后的校准轨迹数据。这样的数据有如下特点: (i) 采样频率一致; (ii) 长度一致; (iii) 位置精度高。这为我们挖掘轨迹模式从而提炼有价值的知识提供了可靠保障。

### 2.1.2 分布式 k 近邻轨迹查询

轨迹数据往往是分布式采集并存储的。为此假设有  $M$  个远程结点, 每个远程结点  $i$  包含轨迹数据集  $TS_i$ 。那么整个分布式轨迹数据集  $TS = \bigcup_{i=1}^M TS_i$ 。我们的目标是给定查询轨迹, 从分布式存储的  $TS$  数据集中, 找出与其距离最近的  $k$  条轨迹。下面我们将给出查询的形式化定义:

**定义 2.1.2** (分布式 k 近邻轨迹查询). 该查询形式为  $query(Q, TS, DM, k)$ , 其中  $Q$  为给定查询轨迹,  $TS$  为分布式轨迹数据集,  $DM$  为距离度量准则以及  $k$  为返回结

果集大小。查询的目标是返回满足如下条件的轨迹集  $\mathcal{S}$ : (1)  $\mathcal{S} \subseteq TS$ ; (2)  $|\mathcal{S}| = k$ ; (3)  $\forall \mathcal{C} \in \mathcal{S}, \mathcal{C}' \in TS - \mathcal{S}, DM(\mathcal{Q}, \mathcal{C}) \leq DM(\mathcal{Q}, \mathcal{C}')$ 。

传统的集中式环境下  $k$  近邻轨迹查询相比, 分布式场景下的查询不仅注重查询效率, 而且尤其注重通信开销。这是由于分布式场景中, 远程结点和协调者结点的带宽资源往往是有限的。高的通信开销, 意味着用户可能要花费更多的金钱。因此, 用户允许多花一点时间以达到降低通信开销的目的。

## 2.2 轨迹索引

## 2.3 轨迹降维

轨迹数据

## 2.4 轨迹相似度度量

## 2.5 时间序列 top- $k$ 查询



## 第三章 查询处理框架

本章主要介绍两个查询处理框架以分别处理不同应用场景。首先, 章节3.1阐述了本章的研究思路。其次, 章节3.2介绍了协调者节点和远程结点上的通信接口函数。然后, 章节3.3介绍了同时利用距离函数上、下界的 FTB 框架, 并给出了实现方案。接着, 章节3.4介绍了仅利用距离函数下界的 FLB 框架, 并给出了实现方案。最后, 小结本章的研究内容。

### 3.1 引言

本文的主要研究目标就是降低  $\text{top-}k$  查询时的网络开销, 直接将查询轨迹发送到所有远程结点的方式虽然能保证结果的正确性, 但网络传输开销消耗较大。为降低这一开销, 一个直观的思路是: 协调者结点选用合适的降维策略首先对轨迹数据进行降维处理, 得到一个概要数据并将该概要数据发送给所有远程结点。远程节点在获取到概要数据后, 我们希望根据概要数据能够对查询轨迹和局部轨迹能计算出距离的范围以进行剪枝。此外, 我们希望通过降维得到的概要数据具有多粒度特性, 即粒度越细, 所包含的信息量也越多。同时, 所计算出来的距离范围也越精确。通过不断精确的距离范围, 我们能进一步剪枝不相关的候选。整个剪枝过程如图3.1所示, 随着概要数据从粒度 1 到粒度  $n$  的不断传输, 整个系统会过滤掉越来越多的候选。这里有两点需要注意: (i) 粒度越高, 对应的概要数据所能表示的信息量也越接近原始轨迹数据, (iii) 在由粗到细剪枝过程中, 可能不需要到最细粒度, 结果已经找出。

从以上思路出发, 本章从所得到的距离范围出发考虑了两种情况: (i) 根据概要数据能同时计算出所给距离函数的上界和下界, (ii) 根据概要数据仅能计算出所给距离函数的下界。至于仅能计算出上界的情况不作考虑, 这是因为我们需要找的是距离最近的  $k$  个轨迹。为此, 我们首先给出了上、下界特征定义。

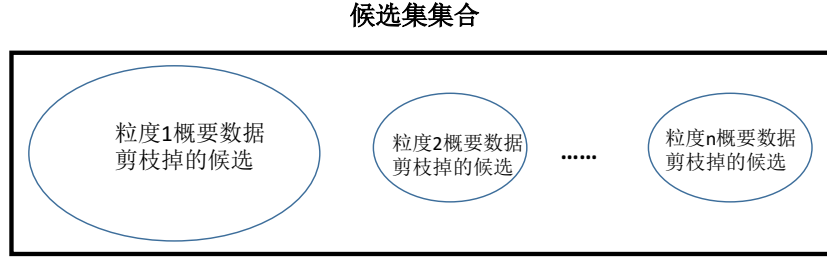


图 3.1: 逐步剪枝策略

**定义 3.1.1.** 界特征 (*Bound Feature, BF*) . 界特征,  $\langle id, ub, lb \rangle$ , 记录了一条轨迹的标识 ( $id$ ), 以及它与待查询轨迹的距离上界 ( $ub$ ) 和下界 ( $lb$ )。

对任意一条轨迹其默认的距离下界为 0, 上界为正无穷。我们的目标是不断提高界特征的下界或降低界特征的上界。此外, 根据前面所述, 我们有时不能同时获得一个较紧的上界和下界。此时, 只需使用其保留下界信息。在本章, 我们将提出两个查询处理框架。其中前一个框架中将同时使用距离函数的上界和下界。而后一个用来处理仅有下界存在的情况。

### 3.2 接口函数

在介绍如何进行查询处理框架之前, 本节将首先介绍, 查询处理所用到的基本通信和处理接口函数。这些接口均为抽象函数, 所以在具体应用中, 用户可以添加具体的内容 (我们再接下来两章将会介绍这些接口函数在使用欧式和动态时间卷曲距离下的具体实现)。我们将这些接口函数按照运行所处位置分为两类。一类是协调者结点上接口函数, 另一类是远程结点接口函数。

**协调者结点函数接口:**

- `coordinatorInit( $\mathcal{Q}, \mathcal{R}$ )`. 协调者结点运行该函数以实现查询初始化, 涉及的内容包含对查询轨迹  $\mathcal{Q}$  进行概要数据计算以及协调者结点跟所有远程结点的信息传递。
- `generateInfo()`. 协调者结点生成所需要发送的数据。

- $\text{sendToRemoteSites}(\mathcal{R}, \xi)$ . 协调者节点将数据  $x$  发给所有在集合  $R$  中的远程节点。
- $\text{getFromRemoteSites}(\mathcal{R})$ . 协调者节点从集合  $\mathcal{R}$  中的每个远程结点获取信息。

协调者结点函数接口：

- $\text{remoteInit}(TS_r, S_r)$ . 远程结点运行该函数以实现初始化，主要工作其为保存在集合  $TS_r$  中的每条轨迹初始化界特征，并将这些特征值放在集合  $S_r$  中。此外，该函数还可能涉及到对每条轨迹数据进行概要数据计算。
- $\text{getFromCoordinator}()$ . 该函数接受协调者结点运行  $\text{sendToRemoteSites}$  函数时所发数据。
- $\text{sendToCoordinator}(x)$ . 远程结点将信息  $x$  发送给协调者结点。协调者节点则通过  $\text{getFromRemoteSites}$  函数接受所发消息。
- $\text{updateBounds}(S_r, x)$ . 远程结点根据获取到的信息  $x$  更新界特征集合中值。

以上接口函数，为协调者结点与远程结点间的通信提供了保障。接下来的章节，我们将介绍如何利用这些接口函数所传递的信息来进行剪枝。

### 3.3 FTB 框架

本节将介绍 FTB 框架的设计原理及实现方案。

#### 3.3.1 FTB 框架设计原理

本章第一节提到过，我们假设能对查询对象找到一个数据存储空间较小的概要数据，并利用该概要数据同时计算出距离的上、下界。FTB 框架核心思想就是从这点假设出发，具体的是：使用由粗到细粒度的概要数据，以不断获取更加紧凑的距离上、下界，并以此来进行剪枝过滤。图3.2给出了使用多粒度概要数据不断逼近与某一候选轨迹真实距离的过程。图中，横坐标为概要数据的粒度，纵坐标表示



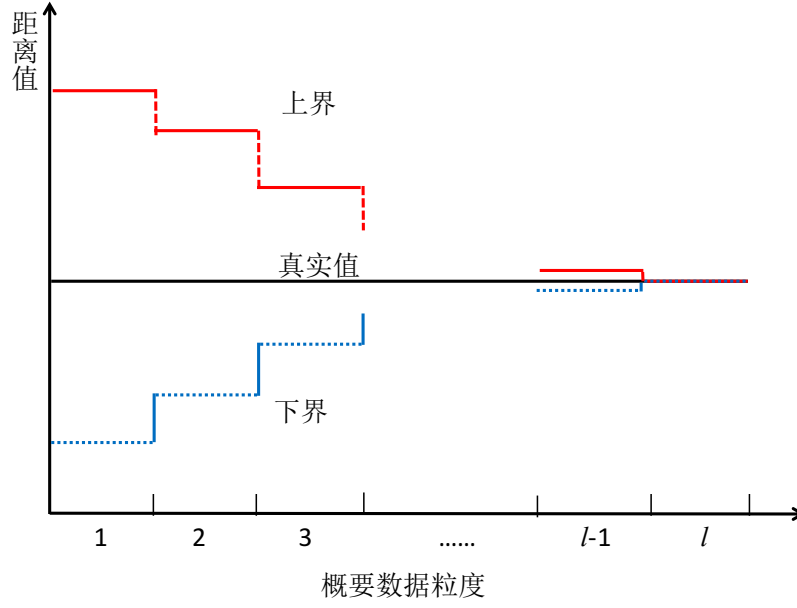


图 3.2: 多粒度概要数据逼近与某候选轨迹的真实距离

相似度距离的值。当概要数据粒度，不断增加的过程中，我们所计算出的相似度上界（红色实线）和下界（蓝色虚线）不断逼近相似度的真实值（黑色实线），并最终等于真实值。当获得了所有轨迹的界特征（上、下界）后，我们就可以使用它们进行剪枝。

**引理 3.3.1 (剪枝原理).** 给定一界特征集合  $S$ ，若某界特征的下界大于集合中第  $k$  小的上界，则该界特征所对应轨迹不会进入最终结果集，即可以被剪枝掉。

**证明.** 对  $S$  中的界特征按上界值由小到大排序，并记排序后的第  $k$  个界特征为  $S[k]$ 。假设某轨迹的特征为  $S[c]$  ( $c > k$ )，且  $S[c].lb > S[k].ub$ 。由于  $S[c]$  所对应的真实值大于  $S[c].lb$ ，所以有  $S[c]$  的真实值大于  $S[k].ub$ 。此外，又由于  $S[k].ub$  大于前  $k-1$  个界特征的上界，则  $S[k].ub$  大于前  $k-1$  个界特征所对应轨迹的真实距离值。因此， $S[c]$  所对应的轨迹真实距离值大于前  $k-1$  个界特征所对应轨迹的真实距离值。故  $S[c]$  可以被剪枝掉。  $\square$

图3.3展示介绍了剪枝原理。首先，将界特征按上界有小到大拍完序后。对于第  $c$  个轨迹 ( $c > k$ )，若其下界大于第  $k$  个的上界。则该轨迹的真实值必然大于前

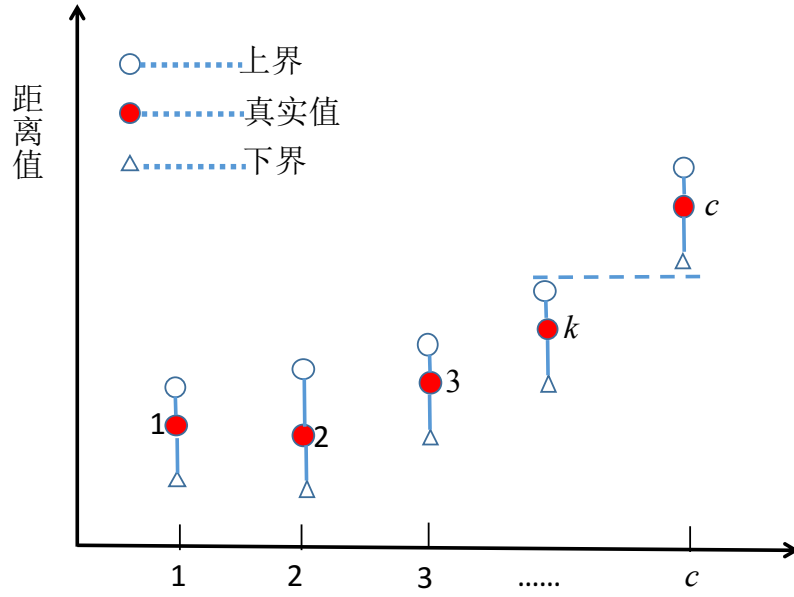


图 3.3: 剪枝示例

$k$  个轨迹的真实值。由于我们的差值只找距离最小的  $k$  条轨迹，所以第  $c$  个轨迹不可能存在于结果集中。此时，根据对其计算出来的上、下界就可以将其从候选集合中移除。

### 3.3.2 FTB 框架实现

本节将首先介绍 FTB (Framework with Two Bounds) 框架，以处理那些能同时得到距离上、下界的情况。此外，需要保证当使用最细粒度的概要数据计算上、下界时，上界等于下界。即此时计算出来的上、下界等于距离的真实值。该框架分为两个部分，一部分是运行在协调者结点的算法 1，另一部分是运行在所有远程结点的算法 2。在框架运行过程中，这两个部分互相通信并协调工作。整个过程，可以分为 3 个阶段：初始化阶段，迭代交互式剪枝阶段，最终结果获取阶段。下面将对这三个部分分别进行介绍：

**初始化阶段：**协调者结点在此阶段执行多种操作，如获取远程结点的列表，概要数据计算以及可能预发送一些信息。在此过程中，我们使用  $\mathcal{R}$  来表示远程结点的集合（算法 1:1-2 行）。与之对应的，远程结点在该阶段的初始化主要工作是为

**算法 1** FTB 之协调者结点**输入:** 查询轨迹  $Q$ , 结果集大小  $k$ ;**输出:**  $k$  条最相似轨迹的 ID;

```

1:  $\mathcal{R} \leftarrow$  远程结点集合;
2: coordinatorInit( $Q, \mathcal{R}$ );
3: while true do
4:   /* 生成全局第  $k$  小上界 */
5:    $Info \leftarrow$  generateInfo(); # 准备概要数据
6:   sendToRemoteSites( $\mathcal{R}, Info$ );
7:    $GUBS \leftarrow$  getFromRemoteSites( $\mathcal{R}$ );
8:    $gkub \leftarrow \operatorname{argmin}_{\tau}(|x \in GUBS, x < \tau| \geq k)$ ;
9:   sendToRemoteSites( $\mathcal{R}, gkub$ );
10:  /* 获取候选集大小  $\mathcal{R}$  */
11:   $CSS \leftarrow$  getFromRemoteSites( $\mathcal{R}$ );
12:   $\mathcal{R} \leftarrow \{x.r | x \in CSS, x.|S_r| > 0\}$  # 剪枝远程结点
13:   $sum \leftarrow \sum_{x \in CSS} x.|S_r|$  # 候选集大小
14:  if  $sum == k$  then
15:    sendToRemoteSites( $\mathcal{R}, finish$ );
16:    break;
17:   $ids \leftarrow$  getFromRemoteSites( $\mathcal{R}$ );
18: return  $ids$ ;
```

保存在本地数据集  $TS_r$  中轨迹初始化界特征并将结果存在局部候选集  $S_r$  中。此外，它也会接受协调者结点发过来的数据（算法 2:1-2行）。一开始，所有轨迹都是候选。我们把包含候选的远程结点称为候选（远程）结点。

**迭代交互式阶段:** 在此阶段, 协调者结点与候选远程结点交互通信直到剪枝完毕。首先, 远程结点为准好粗粒度的概要数据并发送给所有候选结点（算法 1:5-6行）。在接收到概要数据后, 候选远程结点利用该概要数据进行计算距离上、下界并更新界特征。需要注意的是: 随着迭代次数的增加, 所生成的概要数据粒度越细。这使得我们计算出来的上、下界越来越紧。根据最新的界特征, 每个候选结点找出局部最小的  $k$  个上界并将它们发送给协调者结点 (算法 2:5-10行)。当协调者结点获取到所有候选远程结点发送来的上界后, 它从中选择第  $k$  小的上界, 记为  $gkub$ , 并将其值发送给那些候选远程结点 (算法 1:7-9行)。候选结点在接收到  $gkub$  后, 就可以剪枝掉局部的一些候选轨迹。具体剪枝做法是, 每个局部候选的下界与  $gkub$  进行比较。如下界大于  $gkub$ , 则将该轨迹从局部候选中删除。剪枝后, 候

**算法 2** FTB 之远程结点**输入:** 局部轨迹集合  $TS_r$ ;**输出:** 属于 top- $k$  结果集的轨迹 ID;

```

1:  $S_r \leftarrow$  局部轨迹界特征集;
2: remotelnit( $TS_r, S_r$ );
3: while true do
4:    $m \leftarrow$  getFromCoordinator(); # 从协调者结点获取信息
5:   if  $m$  is Info then
6:      $S_r \leftarrow$  UpdateBounds( $S_r, m$ );
7:     /* 产生局部最小  $k$  个上界 */
8:      $\hat{ub} \leftarrow \operatorname{argmin}_\tau (|x \in S_r, x.ub < \tau| \geq k)$ ;
9:      $S' \leftarrow \{\alpha.ub \mid \alpha \in S_r, \alpha.ub \leq \hat{ub}\}$ ;
10:    SendToCoordinator( $S'$ ); # 将最小的  $k$  个上界返回给协调者结点
11:   else if  $m$  is gkub then
12:      $S_r \leftarrow \{\beta \mid \beta \in S_r, \beta.lb \leq m\}$  # 局部剪枝
13:     SendToCoordinator( $\langle r, |S_r| \rangle$ );
14:     if  $|S_r| = 0$  then
15:       return ; # 当远程结点无候选, 则停止运行
16:   else
17:     /*  $m$  is finish */
18:      $ids \leftarrow \{a.id \mid a \in S_r\}$ ;
19:     SendToCoordinator( $ids$ );
20:   return ;

```

选结点将其所剩候选的个数发给协调者结点。若候选结点的所有轨迹都被剪枝掉, 则该结点运行结束 (算法 2:11-15行)。在此之后, 协调者结点收集候选远程结点所发送的局部候选的个数, 并求和计算总的个数。若发现某结点不再包含候选, 则将其从候选结点集中移除。此外, 若发现当前所剩候选总数正好为  $k$  个, 则迭代终止。协调者将向剩下的候选结点发送终止信号 *finish*。若所剩个数大于  $k$ , 则迭代继续 (算法 1:11-16 行)。

**最终结果获取阶段:** 经过上一阶段, 最终会剩余  $k$  个候选。这些候选即为最终的结果。所剩的候选结点在收到上一阶段所发的结束信号后, 会将本地所剩候选的 ID 发送给协调者节点。发送成功后, 自身会结束查询 (算法2: 18-20行)。而协调者节点则会收集所有候选结点发送过来的 ID, 并返回给用户 (算法1: 17-18行)。

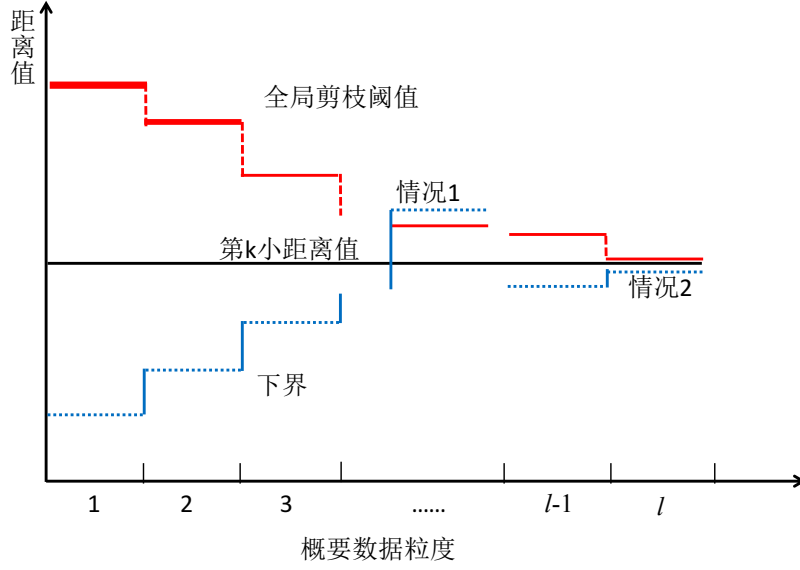


图 3.4: 利用下界和全局阈值的剪枝过程

### 3.4 FLB 框架

本节将介绍 FLB 框架的设计原理及实现方案。与 FTB 框架不同的是，本框架主要为那些仅能根据概要数据得到距离下界的距离准则而设计。

#### 3.4.1 FLB 框架设计原理

针对那些不能同时获得上界的距离度量准则，FTB 在使用由粗到细粒度的概要数据以不断获取更加紧凑的距离下界的同时，还不断计算一个越来越紧的全局阈值以剪枝数据。假设与待查询轨迹距离第  $k$  小的值为  $gk$ ，若某轨迹的界特征的下界大于该值，则该轨迹不会成为最终结果。但  $gk$  值很难一开始就计算出来。为此，本文的做法是以最小的代价（通信和时间开销）计算  $gk$  的上界  $\theta$ ，并不断将  $\theta$  逼近  $gk$ 。然后，在每轮迭代中，用  $\theta$  来剪枝候选。 $\theta$  就是我们用来剪枝的全局阈值。需要区别的是，在 FTB 框架中，是利用界特征的上界进行剪枝。

图3.4示例介绍了，FTB 框架对某条轨迹的剪枝过程。图中，中间的水平黑色直线表示查询轨迹与所有轨迹距离中第  $k$  小的距离值（ $gk$  值）。蓝色分段虚线介绍了轨迹距离下界随着概要数据粒度增加，该下界值越来越逼近其真实距离的过程。红

色分段实线介绍了全局阈值不断逼近  $gk$  值。在两者的不断逼近过程中会出现两种情况：第一种情况就是不断逼近过程中，该轨迹的下界大于阈值  $\theta$ 。此时，我们可以将轨迹剪枝掉。第二种情况就是当最细粒度的概要数据后，轨迹的下界仍小于阈值  $\theta$ 。此时，我们将对该轨迹继续保留为候选以待进一步分析。最后，当根据最细粒度的概要数据进行剪枝后，若所保留的候选数仍超过  $k$ 。此时，FTB 框架将向包含这些候选的远程结点发送原始轨迹，远程结点则可以计算出真实的距离值并找出最终的  $k$  个距离最近的轨迹。

---

**算法 3** FLB 之协调者结点

---

**输入:** 查询轨迹  $Q$ , 结果集大小  $k$ ;

**输出:**  $k$  条最相似轨迹的 ID;

```

1:  $\mathcal{R} \leftarrow$  远程结点集合,  $\mathcal{R}' \leftarrow null$ ;
2: coordinatorInit( $Q, \mathcal{R}$ );
3: while true do
4:    $Info \leftarrow generateInfo()$ ; # 生成待发送概要数据
5:   if  $Info \neq null$  then
6:     sendToRemoteSites( $\mathcal{R}, Info$ ); # /*(6-13行): 更新全局阈值  $\theta$ */
7:      $GLBS \leftarrow getFromRemoteSites(\mathcal{R})$ ; # 从远程结点获取局部 top- $k$  下界
8:      $gklbs \leftarrow argmin_{\tau}(|x \in GLBS, x.lb < \tau| \geq k)$ ; # 选取下界最小的  $k$  条轨迹
9:      $\mathcal{R}'' \leftarrow$  the remote sites that contain  $gklbs$ ; # 找出包含上述轨迹的远程结点
10:    sendToRemoteSites( $\mathcal{R}'' - \mathcal{R}', \langle Q, gklbs \rangle$ ); # 对未收到  $Q$  的结点进行处理
11:    sendToRemoteSites( $\mathcal{R}'' \cap \mathcal{R}', \langle null, gklbs \rangle$ ); # 对已收到  $Q$  的结点进行处理
12:     $sv \leftarrow sv \cup getFromRemoteSites(\mathcal{R}'')$ ;
13:     $\theta \leftarrow argmin_{\tau}(|x \in sv, x < \tau| \geq k)$ ; # 选取第  $k$  小相似度值
14:    sendToRemoteSites( $\mathcal{R}, \theta$ );
15:     $\mathcal{R}' = \mathcal{R}' \cup \mathcal{R}''$ ;
16:    /*Step 2: 剪枝候选远程结点 */
17:     $CSS \leftarrow getFromRemoteSites(\mathcal{R})$ ; # /*(17-22行): 剪枝并反馈 */
18:     $\mathcal{R} \leftarrow \{x.r | x \in CSS, x.|S_r| > 0\}$ ;
19:     $sum \leftarrow \sum_{x \in CSS} x.|S_r|$ ;
20:    if  $sum == k$  then
21:      sendToRemoteSites( $CS, finish$ ); # top- $k$  结果已经找到, 发结束信号
22:      return getFromRemoteSites( $\mathcal{R}$ );
23:    else
24:      sendToRemoteSites( $\mathcal{R} - \mathcal{R}', \langle null, Q \rangle$ ); # /*(24-28行): 结果提炼 */
25:      sendToRemoteSites( $\mathcal{R} \cap \mathcal{R}', \langle null, null \rangle$ );
26:       $CSet \leftarrow getFromRemoteSites(\mathcal{R})$ ;
27:       $gkSim \leftarrow argmin_{\tau}(|x \in CSet, x.dis < \tau| \geq k)$ ;
28:      return  $\{x.id | x \in CSet, x.dis \leq gkSim\}$ ;

```

---

### 3.4.2 FLB 框架实现

本节将首先介绍 FLB (Framework with Two Bounds) 框架, 以处理那些仅能获取到下界的情况。该框架分为两个部分, 一部分是运行在协调者结点的算法 3, 另一部分是运行在所有远程结点的算法 4。在框架运行过程中, 这两个部分互相通信并协调工作。整个过程, 可以分为 4 个阶段: 初始化阶段, 全局阈值计算阶段, 剪枝阶段和结果提炼阶段。下面将对这四个阶段分别进行介绍:

**初始化阶段:** 协调者结点在此阶段执行多种操作, 如获取远程结点的列表, 概要数据计算以及可能预发送一些信息。在此过程中, 我们使用  $\mathcal{R}$  来表示远程结点的集合,  $\mathcal{R}'$  表示那些已经接受了查询轨迹的结点 (算法 3:1-2 行)。与之对应的, 远程结点在该阶段的初始化主要工作是为保存在本地数据集  $TS_r$  中轨迹初始化界特征并将结果存在局部候选集  $S_r$  中。此外, 它也会接受协调者结点发过来的数据 (算法 2:1-2行)。

**全局阈值计算阶段:** 该阶段, 我们的主要目标是通过迭代通信交互, 逐步获取更加精确的全局阈值。在每次迭代中, 协调者结点首先发送概要数据给所有候选远程结点 (算法 3 6行)。候选远程结点在接受到概要数据后, 更新每个候选轨迹界特征值 (注: 在框架中, 界特征只保留 ID 和下界, 无需保持上界)。接着选取局部下界最小的  $k$  个界特征返回给协调者节点 (算法 4: 6-9行)。其次, 协调者选取包含最小的  $k$  个下界的界特征集合  $gklbs$ , 及包含  $gklbs$  的远程结点集合  $\mathcal{R}''$  (算法 3: 7-8行)。然后, 它将待查询轨迹  $Q$  发送给  $\mathcal{R}''$  中的远程结点。在发送前, 他将  $\mathcal{R}''$  中的结点分为两类: 第一类是接收过  $Q$  的结点, 此时我们需将  $Q$  以及  $gklbs$  中属于该结点的轨迹发送给对应的结点。第二类是已接收过  $Q$  的结点, 此时, 仅需将对应结点的轨迹返回 (算法 3: 10-11)。当候选远程结点接收到信息这样的成对信息后, 针对本地出现在  $gklbs$  中的轨迹计算出距离值并返回给用户 (算法 4: 11-12 行)。需要注意的是, 对于接受成对信息的结点, 我们并没有对本地所有候选轨迹计算真实值。其原因是尽管使用全部候选计算出来的距离能有更好的逼近第  $k$  小距离值, 但这样做可能导致时间开销较大, 尤其是当数据集中在这些节点上时。远

**算法 4 FLB 之远程结点****输入:** 局部轨迹集合  $TS_r$ ;**输出:** 属于 top- $k$  结果集的轨迹 ID;

```

1:  $S_r \leftarrow$  局部轨迹界特征集;
2: remotelnit( $TS_r, S_r$ );
3: while true do
4:    $m \leftarrow$  getFromCoordinator(); # 从协调者结点获取信息
5:   if  $m$  is Info then
6:      $S_r \leftarrow$  UpdateLowerBound( $S_r, m$ );
7:      $\hat{lb} \leftarrow \operatorname{argmin}_\tau(|x \in S_r, x.lb < \tau| \geq k)$ ;
8:      $S' \leftarrow \{x \mid x \in S_r, x.lb \leq \hat{lb}\}$ ;
9:     SendToCoordinator( $\langle r, S' \rangle$ );
10:  else if  $m = \langle Q/null, gklbs \rangle$  then
11:     $sv \leftarrow \{distance(Q, x) \mid x \in gklbs\}$ ;
12:    SendToCoordinator( $sv$ );
13:  else if  $m = \theta$  then
14:     $S_r \leftarrow \{x \mid x \in S_r, x.lb \leq \theta\}$ ; # 局部剪枝
15:    SendToCoordinator( $\langle r, |S_r| \rangle$ );
16:    if  $|S_r| = 0$  then
17:      return ; # 结点不包含候选, 则结束
18:  else if  $m = finish$  then
19:     $ids \leftarrow \{x.id \mid x \in S_r\}$ ; # 收到结束信号, 则返回候选 ID 并结束
20:    SendToCoordinator( $ids$ );
21:    return;
22:  else
23:    SendToCoordinator( $\{\langle x.id, distance(x, Q) \rangle \mid x.id \in S_r\}$ );
24:  return ;

```

程结点在接收到那些包含  $gklbs$  的结点发送过来的距离值后, 选择最小的值作为全局阈值, 并将该值发送给候选结点 (算法 3: 13-14行)。

**剪枝阶段:** 该阶段主要利用  $\theta$  和下界进行剪枝。当候选远程结点接收到  $\theta$  后, 将会将下界小于所接收的阈值的轨迹从候选集中删除。剪枝完毕后, 该结点会将所剩候选数发送给协调者结点。此外, 当该结点不再包含任何候选时, 则停止运行 (算法 4: 13-17行)。协调者结点在接收到所有候选远程结点发送过来的候选数后, 计算出总的所剩候选数。若所剩候选总数正好为  $k$  个, 则说明所有候选已经找出。协调结点向候选远程结点发结束信号并等待接收最终结果的 ID (算法 3: 21-22行)。候选远程在接收到结束信号后, 则将本地候选的 ID 发送给协调者结点



(算法 4: 18-21行)。若所剩候选总数仍超过  $k$ , 则迭代执行第二和本阶段任务直到  $k$  个候选被找出或概要数据发送完毕。

**结果提炼阶段:** 由于界下界最终不一定能逼近到原始距离值, 且全局阈值不一定能逼近到第  $k$  小距离值。所以存在着当概要数据发送完毕, 所剩候选数仍超过  $k$  的情况。此时, 需要对剩下的候选进行甄别。**FLB** 框架的做法是将查询轨迹  $Q$  发送到剩下的候选结点中, 并对所有剩余候选计算真实的距离值以找出最终的  $k$  个结果 (算法 4: 24-28行)。需要注意的是在发送  $Q$  时, 若某候选已经接收过  $Q$  则无需再次发送。

### 3.5 本章小结

本章首先介绍了使用概要数据计算距离界特征, 并利用特征剪枝的思想。然后介绍了 **FTB** 框架以处理那些能根据概要数据同时计算出上、下界的距离函数。进一步的介绍了 **FLB** 框架以处理仅能根据概要数据计算出下界的距离函数。对比 **FTB** 和 **FLB** 两个框架的介绍, 我们可以发现两者的共同点都是计算一个全局值来和轨迹下界来剪枝。它们的区别是 **FTB** 框架计算出的全局值是从候选轨迹的上界中选取出来的, 而 **FLB** 框架是对某些轨迹计算真实的距离值, 并从这些距离值中选取出来的。所以, 尽管 **FLB** 框架也可以应用于能同时获取上、下界的距离函数, 由于其计算全局阈值需要额外的计算和通信开销。故对于能同时根据概要数据计算上、下界的距离函数, 我们使用 **FTB** 框架来处理。



## 第四章 FTB 框架的应用

本章主要介绍如何使用 FTB 框架处理基于欧式距离的查询。首先，章节4.1介绍了利用哈尔小波为欧式距离提供概要数据。然后，章节4.2介绍了基于小波系数的欧式距离上、下界。其次，章节4.3介绍了算法 ED-FTB，以处理当使用欧式距离作为距离度量准则时的查询。接着，章节4.3展示了 ED-FTB 算法的有效性和可扩展性。再其次，章节 4.5小结本章的研究内容。最后，本章涉及到的证明内容放在附件部分。

### 4.1 基于欧式距离的概要数据

#### 4.1.1 基于欧式距离的轨迹相似度量

欧式距离用于处理两条长度相同的轨迹。假设待查询轨迹  $\mathcal{Q}$  表示为  $\mathcal{Q} = \{q_0, q_1, \dots, q_{n-1}\}$ ，一条候选轨迹  $\mathcal{C}$  表示为  $\mathcal{C} = \{c_0, c_1, \dots, c_{n-1}\}$ ，其中  $n$  为轨迹长度，每个轨迹点来自  $d$  维空间，即  $q_i, c_i \in R^d$ 。我们首先给出点之间的距离定义，不失一般性的，我们使用欧式距离来度量。

$$ED(q_i, c_i) = \|q_i - c_i\| = \sqrt{\|q_i\|^2 + \|c_i\|^2 - 2q_i \cdot c_i} \quad (4.1)$$

其中  $\|q_i\|$  (或  $\|c_i\|$ ) 表示  $q_i$  (或  $c_i$ ) 的二范数的值， $q_i \cdot c_i$  表示向量  $q_i$  和  $c_i$  之间的点 (内) 积。在此基础上，我们给出轨迹间的欧式距离定义。

$$ED(\mathcal{Q}, \mathcal{C}) = \sqrt{\sum_{i=0}^{n-1} ED(q_i, c_i)^2} \quad (4.2)$$

为简化计算和表示，我们使用欧式距离的平方 (Squared Euclidean Distance, SED) 来替换距离，即  $SED(q_i, c_i) = ED(q_i, c_i)^2$  and  $SED(\mathcal{Q}, \mathcal{C}) = ED(\mathcal{Q}, \mathcal{C})^2$ 。此时，轨迹距离的欧式距离可以看作是点距离的累加和。

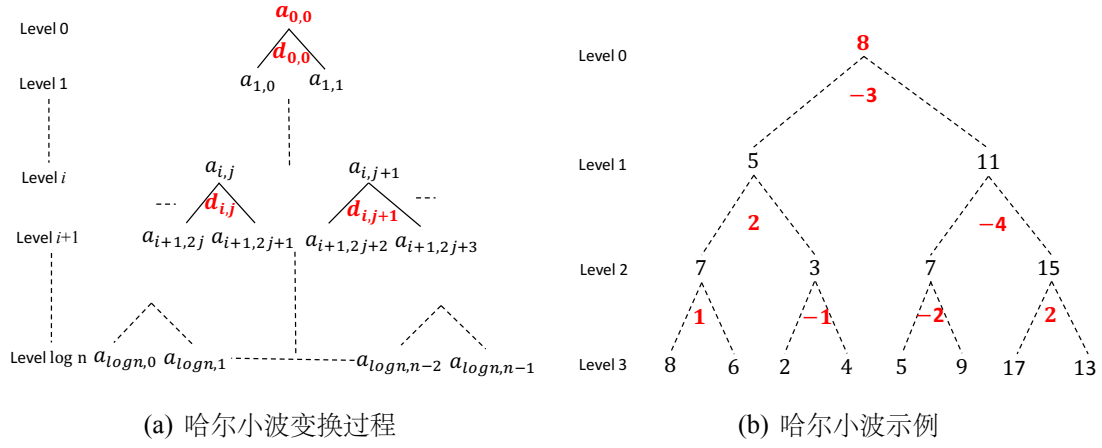


图 4.1: 哈尔小波变换

#### 4.1.2 基于哈尔小波的轨迹概要数据抽取

哈尔小波变换 (Haar Wavelet Transform, HWT) 是降维时间序列、图像数据等的有效方法。它将原始数据从多个解析度来展示, 每个解析度代表了不同频域下的信息。其变换过程又可以被抽象成如图4.1(a)所示的自底向上构建一颗二叉树 (称为误差树, Error-tree) 的过程。在该图的最底层 (叶子层) 自左往右是时间序列原始数据, 非叶子节点保留两个值  $a_i^j$  和  $d_i^j$ 。  $a_i^j$  记录了该结点两个孩子结点的均值的正则化均值, 即  $a_{i,j} = (a_{i+1,2j} + a_{i+1,2j+1})/nf$ 。  $d_i^j$  记录了该结点两个孩子结点均值的正则化差值信息, 即  $d_{i,j} = (a_{i+1,2j} - a_{i+1,2j+1})/nf$ 。  $nf$  称为正则化因子 (normalization factor), 其取值为  $1/2$  或  $1/\sqrt{2}$ , 具体是应用场景而定。由于最底层 (叶子节点层) 结点仅包含原始数据, 故倒数第二层结点直接从原始值计算出来。在自底向上构建的过程中, 每层结点个数减半, 且直到某层仅包含一个节点为止 (第 0 层)。

在误差树结构中, 每层的均值序列可以看做对原始时间序列的一层概要数据, 自上而下粒度越来越细。然而, 哈尔小波并没有直接保存这些均值, 它依次保留了最终的均值及由上到下每层的差值, 并将保留的值称为 (哈尔小波) 系数。我们根据系数值能恢复出每层的均值。具体的, 如图4.1(b)所示, 给定时间序列  $f(t) = \{8, 6, 2, 4, 5, 9, 17, 13\}$  且  $nf = 1/2$  时, 经过哈尔小波变换后的系数为  $H(f(t)) = \{8, -3, 2, -4, 1, -1, -2, 2\}$ 。给定第 0 层的系数 8, -3, 我们能计算出第 1 层的均值分别为  $8 + (-3) = 5, 8 - (-3) = 11$ 。此时, 我们仅使用了 2 个值就表达了与 3 个

值（第 0 和第 1 层均值）同等量的信息。此外，若继续给出第 1 层的系数，我们使用相同的方法能恢复出第 2 层的均值。值得注意的是，为方便解释，示例中正则化因子为  $1/2$ ，本章接下来的应用中将使用  $1/\sqrt{2}$  来进行变换。

上面部分介绍了哈尔小波变换以及如何使用它处理一维时间序列数据。但轨迹数据天然是多维时间序列数据，能否使用哈尔小波变换对轨迹数据进行多粒度解析呢？为此，本章重点讲介绍如何使用小波的多粒度特性来对轨迹数据进行概要数据抽取。首先，给定查询轨迹  $Q$  和候选候选轨迹  $C$ 。它们的长度均为  $n$ ，且假设  $n$  是 2 的正整数次方。由于长度为  $n$  的时间序列，其对应误差树的深度为  $L+1$ ， $L = \log_2 n$ 。对于  $Q$  和  $C$  的哈尔小波系数分别为  $HQ = \{a_{0,0}^Q, d_{0,0}^Q, d_{1,0}^Q, \dots, d_{L-1,n/2-1}^Q\}$  和  $HC = \{a_{0,0}^C, d_{0,0}^C, d_{1,0}^C, \dots, d_{L-1,n/2-1}^C\}$ ，其中  $a_{0,0}^Q$  和  $a_{0,0}^C$  分别是  $H(Q)$  和  $H(C)$  变换后的最终的正则化的均值， $d_{i,j}^Q$  和  $d_{i,j}^C$  是正则化后的差值，且  $a_{i,j}^Q, a_{i,j}^C, d_{i,j}^Q, d_{i,j}^C \in R^d$ 。根据正则化小波变换过程， $Q$  的误差树中第  $i$  层第  $j$  个非叶子节点的内容  $a_i^j$  和  $d_i^j$  可以由如下公式计算。

$$a_{i,j}^Q = \frac{a_{i+1,2j}^Q + a_{i+1,2j+1}^Q}{\sqrt{2}}, \quad d_{i,j}^Q = \frac{a_{i+1,2j}^Q - a_{i+1,2j+1}^Q}{\sqrt{2}} \quad (4.3)$$

## 4.2 轨迹欧式距离上下界

上节介绍了如何对轨迹数据使用哈尔小波进行轨迹变换，本节将介绍如何使用哈尔小波系数构建欧式距离的上下界。

### 4.2.1 基于哈尔小波的欧式距离表示

首先，对于  $Q$  和  $C$  的误差树中对应相邻节点对： $\{a_{i+1,2j}^Q, a_{i+1,2j+1}^Q\}$  和  $\{a_{i+1,2j}^C, a_{i+1,2j+1}^C\}$ ，则节点对相应元素的欧式距离的和可以如下表示：

$$\begin{aligned} & SED(a_{i+1,2j}^Q, a_{i+1,2j}^C) + SED(a_{i+1,2j+1}^Q, a_{i+1,2j+1}^C) \\ &= SED\left(\frac{a_{i,j}^Q + d_{i,j}^Q}{\sqrt{2}}, \frac{a_{i,j}^C + d_{i,j}^C}{\sqrt{2}}\right) + SED\left(\frac{a_{i,j}^Q - d_{i,j}^Q}{\sqrt{2}}, \frac{a_{i,j}^C - d_{i,j}^C}{\sqrt{2}}\right) \end{aligned} \quad (4.4)$$

$$\begin{aligned}
 &= \left\| \frac{\mathbf{a}_{i,j}^Q + \mathbf{d}_{i,j}^Q}{\sqrt{2}} \right\|^2 + \left\| \frac{\mathbf{a}_{i,j}^C + \mathbf{d}_{i,j}^C}{\sqrt{2}} \right\|^2 - 2 \frac{\mathbf{a}_{i,j}^Q + \mathbf{d}_{i,j}^Q}{\sqrt{2}} \cdot \frac{\mathbf{a}_{i,j}^C + \mathbf{d}_{i,j}^C}{\sqrt{2}} \\
 &\quad + \left\| \frac{\mathbf{a}_{i,j}^Q - \mathbf{d}_{i,j}^Q}{\sqrt{2}} \right\|^2 + \left\| \frac{\mathbf{a}_{i,j}^C - \mathbf{d}_{i,j}^C}{\sqrt{2}} \right\|^2 - 2 \frac{\mathbf{a}_{i,j}^Q - \mathbf{d}_{i,j}^Q}{\sqrt{2}} \cdot \frac{\mathbf{a}_{i,j}^C - \mathbf{d}_{i,j}^C}{\sqrt{2}} \\
 &= SED(\mathbf{a}_{i,j}^Q, \mathbf{a}_{i,j}^C) + SED(\mathbf{d}_{i,j}^Q, \mathbf{d}_{i,j}^C)
 \end{aligned}$$

公式4.4说明，细粒度的概要数据间的距离可以由粗粒度的数据计算出来。为进一步表示，我们用  $S_i(Q, C)$  表示  $Q$  和  $C$  误差树的第  $i$  层的对应均值间的欧式距离的平方和，用  $SED_i(Q, C)$  表示  $Q$  和  $C$  误差树的第  $i$  层的对应差间的欧式距离的平方和。 $S_i(Q, C)$  和  $SED_i(Q, C)$  的定义如下所示：

$$S_i(Q, C) = \sum_{j=0}^{2^i-1} SED(\mathbf{a}_{i,j}^Q, \mathbf{a}_{i,j}^C) \quad (4.5)$$

$$SED_i(Q, C) = \sum_{j=0}^{2^i-1} SED(\mathbf{d}_{i,j}^Q, \mathbf{d}_{i,j}^C) \quad (4.6)$$

此外，由于原始轨迹间的距离可以根据两个误差树的叶子计算出来，即  $SED_L(Q, C) = SED(Q, C)$ 。根据这一结论和以上定义，我们得到如下定理。

**引理 4.2.1.** 给定两条轨迹  $Q$  和  $C$ ， $HQ$  和  $HC$  分别表示  $Q$  和  $C$  经过哈尔小波变换后的系数序列。我们有如下结论： $SED(Q, C) = SED(HQ, HC)$ 。

引理4.2.1既说明了原始轨迹间的欧式距离等于哈尔小波系数间距离（证明见本章附件），又说明了通过对查询轨迹哈尔变换后的系数可以替换原始轨迹用于距离计算。此外，对于轨迹长度不是2的次方的情况，我们可以通过将轨迹切分成若干个字轨迹，只需保证每个轨迹是2的次方。然后，可以对子轨迹进行哈尔小波变换，并将所有子轨迹系数的欧式距离累加等到整体轨迹间的距离。因此，哈尔小波可以被用来分解任意长度的轨迹。

#### 4.2.2 基于哈尔小波的欧式距离上下界

上一节介绍了，利用完整的哈尔小波系数，即全部的概要数据，能够用来计算原始的轨迹间欧式距离。但全部概要数据的数据量于原始轨迹数据量相同，不

能达到降低通信开销的目的。为此本节将介绍如何利用部分概要来计算轨迹的上、下界。首先根据引理4.2.1, 我们有:

$$\begin{aligned} SED(Q, \mathcal{C}) &= \sum_{i=0}^{L-1} SED_i(Q, \mathcal{C}) \\ &= \sum_{i=0}^l SED_i(Q, \mathcal{C}) + \sum_{i=l+1}^{L-1} SED_i(Q, \mathcal{C}) \end{aligned} \quad (4.7)$$

假设远程结点由粗到细已经获取了查询轨迹  $Q$  的前  $l+1$  层概要数据, 我们的目标就是根据这些已有的数据计算出  $Q$  与任一候选  $\mathcal{C}$  的欧式距离上、下界。公式4.7有半部分分为两部分, 前半部分可以根据已有概要数据计算出来, 后半部分无法直接计算。为此, 我们将后半部分继续展开, 得到如下:

$$\sum_{i=l+1}^{L-1} SED_i(Q, B) = \sum_{i=l+1}^{L-1} \sum_{j=0}^{2^i-1} (\|d_{i,j}^Q\|^2 + \|d_{i,j}^C\|^2 - 2d_{i,j}^Q \cdot d_{i,j}^C) \quad (4.8)$$

公式4.8右半部分含有3个元素, 第一个元素为查询轨迹  $Q$  剩下概要数据的和, 该值可根据已有概要数据计算出来, 计算方法为  $SSQ - \sum_{i=0}^l \sum_{j=0}^{2^i-1} \|d_{i,j}^Q\|^2$ , 其中  $SSQ$  为  $Q$  所有概要数据的累加和。远程结点只需一开始就把  $SSQ$  发给所有远程结点即可。同理, 第二个元素各远程结点可以计算出来。难点在于对第三个关于系数内积累加和的计算。本文的做法是使用两次柯西—施瓦茨不等式估计其区间。第一次使用具体过程如下:

$$-2 \sum_{i=l+1}^{L-1} \sum_{j=0}^{2^i-1} \|d_{i,j}^Q\| \cdot \|d_{i,j}^C\| \leq \sum_{i=l+1}^{L-1} \sum_{j=0}^{2^i-1} -2d_{i,j}^Q \cdot d_{i,j}^C \leq 2 \sum_{i=l+1}^{L-1} \sum_{j=0}^{2^i-1} \|d_{i,j}^Q\| \cdot \|d_{i,j}^C\| \quad (4.9)$$

接着我们对  $\sum_{i=l+1}^{L-1} \sum_{j=0}^{2^i-1} \|d_{i,j}^Q\| \cdot \|d_{i,j}^C\|$  进行再次使用柯西-施瓦茨不等式放缩:

$$\sum_{i=l+1}^{L-1} \sum_{j=0}^{2^i-1} \|d_{i,j}^Q\| \cdot \|d_{i,j}^C\| \leq \sqrt{\sum_{i=l+1}^{L-1} \sum_{j=0}^{2^i-1} \|d_{i,j}^Q\|^2} \cdot \sqrt{\sum_{i=l+1}^{L-1} \sum_{j=0}^{2^i-1} \|d_{i,j}^C\|^2} \quad (4.10)$$

结合公式4.9和4.10, 我们得到如下完整的对  $\sum_{i=l+1}^{L-1} \sum_{j=0}^{2^i-1} -2d_{i,j}^Q \cdot d_{i,j}^C$  计算出如下上、

下界。

$$\begin{aligned}
 & -2\sqrt{\sum_{i=l+1}^{L-1} \sum_{j=0}^{2^i-1} \|d_{i,j}^{\mathcal{Q}}\|^2} \cdot \sqrt{\sum_{i=l+1}^{L-1} \sum_{j=0}^{2^i-1} \|d_{i,j}^{\mathcal{C}}\|^2} \\
 & \leq \sum_{i=l+1}^{L-1} \sum_{j=0}^{2^i-1} -2d_{i,j}^{\mathcal{Q}} \cdot d_{i,j}^{\mathcal{C}} \leq 2\sqrt{\sum_{i=l+1}^{L-1} \sum_{j=0}^{2^i-1} \|d_{i,j}^{\mathcal{Q}}\|^2} \cdot \sqrt{\sum_{i=l+1}^{L-1} \sum_{j=0}^{2^i-1} \|d_{i,j}^{\mathcal{C}}\|^2} \quad (4.11)
 \end{aligned}$$

最终，我们对欧式距离获得如下上、下界：

$$HLB_l(\mathcal{Q}, \mathcal{C}) = \sum_{i=0}^l SED_i(\mathcal{Q}, \mathcal{C}) + \sum_{i=l+1}^{L-1} \sum_{j=0}^{2^i-1} (\|d_{i,j}^{\mathcal{Q}}\|^2 + \|d_{i,j}^{\mathcal{C}}\|^2) + S_0(\mathcal{Q}, \mathcal{C}) \quad (4.12)$$

$$\begin{aligned}
 & -2\sqrt{\sum_{i=l+1}^{L-1} \sum_{j=0}^{2^i-1} \|d_{i,j}^{\mathcal{Q}}\|^2} \cdot \sqrt{\sum_{i=l+1}^{L-1} \sum_{j=0}^{2^i-1} \|d_{i,j}^{\mathcal{C}}\|^2} \\
 HUB_l(\mathcal{Q}, \mathcal{C}) &= \sum_{i=0}^l SED_i(\mathcal{Q}, \mathcal{C}) + \sum_{i=l+1}^{L-1} \sum_{j=0}^{2^i-1} (\|d_{i,j}^{\mathcal{Q}}\|^2 + \|d_{i,j}^{\mathcal{C}}\|^2) + S_0(\mathcal{Q}, \mathcal{C}) \\
 & + 2\sqrt{\sum_{i=l+1}^{L-1} \sum_{j=0}^{2^i-1} \|d_{i,j}^{\mathcal{Q}}\|^2} \cdot \sqrt{\sum_{i=l+1}^{L-1} \sum_{j=0}^{2^i-1} \|d_{i,j}^{\mathcal{C}}\|^2} \quad (4.13)
 \end{aligned}$$

进一步的我们提出两个性质，这两个性质表明我们的上、下界会随着概要数据的增加而越来越紧。它们的证明过程见本章附件。

**性质 4.2.2.**  $HLB$  会随着粒度的概要数据粒度的增加而逐渐上升，即  $HLB_l \leq HLB_{l+1}$ 。

**性质 4.2.3.**  $HUB$  会随着粒度的概要数据粒度的增加而逐渐下降，即  $HUB_l \geq HUB_{l+1}$ 。

### 4.3 基于欧式距离的查询算法:ED-FTB

#### 4.3.1 ED-FTB 算法实现

在上一节，我们已经利用概要数据提出了欧氏距离的上、下界。这使得在 FTB



**算法 5** *ED-FTB*在协调者结点

---

```

/* ED-FTB 协调者结点函数接口实现 */
coordinatorInit( $\mathcal{Q}, \mathcal{R}$ )
1:  $l \leftarrow -1$ ;
2:  $HQ \leftarrow$  Haar wavelet coefficients of  $\mathcal{Q}$ ;           # 待查询轨迹获取哈尔小波系数
3:  $SSQ = \sum_{i=0}^{n-1} \|HQ_i\|^2$ ;                         # 所有系数的平方和
4: sendToRemoteSites( $\mathcal{R}, SSQ$ );

generateInfo()
1:  $l \leftarrow l + 1$ ;
2: return  $\hat{Q}_l$ ;                                           # 返回第  $l$  层哈尔小波系数

```

---

框架中插入欧式距离成为可能。本节将详细介绍将欧式距离跟 **FTB** 相结合的查询算法 **ED-FTB**。**ED-FTB** 维持了 **FTB** 框架的主要结构，只对本章第一节所介绍的接口进行了具体实现。

算法5介绍了协调者节点的上的函数接口的实现方法。在 **coordinatorInit** 函数中，我们对待查询轨迹进行哈尔小波变换，并计算出其所有系数的平方和。接着将该平方和发送给所有远程结点。远程结点在将来将会利用该值进行上、下界计算。由于 **FTB** 框架是迭代式由粗到细的通信计算框架，在每轮的迭代中会将某一层的概要数据（哈尔小波系数）发送给远程结点。因此，在初始化过程中我们用  $l$  来记录当前已经发送到哪一层的数据，并将  $l$  初始化为  $-1$  以便从第 0 层开始。此外，我们在 **generateInfo** 中准备将要发送的下一层概要数据，以便协调者结点发送。

接着，算法6介绍运行在远程结点的函数。对于 **RemoteInit** 函数，若远程结点是第一次接受查询，则会为每条候选轨迹进行哈尔小波变换并计算系数的平方和。若不是，则为该结点所包含的每条轨迹初始化界特征信息（下界初始化为 0，上界初始化为正无穷），并接受查询轨迹的系数平方和。在查询执行过程中 **UpdateBounds** 函数为候选更新上、下界。直接实现方式为根据上、下界的计算公式，计算出这两个值。但由于查询过程中的主要计算开销就是对所有候选更新界特征。所以，降低该过程的计算开销很有必要。

为降低更新界特征的计算开销，本文引入了在更新界过程中进行剪枝的思想。在介绍此方法前，我们再次回顾下我们的下界。我们的下界计算主要包含 3 个算

**算法 6** *ED-FTB*在远程结点

---

```

/* ED-FTB 远程结点函数接口实现 */
remoteInit( $TS_r, S_r$ )
1: if 第一次接受查询 then
2:   for all  $C$  in  $TS_r$  do
3:      $HC \leftarrow$  Haar wavelet coefficients of  $C$ ;           # 候选轨迹获取哈尔小波系数
4:      $SSC = \sum_{i=0}^{n-1} \|HQ_i\|^2$ ;                         # 所有系数的平方和
5:   for all  $C$  in  $TS_r$  do
6:      $C\_BF \leftarrow \langle 0, \infty \rangle$ ;                     # 为  $C$  初始化界特征
7:      $S_r = S_r \cup C\_BF$ ;
8:    $SSQ \leftarrow$  getFromCoordinator();

updateBounds( $S_r, M$ ) //边更新边过滤。
1: for all  $C$  in  $TS_r$  do
2:    $a = \sum_{i=l+1}^{L-1} \sum_{j=0}^{2^i-1} \|d_{i,j}^Q\|^2$ ;  $b = \sum_{i=l+1}^{L-1} \sum_{j=0}^{2^i-1} \|d_{i,j}^C\|^2$ ;
3:    $tmp = a + b + S_0(Q, C)$ ;
4:    $lb = tmp - 2\sqrt{ab}$ ;  $ub = tmp + 2\sqrt{ab}$ ;
5:   if  $lb < gkub$  &&  $(lb = lb + \sum_0^l SED_i(Q, C)) < gkub$  then
6:     将轨迹  $C$  的界特征的下界更新为  $lb$ ;
7:     if  $ub < gkub$  &&  $(ub = ub + \sum_0^l SED_i(Q, C)) < gkub$  then
8:        $ub = ub + \sum_0^l SED_i(Q, C)$ ;
9:     将轨迹  $C$  的界特征的上界更新为  $ub$ ;
10:  else
11:    将该轨迹从  $S_r$  中移除;
    
```

---

子: 首先是  $\sum_{i=0}^l SED_i(Q, C)$ , 其计算复杂度为  $O(2^l)$ 。其次是  $\sum_{i=l+1}^{L-1} \sum_{j=0}^{2^i-1} \|d_{i,j}^Q\|^2$ , 由于该部分可以通过  $SSQ - \sum_{i=0}^l \sum_{j=0}^{2^i-1} \|d_{i,j}^Q\|^2$  计算, 而且  $\sum_{i=0}^{l-1} \sum_{j=0}^{2^i-1} \|d_{i,j}^Q\|^2$  的值已知。故该部分的计算复杂度也为  $O(2^l)$ 。同理, 第三部分  $\sum_{i=l+1}^{L-1} \sum_{j=0}^{2^i-1} \|d_{i,j}^C\|^2$  计算复杂度也为  $O(2^l)$ 。除这三个算子的算法外, 剩余部分的计算复杂度为  $O(1)$ 。

基于以上分析。假设我们已知若轨迹下界值超过  $\alpha$ , 那该条轨迹即不可能称为候选。此时, 我们可以在更新下界的同时进行剪枝 (算法6: **UpdateBounds** 函数)。具体的做法是我们首先计算出第二和第三两个算子, 并根据这两个算子的值计算出下界中除  $\sum_{i=0}^l SED_i(Q, C)$  以外部分的值。若此时计算出来的值已超过  $gkub$ , 则无需继续求解下界和上界, 直接将该候选删除。若仍小于  $gkub$ , 则计算出完整的下界, 并判断此时下界值是否小于  $gkub$ , 若小于则停止计算上界并将该候选删除。当更新完下界后, 我们先计算出上界中除  $\sum_{i=0}^l SED_i(Q, C)$  以外部分的值。若该

部分值超过  $gkub$ ，则该轨迹的上界对选取新一轮的全局最小的  $k$  个上界没有帮助，无需计算该轨迹的具体上界值。否则，计算出具体的上界并更新界特征。此时，需要注意的是，在 FTB 框架算法的第 7-9 行中，我们仅需传递上界小于  $gkub$  的  $k$  个最小上界。若个数不足  $k$  个，则只传递满足  $gkub$  的上界。

### 4.3.2 ED-FTB 算法性能分析

在分析前，我们先介绍对比算法 LEEWAVE-CL。LEEWAVE-CL 是在 LEEWAVE 算法的基础上使用了本文所提下界（比原始的下界更紧）。LEEWAVE-CL 算法也是迭代式算法。在它的每次迭代中，远程结点根据获取的概要数据，为每个候选计算如下两个算子： $\sqrt{\sum_{i=l+1}^{L-1} \sum_{j=0}^{2^i-1} \|\mathbf{d}_{i,j}^c\|^2}$  和  $S_0(\mathcal{Q}, \mathcal{C}) + \sum_{i=0}^l SED_i(\mathcal{Q}, \mathcal{C})$ 。然后协调者节点获取者些参数并为每个候选计算上、下界，并利用界特征进行过滤。然后，将候选列表发给对应远程结点。LEEWAVE-CL 方法与本文方法的最大不同就是，它在协这结点进行界特征的计算和过滤，而 ED-FTB 是在所有远程结点进行。接下来，我们将从时间和通信两个方面对 ED-FTB 和 LEEWAVE-CL 进行对比分析。在此之前，我们使用如下标记：(i) 使用  $|C_i|$  和  $|CS_i|$  分别表示第  $i$  ( $i \geq 0$ ) 次迭代前候选轨迹的数量和包含候选轨迹的远程结点数量。由于采用相同的上、下界，这两个算法的迭代次数相同，我们假定迭代次数为  $\lambda$  ( $\lambda \leq \log_2 n$ )。

**时间复杂度：**不考虑系统初始化时对所有候选进行哈尔小波变换的时间开销，ED-FTB 和 LEEWAVE-CL 两者的时间开销来自迭代式计算时更新上下界，所以他们的时间复杂度一样。计算复杂度最坏的情况就是所有候选都集中在一个结点上，则更新上、下界的时间复杂度为  $O(\sum_{i=0}^{\lambda-1} d \cdot |C_i| \cdot 2^i)$ 。此外，由于  $k$  值一般较小，算法运行过程中设计到的找  $\text{top}k$  上界的时间较低，可以忽略。总的来说，ED-FTB 和 LEEWAVE-CL 的时间复杂度均为  $O(\sum_{i=0}^{\lambda-1} d \cdot |C_i| \cdot 2^i)$ 。但 ED-FTB 由于在各个结点进行界特征的计算，且采用了边计算边过滤的策略，故其实际计算开销低于 LEEWAVE-CL。

**通信复杂度：**ED-FTB 和 LEEWAVE-CL 的通信开销主要集中在迭代式计算过

程中。ED-FTB 在第  $i$  轮迭代时, 需要花费  $O(d \cdot 2^i \cdot |CS_i| + |C_i|)$  代价以将第  $i$  层概要数据发送给候选结点, 并至多接收  $|C_i|$  个上界 (用于计算全局的剪枝阈值)。在  $\lambda$  次迭代后, 总的通信开销为  $O(\sum_{i=0}^{\lambda-1} (d \cdot 2^i \cdot |CS_i| + |C_i|))$ 。LEEWAVE-CL 在第  $i$  轮迭代需要花费  $O(|CS_i| \cdot (d \cdot 2^i + |C_{i+1}|))$  来讲概要数据和所有候选的列表发送给所有候选结点。此外, 还需要花费  $O(d \cdot |C_{i+1}|)$  代价以接受每个候选轨迹的两个算子值以便在协调者结点计算出上、下界。所以, 当  $\lambda$  次迭代后, 其总的通信开销为  $O(|CS_i| \cdot (d \cdot 2^i + |C_{i+1}|) + d \cdot |C_{i+1}|)$ 。对比两个算法的开销值, 我们可以发现 ED-FTB 的能节省更多通信开销。

## 4.4 实验分析

本节将在真实世界的轨迹数据上评估本章节所提出的算法。本小姐首先介绍使用的真实世界数据集合实验设置。接着从多个角度验证算法的性能<sup>(1)</sup>。

### 4.4.1 实验设置

本章工作在实验中使用北京出租车数据集, 该数据集已被广泛应用于轨迹数据分析。该数据集采集了北京市三万多辆出租车在 2013 年 10 月份到 12 月份三个月内的行驶 GPS 轨迹。每个 GPS 轨迹点包含的数据维度较多, 我们仅选取了位置 (经度和纬度)、时间、速度和角度这五个维度的值, 并进行了归一化预处理。我们从该数据集中截取了两个子数据集: *T-Small* 和 *T-Big* 以分别验证算法的有效性和可扩展性。*T-Small* 截取了 10 月 1 至 7 号间上午 8 点到 10 点的轨迹数据, 并从中选出最长的 1 万条轨迹进行分析。*T-Big* 截取了 11 月 1 号至 12 月 31 号间每天上午 8 至 10 点和下午 5 至 7 点两个时间段的 1 百万条轨迹数据。为满足实验需求, 这两个数据集的每条轨迹长度均超过 4,096。

在本章的实验中, 我们将比较 ED-FTB 和 LEEWAVE-CL 算法的性能。LEEWAVE-CL 算法在已有 LEEWAVE 算法的基础上使用了本文所提供的下界。两个算法均用

<sup>1</sup>扩展

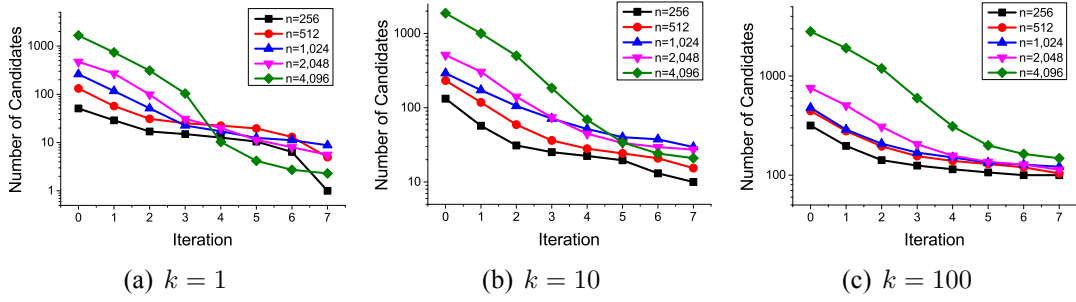


图 4.2: ED-FTB 剪枝效果图

JAVA 实现，并运行在一个包含 12 个结点的 Spark 集群上。每个结点包含 8 核英特尔 E5335 2.0 GHz 中央处理器和 16GB 的内存。我们在 *T-Small* 数据集上验证了算法的有效性，在 *T-Big* 数据集上验证了算法的可扩展性。

#### 4.4.2 算法有效性

首先，我们通过观察迭代过程中每轮结束后的剩余候选数来研究 ED-FTB 算法的剪枝效果。在该组实验中，我们将  $M$  设置为 10,000，并研究了不同长度轨迹剪枝的效果。图 4.2 介绍了前 8 轮迭代中每轮迭代后的候选数。我们可以看出候选数在前 5 轮迭代中下降的较快且已经过滤掉绝大多数候选。这说明了我们的上下级收敛的很快，具有较好的剪枝效果。此外，我们可以发现短轨迹在前几轮中候选数下降的比长轨迹快。这是由于相同数据量的概要数据下短轨迹比长轨迹包含更多的原始轨迹的信息。最后，对比不同的  $k$  值，我们发现  $k$  值越小，每轮所剩候选数越少。即  $k$  越小，剪枝效果越好。这是由于  $k$  值越小，我们所获得的全局第  $k$  小上界就越小。而无论  $k$  值如何选取，每个候选的下界值不变。因此，越小的全局上界能通过下界剪枝掉更多的候选。同时，由于实际应用中  $k$  值通常都是一个很小的数。因而我们的算法能往往取得较好的效果。

然后，我们对比研究了 ED-FTB 的剪枝效果和 ED-FTB 使用 LEEWAVE 中的下界后的剪枝效果，以分析下界对剪枝效果的影响。在该组实验中，我们将  $M$  值设置为 10,000，轨迹长度设为 1,024。图 4.3 对比展示了不同  $k$  值下，不同下界对剪枝的影响。其中 LEEWAVE 代表 ED-FTB 算法使用 LEEWAVE 算法中的下界后的

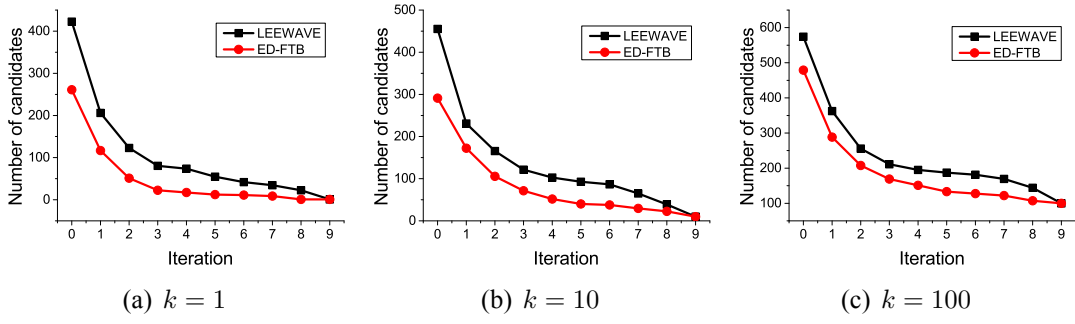
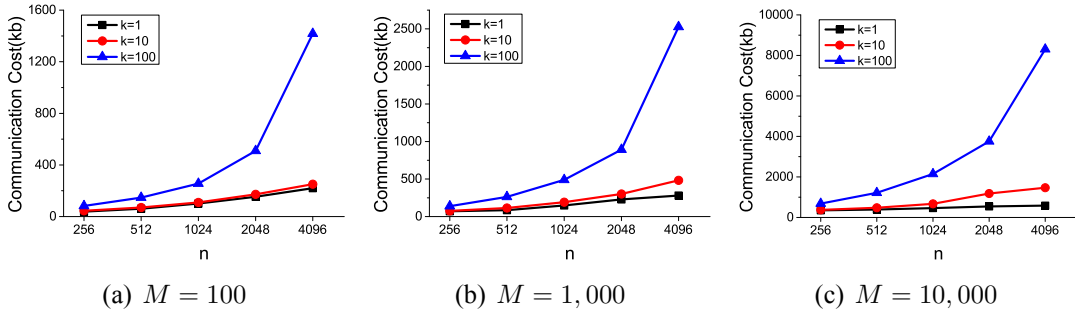


图 4.3: 下界对剪枝结果的影响


 图 4.4:  $n, k$  和  $M$  对 ED-FTB 算法通信开销的影响

结果。从图中，我们可以看出由于本文所提出的下界比已有算法的下界更紧，导致每轮迭代结束后 ED-FTB 所剩的候选数更少。这说明了本文所提下界的优越性。此外我们可以发现  $k$  值越小，两条曲线间的间隔越大。更能说明本文下界所取得的优越性越明显。

其次，我们研究了  $n, k$  和  $M$  对 ED-FTB 算法通信开销的影响。在该组实验中，我们将  $n$  的值从 256 变化到 4,096，将  $k$  的值从 1 变化到 100。图4.4展示了不同  $M$  值下（ $M$  从 100 编化到 10,000），ED-FTB 算法通信开销随参数变化的结果。首先，我们看出随着  $k$  值的增加，通信开销逐步增加。这是由于  $k$  值越大，每轮所剩候选数越多，候选所在的远程结点也就越多。从而每轮需要将概要数据发送到更多的候选结点中。其次，随着  $n$  值的增加通信开销也在增加。这是由于长轨迹需要发送更多的数据才能达到与短轨迹形同的剪枝效果。最后，对比三幅图可以发现，随着远程结点数据的增加，通信开销也在增加。这是由于在每轮迭代中，远程结点增加后，候选轨迹会分布到更多的结点中。因而，算法需要将概要数据发送到更多

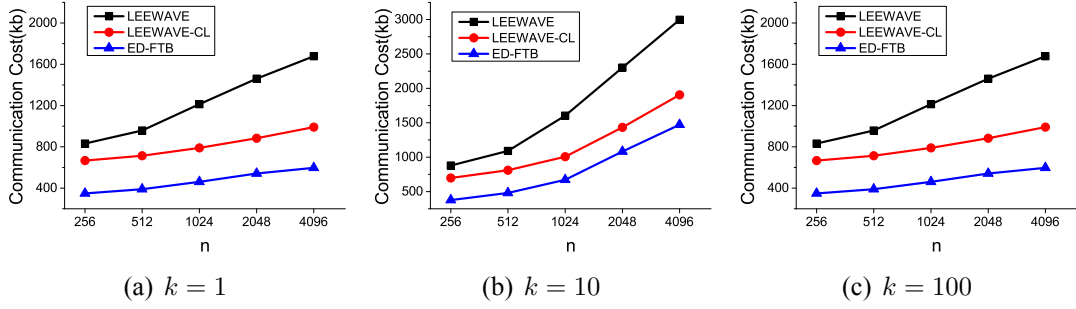


图 4.5: Communication cost comparison on T-Small

的结点中。因而总的通信开销也会增加。但是，对于极端情况  $M = 10,000$  时，此时每个远程结点仅包含一条轨迹时我们的通信开销仍不到 10Mb，远小于直接方法的开销（约 640Mb）。因此，ED-FTB 算法具有较高的通信性能。

接下来，我们将 ED-FTB 与 LEEWAVE 和 LEEWAVE-CL 进行通信性能比较。在该组实验中我们将轨迹长度  $n$  从 256 变化到 4,096。图4.5展示了当  $k$  取值为 1, 10 和 100 下的结果。从该组图中我们可以看出，本文所提 ED-FTB 算法性能最好，LEEWAVE-CL 其次，最差的是 LEEWAVE。LEEWAVE-CL 比 LEEWAVE 好的原因是它使用了本文所提下界进行剪枝。本文所提下界由于比 LEEWAVE 中的更加紧凑，因而剪枝效果更好。最终导致通信开销更低。而 ED-FTB 算法比 LEEWAVE-CL 好的原因是，本文采用的是在远程结点剪枝的策略，只需发送概要数据。而 LEEWAVE 和 LEEWAVE-CL 均是采用在协调者结点剪枝的策略，除了发概要数据还要发送和接受其他额外数据。因而开销更高。最后，我们可以发现，随着轨迹长度  $n$  和  $k$  值的增加，EDD-FTB 算法所节省的通信开销逐步增大。这进一步说明了本文算法的优越性。

#### 4.4.3 算法可扩展性

<sup>2</sup> 本小节我们将在 *T-Big* 数据集上从时间和通信两个角度研究了 ED-FTB 算法的可扩展性。首先，考虑了轨迹数量  $N$ （即数据量大小）和轨迹长度  $n$  对可扩展性的影响。在该组实验中，我们将  $k$  设置为 1， $M$  设置为 10,000。我们将轨迹长度

<sup>2</sup>重做实验

从 256 到 4096 进行指数级变化, 同时将轨迹数量从 10 万到 1 百万进行线性变换。图 4.6 分别对这两个角度的结果进行了介绍。其中图 4.6(a) 介绍了时间性能受  $n$  和  $N$  的影响。我们可以看到对于任意长度的轨迹数据集, ED-FTB 算法的运行时间都随着轨迹数据量的增加而线性增加。此外, 随着轨迹长度的指数增加, 算法的运行时间也指数增加, 即运行时间与轨迹长度呈一定的线性关系。这一结果反映了由于 ED-FTB 的剪枝效果较好, 导致迭代结束后, 所剩候选数较少。即我们对 ED-FTB 算法时间复杂度分析部分的  $N'$  较少。结合以上两点, 我们可以看出 ED-FTB 算法运行时间随着数据集的大小而线性变换, 因此运行效率具有较好的可扩展性。

#### 4.5 本章小结

本章节介绍了利用 FTB 框架在嵌入欧式距离下的具体实现算法 ED-FTB。本章节, 首先针对欧式距离, 提出了基于哈尔小波系数的概要数据。接着, 提出了基于 Haar 小波系数的欧式距离上、下界。该上、界能够随着小波粒度的增加而逐渐变紧, 这使得我们利用将该距离嵌入到 FTB 框架称为可能。然后, 我们提出了 ED-FTB 算法以实现基于欧式距离的查询。在我们的查询算法中, 我们引入了边计算下界边剪枝的查询优化措施, 以提高执行效率。通过在真实数据集上进行的实验, 表明所提方法剪枝效果由于现有方案, 且具有较好的可扩展性。

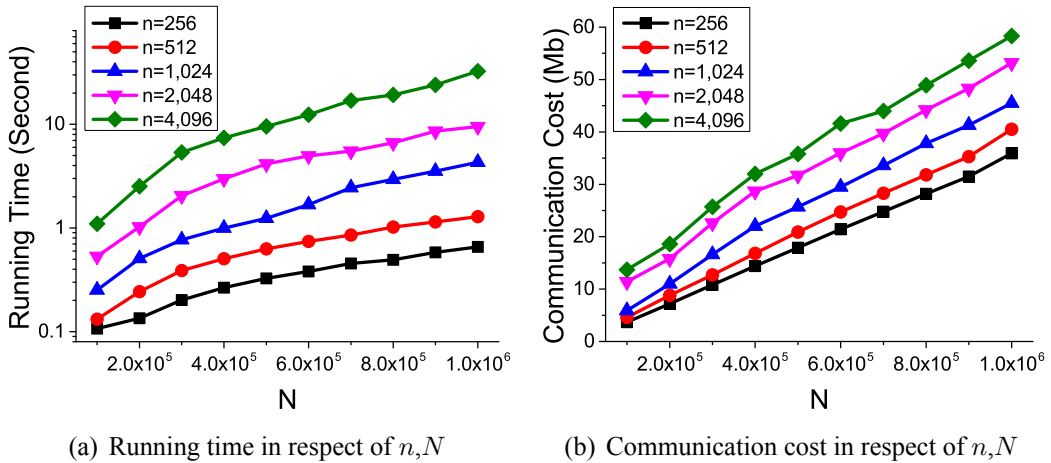


图 4.6: Scalability of ED-FTB on T-Big



## 4.6 附件

**引理 4.2.1.** 给定两条轨迹  $Q$  和  $C$ ,  $HQ$  和  $HC$  分别表示  $Q$  和  $C$  经过哈尔小波变换后的系数序列。我们有如下结论:  $SED(Q, C) = SED(HQ, HC)$ 。

**证明.** 首先, 根据  $S_i(Q, C)$  和  $SED_i(Q, C)$  定义 (定义4.5) 我们有

$$\begin{aligned}
 S_{i+1}(Q, C) &= \sum_{j=0}^{2^{i+1}-1} SED(a_{i+1,j}^Q, a_{i+1,j}^C) \\
 &= SED(a_{i+1,0}^Q, a_{i+1,0}^C) + SED(a_{i+1,1}^Q, a_{i+1,1}^C) + \cdots + \\
 &\quad SED(a_{i+1,2^{i+1}-2}^Q, a_{i+1,2^{i+1}-2}^C) + SED(a_{i+1,2^{i+1}-1}^Q, a_{i+1,2^{i+1}-1}^C) \\
 &= SED(a_{i,0}^Q, a_{i,0}^C) + SED(d_{i,0}^Q, d_{i,0}^C) + \cdots + \\
 &\quad SED(a_{i,2^i-1}^Q, a_{i,2^i-1}^C) + SED(d_{i,2^i-1}^Q, d_{i,2^i-1}^C) \\
 &= \sum_{j=0}^{2^i-1} SED(a_{i,j}^Q, a_{i,j}^C) + \sum_{j=0}^{2^i-1} SED(d_{i,j}^Q, d_{i,j}^C) \\
 &= S_i(Q, C) + SED_i(Q, C)
 \end{aligned} \tag{4.14}$$

根据如上公式, 对于叶子层结点, 我们有:

$$\begin{aligned}
 S_L(Q, C) &= S_0(Q, C) + \sum_{i=0}^{L-1} SED_i(Q, C) \\
 &= SED(HQ, HC)
 \end{aligned} \tag{4.15}$$

此外, 第  $L$  层为叶子节点层, 包含了轨迹的原始信息。所以我们又有  $SED(Q, C) = S_L(Q, C)$ 。此时结合公式4.15 我们得到  $SED(Q, C) = SED(HQ, HC)$ 。原问题得证。  $\square$

**Property 4.2.2.**  $HLB$  会随着粒度的概要数据粒度的增加而逐渐上升, 即  $HLB_l \leq HLB_{l+1}$ 。至此, 我们根据哈尔小波变换得到原始轨迹不同粒度概要数据, 并根据概要数据提出了欧式距离的上、下界。

证明. 我们的策略是证明  $HLB_{l+1} - HLB_l \geq 0$ . 我们首先将其左半部分展开:

$$HLB_{l+1} - HLB_l = SED_{l+1}(\mathcal{Q}, \mathcal{C}) - \sum_{j=0}^{2^{l+1}-1} (\|\mathbf{d}_{l+1,j}^{\mathcal{Q}}\|^2 + \|\mathbf{d}_{l+1,j}^{\mathcal{C}}\|^2) + 2\left(\sqrt{\sum_{i=l+1}^{L-1} \sum_{j=0}^{2^i-1} \|\mathbf{d}_{i,j}^{\mathcal{Q}}\|^2} \cdot \sqrt{\sum_{i=l+1}^{L-1} \sum_{j=0}^{2^i-1} \|\mathbf{d}_{i,j}^{\mathcal{C}}\|^2} - \sqrt{\sum_{i=l+2}^{L-1} \sum_{j=0}^{2^i-1} \|\mathbf{d}_{i,j}^{\mathcal{Q}}\|^2} \cdot \sqrt{\sum_{i=l+2}^{L-1} \sum_{j=0}^{2^i-1} \|\mathbf{d}_{i,j}^{\mathcal{C}}\|^2}\right)$$

接着, 我们将  $SED_{l+1}(\mathcal{Q}, \mathcal{C})$  展开得到  $SED_{l+1}(\mathcal{Q}, \mathcal{C}) = \sum_{j=0}^{2^{l+1}-1} (\|\mathbf{d}_{l+1,j}^{\mathcal{Q}}\|^2 + \|\mathbf{d}_{l+1,j}^{\mathcal{C}}\|^2 - 2\mathbf{d}_{l+1,j}^{\mathcal{Q}} \cdot \mathbf{d}_{l+1,j}^{\mathcal{C}})$ . 我们的问题变为证明如下不等式成立。

$$\begin{aligned} \sum_{j=0}^{2^{l+1}-1} \mathbf{d}_{l+1,j}^{\mathcal{Q}} \cdot \mathbf{d}_{l+1,j}^{\mathcal{C}} &\leq \sqrt{\sum_{i=l+1}^{L-1} \sum_{j=0}^{2^i-1} \|\mathbf{d}_{i,j}^{\mathcal{Q}}\|^2} \cdot \sqrt{\sum_{i=l+1}^{L-1} \sum_{j=0}^{2^i-1} \|\mathbf{d}_{i,j}^{\mathcal{C}}\|^2} \\ &\quad - \sqrt{\sum_{i=l+2}^{L-1} \sum_{j=0}^{2^i-1} \|\mathbf{d}_{i,j}^{\mathcal{Q}}\|^2} \cdot \sqrt{\sum_{i=l+2}^{L-1} \sum_{j=0}^{2^i-1} \|\mathbf{d}_{i,j}^{\mathcal{C}}\|^2} \end{aligned} \quad (4.16)$$

对于不等式 4.16 的左半部分我们有  $\sum_{j=0}^{2^{l+1}-1} \mathbf{d}_{l+1,j}^{\mathcal{Q}} \cdot \mathbf{d}_{l+1,j}^{\mathcal{C}} \leq \sqrt{\sum_{j=0}^{2^{l+1}-1} \|\mathbf{d}_{l+1,j}^{\mathcal{Q}}\|^2} \cdot \sqrt{\sum_{j=0}^{2^{l+1}-1} \|\mathbf{d}_{l+1,j}^{\mathcal{C}}\|^2}$ . 所以我们的目标变为证明  $\sqrt{\sum_{j=0}^{2^{l+1}-1} \|\mathbf{d}_{l+1,j}^{\mathcal{Q}}\|^2} \cdot \sqrt{\sum_{j=0}^{2^{l+1}-1} \|\mathbf{d}_{l+1,j}^{\mathcal{C}}\|^2}$  小于不等式 4.16 的右半部分. 为方便表示, 我们令  $x = \sum_{j=0}^{2^{l+1}-1} \|\mathbf{d}_{l+1,j}^{\mathcal{Q}}\|^2$ ,  $y = \sum_{j=0}^{2^{l+1}-1} \|\mathbf{d}_{l+1,j}^{\mathcal{C}}\|^2$ ,  $\alpha = \sum_{i=l+2}^{L-1} \sum_{j=0}^{2^i-1} \|\mathbf{d}_{i,j}^{\mathcal{Q}}\|^2$ ,  $\beta = \sum_{i=l+2}^{L-1} \sum_{j=0}^{2^i-1} \|\mathbf{d}_{i,j}^{\mathcal{C}}\|^2$ . 则不等式 4.16 等价于:

$$\sqrt{x \cdot y} + \sqrt{\alpha \cdot \beta} \leq \sqrt{(\alpha + x) \cdot (\beta + y)} \quad (4.17)$$

我们将如上不等式两边平方得到如下不等式:

$$2\sqrt{x \cdot y \cdot \alpha \cdot \beta} \leq \alpha \cdot y + \beta \cdot x \quad (4.18)$$

根据算数-几何平均不等式, 我们得不等式 4.18 成立. 原问题得证.  $\square$

**Property 4.2.3.**  $HLB$  会随着粒度的概要数据粒度的增加而逐渐上升, 即  $HLB_l \leq HLB_{l+1}$ .

证明. 性质4.2.2的证明过程与性质4.2.2类似, 故不再赘述。

□



## 第五章 FLB 框架的应用

本章主要介绍如何使用 FLB 框架处理基于动态时间卷曲距离的查询。首先, 章节5.1介绍了利用包围信封为 DTW 距离提供概要数据。然后, 章节5.2介绍了基于包围信封的 DTW 距离下界。其次, 章节5.3介绍了算法 DTW-FLB, 以处理当使用动态时间卷曲距离作为距离度量准则时的查询。接着, 章节5.3展示了 DTW-FLB 算法的有效性和可扩展性。再其次, 章节 5.5小结本章的研究内容。最后, 本章涉及到的证明内容放在附件部分。

### 5.1 基于动态时间卷曲距离的概要数据

#### 5.1.1 基于动态时间卷曲距离的轨迹相似度量

给定查询轨迹  $Q$  表示为  $Q = \{q_0, q_1, \dots, q_{n-1}\}$ , 以及一条候选轨迹  $C$  表示为  $C = \{c_0, c_1, \dots, c_{n-1}\}$ , 其中  $n$  为轨迹长度, 每个轨迹点来自  $d$  维空间, 即  $q_i, c_i \in R^d$ 。为使用动态时间卷曲距离匹配  $Q$  和  $C$ , 我们构建了一个  $n \times n$  的代价矩阵  $cost$ 。其中第  $i$  行第  $j$  列的元素记录了使用欧式距离匹配  $q_i$  和  $c_j$  两点之间的代价。动态时间卷曲距离的目的就是发现一条从如图5.1所示左下角出发到右上角结束的代价最小的路线/卷曲路径。一条卷曲路径需满足如下 3 个特别的约束:

- **边界条件:** 路径从代价矩阵的  $[0, 0]$  号格子出发并在  $[n-1, n-1]$  号格子结束。
- **连续性:** 如果路径中相邻的两次权重值分别取自代价矩阵的  $[i, j]$  和  $[i', j']$  两个单元, 那么这两个单元的位置关系必满足如下条件:  $i' - i \leq 1$  和  $j' - j \leq 1$ 。这个约束使得路径在扩展过程中只能从一个格子扩展到其半径为 1 周边的格子。
- **单调性:** 如果路径中相邻的两次权重值分别取自代价矩阵的  $[i, j]$  和  $[i', j']$  两个单元, 那么这两个单元的位置关系还须满足如下条件:  $i' - i \geq 0$  和  $j' - j \geq 0$ 。

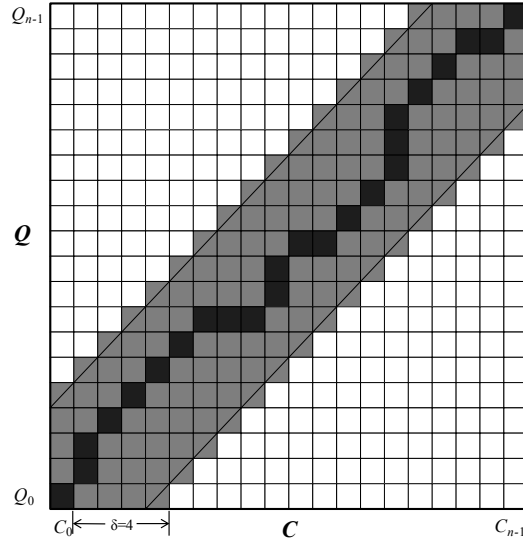


图 5.1: 带约束的动态时间卷曲

这个约束使得路径搜索过程中只能前进不能后退。

根据以上描述，我们形式化给出动态时间卷曲距离的定义：

$$DTW(Q, C) = \min \sum_{k=1}^K w_k \quad (n \leq K < 2n) \quad (5.1)$$

其中， $w_k$  是卷曲路径  $\mathcal{W}$  的第  $k$  个元素，其值来源于代价矩阵的第  $[i, j]$  个格子。

$\mathcal{W} = \{w_1, w_2, \dots, w_K\}$ , 代表了一条匹配  $Q$  和  $C$  卷曲路径。

动态时间卷曲距离的计算过程可通过动态规划算法实现，因为其计算过程可以看作如下最优子结构的递归：

$$DTW = \gamma(n-1, n-1) \quad (5.2)$$

$$\begin{aligned} \gamma(i, j) &= SED(\mathbf{q}_i, \mathbf{c}_j) + \min(\gamma(i-1, j-1), \gamma(i-1, j), \\ &\quad \gamma(i, j-1)) \quad s.t. \quad |i-j| \leq \delta \end{aligned} \quad (5.3)$$

其中  $\delta$  是一个全局约束，用于限制查询路径的可以偏离对角线的范围。图5.1中显示了当  $\delta = 4$  时，灰色部分即为卷曲路径的查询范围。使用全局约束的原因有两个：(i) 该约束可以提高查询的效率， $\delta$  使得动态时间卷曲距离的计算复杂度由原

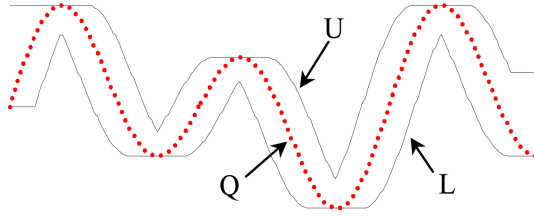


图 5.2: 包围信封

来的  $n^2$  降低为  $\frac{n^2}{\delta}$ 。(ii) 该约束还能有效抑制无效的卷曲路径的产生。比如，一轨迹的一小段匹配到另一轨迹的一大段上情况。

### 5.1.2 基于包围信封的概要数据

包围信封 (Bounding Envelope, BE) 概念来源于时间序列数据分析，其目的是使用一上一下两条边界曲线来包住给定的时间序列。如图5.2所示，红色虚线为一条时间序列， $U$  和  $L$  两条线分别代表了这条时间序列的上界和下界。给定时间序列，构造其包围信封的方法有许多。由于本文的目的是利用包围信封构造概要数据，故所构造的包围信封包含的数据量要低。此外，所构造的包围信封还应具有多粒度特性，以适应 FLB 框架的需求。

为此，引入基于时间序列划分 (Time Series Segmentation) 的包围信封。其做法是给定一条时间序列  $S$ ，将其均匀划分成多个等长度的片段。我们使用  $s_l^p = [lb, ub, len]$  来表示其中一条片段  $\{q_i, q_{i+1}, \dots, q_j\}$  的最小包围盒 (Minimum Bounding Rectangle, MBR)。其中  $l$  代表划分的粒度 (粒度越细，每个片段的长度越短)， $p$  代表该片段在当前划分下所对应的位置。 $lb$  和  $ub$  分别代表该片段的最小值和最大值， $len$  代表该片段的长度。由  $\{s_l^1, \dots, s_l^{n'}\}$  所构成的列表代表了该次划分所对应的包围信封，其用  $S_l$  表示。在第 0 层，整个时间序列被看做一个划分，接着将其均匀切成  $R$  个不相交的相连子片段。这个过程可以一直持续下去，直到每个片段的长度为 1 截止。

表格5.1介绍了给定时间序列  $S = \{5, 2, 4, 3\}$ ，计算其不同粒度包围盒的过程。其中， $R$  的值设置为 2。即从粗粒度到细粒度切分的过程中，每次将一个片段均匀

|                 |                     |                     |                     |                     |
|-----------------|---------------------|---------------------|---------------------|---------------------|
| $\mathcal{S}_0$ | $s_0^0 = [2, 5, 4]$ |                     |                     |                     |
| $\mathcal{S}_1$ | $s_1^0 = [2, 5, 2]$ |                     | $s_1^1 = [3, 4, 2]$ |                     |
| $\mathcal{S}_2$ | $s_2^0 = [5, 5, 1]$ | $s_2^1 = [2, 2, 1]$ | $s_2^2 = [4, 4, 1]$ | $s_2^3 = [3, 3, 1]$ |
| $Q$             | 5                   | 2                   | 4                   | 3                   |

 表 5.1: 时间序列  $\mathcal{S}$  的多粒度包围盒

划分成 2 个子片段。不失一般性，接下来的示例和实验室中，我们都将  $R$  的值设置为 2。值得注意的是，本文采用的划分为均匀划分，而相关文章中所提非均匀划分的方法也可以被使用。此外，由于是均匀划分，我们可以推算出每个片段的长度，故其值无需保存。

## 5.2 动态时间卷曲距离的下界

本节将介绍如何根据包围信封计算动态时间卷曲距离的下界。首先将介绍点的上、下界，然后再介绍如何利用点的范围计算轨迹的下界。

### 5.2.1 满足 DTW 约束的包围信封及下界

令  $\mathbf{q}$  和  $\mathbf{c}$  表示两个来自  $d$  维实数空间的点，其中  $\mathbf{c}$  的值已知， $\mathbf{q}$  的值不知道，但知道其在一个  $d$  维矩形  $\{\mathbf{u}, \mathbf{l}\}$  里，即  $\forall j \in [0, d), \mathbf{l}^j \leq \mathbf{q}^j \leq \mathbf{u}^j$ 。对  $\mathbf{q}$  和  $\mathbf{c}$  之间的实际欧式距离我们无法计算出来，但可以通过如下引理计算出距离的下界。

**引理 5.2.1.** 给定  $\mathbf{q}$ 、 $\mathbf{c}$  以及  $\mathbf{q}$  的包围矩形  $\{\mathbf{u}, \mathbf{l}\}$ ，我们有  $SED\_LB(\{\mathbf{u}, \mathbf{l}\}, \mathbf{c}) \leq SED(\mathbf{q}, \mathbf{c})$ ，其中  $SED\_LB(\{\mathbf{u}, \mathbf{l}\}, \mathbf{c})$  的计算过程如下：

$$SED\_LB(\{\mathbf{u}, \mathbf{l}\}, \mathbf{c}) = \sum_{j=0}^{d-1} \begin{cases} (\mathbf{c}^j - \mathbf{u}^j)^2 & \text{if } \mathbf{u}^j \leq \mathbf{c}^j, \\ 0 & \text{if } \mathbf{l}^j \leq \mathbf{c}^j \leq \mathbf{u}^j, \\ (\mathbf{l}^j - \mathbf{c}^j)^2 & \text{if } \mathbf{c}^j \leq \mathbf{l}^j. \end{cases} \quad (5.4)$$

接着介绍如何构建满足动态时间卷曲距离的包围信封。根据包围信封的介绍，我们知道给定查询轨迹  $Q$ ，构建包围信封的最重要就是构造轨迹的上、下边界。同



时，考虑到动态时间卷曲距离的全局约束特性。我们使用如下方法构建轨迹的上、下边界：We next introduce how to construct a bounding envelope  $\{\mathcal{U}, \mathcal{L}\}$  for trajectory  $\mathcal{Q}$ , as defined below.

$$\begin{aligned}\mathcal{U} &= \{\mathbf{u}_0, \mathbf{u}_1, \dots, \mathbf{u}_{n-1}\} \quad , \quad \forall j \mathbf{u}_i^j = \max(\mathbf{q}_{i-\delta}^j : \mathbf{q}_{i+\delta}^j) \\ \mathcal{L} &= \{\mathbf{l}_0, \mathbf{l}_1, \dots, \mathbf{l}_{n-1}\} \quad , \quad \forall j \mathbf{l}_i^j = \min(\mathbf{q}_{i-\delta}^j : \mathbf{q}_{i+\delta}^j)\end{aligned}\quad (5.5)$$

公式5.5中， $\mathcal{U}$  和  $\mathcal{L}$  分别代表轨迹的上边界和下边界， $\delta$  为动态时间卷曲距离中的全局约束。根据如上方式计算出来的包围盒，我们称之为满足动态时间卷曲约束的包围信封（Dynamic Time Warping constraint Bounding Envelope, DTWBE）。此时，根据 DTWBE $\{\mathcal{U}, \mathcal{L}\}$  和  $\mathcal{C}$ ，我们可以给出如下距离定义：

$$LB\_Keogh(\{\mathcal{U}, \mathcal{L}\}, \mathcal{C}) = \sum_{i=0}^{n-1} SED\_LB(\{\mathbf{u}_i, \mathbf{l}_i\}, \mathbf{c}_i) \quad (5.6)$$

该定义可以看做是原始一维的  $LB\_Keogh$  下界在多维情况下的扩展，并得到如下下界定理：

**定理 5.2.2.** 给定轨迹  $\mathcal{C}$  和待查询轨迹  $\mathcal{Q}$  满足动态时间卷曲约束的包围信封  $\{\mathcal{U}, \mathcal{L}\}$ ，我们有如下结论： $LB\_Keogh(\{\mathcal{U}, \mathcal{L}\}, \mathcal{C}) \leq DTW(\mathcal{Q}, \mathcal{C})$ 。

### 5.2.2 基于多粒度包围信封的下界

为计算上一小节介绍的  $LB\_Keogh$  下界，需要传递 DTWBE，其数据量为  $2*n$ 。因此，不满足我们降低通信的目标。为此，本文首先设计了基于 DTWBE 的多粒度包围信封。多粒度信封的第  $l$  层边界计算方式如下：

$$\begin{aligned}\hat{\mathcal{U}}_l &= \{\hat{\mathbf{u}}_{l,0}, \hat{\mathbf{u}}_{l,1}, \dots, \hat{\mathbf{u}}_{l,2^l-1}\} \quad , \quad \forall j \hat{\mathbf{u}}_{l,i}^j = \max(\mathbf{u}_{i \cdot 2^{L-l}}^j : \mathbf{u}_{(i+1) \cdot 2^{L-l}-1}^j) \\ \hat{\mathcal{L}}_l &= \{\hat{\mathbf{l}}_{l,0}, \hat{\mathbf{l}}_{l,1}, \dots, \hat{\mathbf{l}}_{l,2^l-1}\} \quad , \quad \forall j \hat{\mathbf{l}}_{l,i}^j = \min(\mathbf{l}_{i \cdot 2^{L-l}}^j : \mathbf{l}_{(i+1) \cdot 2^{L-l}-1}^j)\end{aligned}\quad (5.7)$$

第  $l$  层的包围信封计算过程可看作将原来的 DTWBE 均匀切分成  $2^l$  块，然后对每个信封块计算仅使用两个  $d$  维点来构成新的包围信封。根据对公式 5.7 的观察，我们得到两种结论：(i) 第  $l$  层的包围信封  $\{\hat{\mathcal{U}}_l, \hat{\mathcal{L}}_l\}$  数据量是第  $l-1$  层的两倍；(ii) 第  $l-1$  层的数据被第  $l$  层包含。因此，在由粗到细发送包围信封这一概要数据时，除第 0 层外，其他层只需要发送一半的信息量（这一半数据的每个值需要用 0 和 1 标注其位置是在左边还是右边，但位置信息消耗的开销较小可忽略）。那么假定远程结点已获取到查询轨迹  $\mathcal{Q}$  的第  $l$  层包围信封  $\{\hat{\mathcal{U}}_l, \hat{\mathcal{L}}_l\}$ ，我们可通过如下公式来计算给定的候选与  $\mathcal{Q}$  的下界。

**定理 5.2.3.** 给定轨迹  $\mathcal{Q}$  的第  $l$  层包围信封  $\{\hat{\mathcal{U}}_l, \hat{\mathcal{L}}_l\}$  和候选轨迹  $\mathcal{C}$ ，我们得到如下结论： $LB\_HPAA(\{\hat{\mathcal{U}}_l, \hat{\mathcal{L}}_l\}, \bar{\mathcal{C}}_l) \leq DTW(\mathcal{Q}, \mathcal{C})$ 。其中  $LB\_HPAA$  的计算公式如下：

$$LB\_HPAA(\{\hat{\mathcal{U}}_l, \hat{\mathcal{L}}_l\}, \bar{\mathcal{C}}_l) = 2^{L-l} \cdot \sum_{i=0}^{2^l-1} SED\_LB(\{\hat{\mathbf{u}}_{l,i}, \hat{\mathbf{l}}_{l,i}\}, \bar{\mathbf{c}}_{l,i})$$

以上定理说明了给定查询轨迹的第  $l$  层包围信封和候选误轨迹差树的第  $l$  层均值，就能为原始轨迹和候选轨迹计算出动态时间卷曲距离的下界。同时该下界的计算复杂度是线性的，远小于距离本身二次的复杂度。因此，通过计算下界可为剪枝候选提供帮助。

此外，为配合 FLB 框架，我们需要证明， $LB\_HPAA$  能随着粒度的增加而逐渐变紧。为此我们给出如下定理：

**定理 5.2.4.**  $LB\_HPAA$  下界能随着查询轨迹包围信封层次的增加而逐渐变紧。也就是说： $LB\_HPAA(\{\hat{\mathcal{U}}_l, \hat{\mathcal{L}}_l\}, \bar{\mathcal{C}}_l) \leq LB\_HPAA(\{\hat{\mathcal{U}}_{l+1}, \hat{\mathcal{L}}_{l+1}\}, \bar{\mathcal{C}}_{l+1})$ 。

## 5.3 基于动态时间距离的查询算法:DTW-FLB

### 5.3.1 DTW-FLB 算法实现

在上一节，我们已经利用多粒度概要数据得到越来越紧的动态时间卷曲距离下界。这使得在 FLB 框架中插入欧式距离成为可能。本节将详细介绍动态时间卷

**算法 7 DTW-FLB 在协调者节点**


---

```

/* DTW-FLB 协调者结点函数接口实现 */
coordinatorInit( $\mathcal{Q}, \mathcal{R}$ )
1: 根据公式5.5构建 DTWBE  $\{\mathcal{U}, \mathcal{L}\}$ ;
2: for  $i = 1, i < \log_2 n; i++$  do
3:   根据公式5.7构建第  $i$  层包围信封  $\{\hat{\mathcal{U}}_i, \hat{\mathcal{L}}_i\}$ ;
4:  $l \leftarrow -1$ ;
generateInfo()
1:  $l \leftarrow l + 1$ ;
2: return  $\{\hat{\mathcal{U}}_l, \hat{\mathcal{L}}_l\}$ ;

```

---

曲距离跟 FLB 相结合的查询算法 DTW-FLB。DTW-FLB 维持了 FLB 框架的主要结构，只对本章第一节所介绍的接口进行了具体实现。

我们首先介绍如何实现那些运行在协调者节点上的函数。在 **coordinatorInit** 函数中，我们首先对待查询轨迹计算出对应的 DTWBE，接着基于 DTWBE，我们构造多粒度的包围信封（即概要数据）。由于 FLB 框架也是迭代式由粗到细的通信计算框架，在每轮的迭代中会将某一层的概要数据（包围信封）发送给远程结点。因此，在初始化过程中我们用  $l$  来记录当前已经发送到哪一层的数据，并将  $l$  初始化为  $-1$ ，以便从第 0 层开始发送。此外，我们在 **generateInfo** 中准备将要发送的下一层概要数据，以便协调者结点发送。在发送数据时需要注意的是，由于  $\{\hat{\mathcal{U}}_l, \hat{\mathcal{L}}_l\}$  的一半数据量已经包含在  $\{\hat{\mathcal{U}}_{l-1}, \hat{\mathcal{L}}_{l-1}\}$  中，我们只需传输剩下的一半数据。

接着，介绍运行在远程结点的函数。对于 **RemoteInit** 函数，若远程结点是第一次接受查询，则会为每条候选轨迹进行哈尔小波变换。若不是，则为该结点所包含的每条轨迹初始化界特征信息（下界初始化为 0，不记录上界）。在查询执行过程中 **UpdateBounds** 函数为候选更新下界。直接实现方式为根据下界的计算公式直接算出该值。但由于查询过程中的主要计算开销就是对所有候选更新界特征。因此，降低该过程的计算开销很有必要。为降低更新界特征的计算开销，与上一章一样采用在更新界过程中进行剪枝的思想。根据  $LB\_HPAA$  的定义，我们可知该下界是一系列点距离下界的累加和。由于每个点距离下界都是大于 0 的数，因此我们可以在累加的过程中，通过判断当前已经累加的值是否已经超过剪枝的阈值。若

**算法 8 DTW-FLB 在远程结点**


---

```

/* DTW-FLB 远程结点函数接口实现 */
remoteInit( $TS_r, S_r$ )
1: if 第一次接受查询 then
2:   构建 R-Tree 索引。
3:   for all  $C$  in  $TS_r$  do
4:      $\bar{C} \leftarrow C$  的均值序列;           # 候选轨迹获取所有层的均值构成的序列
5:   for all  $C$  in  $TS_r$  do
6:      $C\_BF \leftarrow 0$ ;                 # 为  $C$  初始化下界
7:      $S_r = S_r \cup C\_BF$ ;
updateBounds( $S_r, m$ )      / *  $m = \{\hat{u}_{l,i}, \hat{l}_{l,i}\}$  * /
1: if  $l == 0$  then
2:   使用索引树进行剪枝。
3:   for all  $C$  in  $TS_r$  do
4:      $factor = 2^{L-l}$ ;
5:      $c.lb = 0$ ;
6:     for  $i = 0, i < 2^l; i++$  do
7:        $p\_lb = SED\_LB(\{\hat{u}_{l,i}, \hat{l}_{l,i}\}, \bar{c}_{l,i})$ ;
8:        $c.lb += p\_lb$ ;
9:       if  $c.lb > \theta$  then
10:        将该轨迹从  $S_r$  中移除;
11:   将  $C$  的下界更新为  $c.lb$ ;

```

---

超过了则可提前将该值过滤掉。

除了以上步骤外，我们引入了 R-tree 来对轨迹构建索引以提高查询效率。具体地：首先，每个远程结点对自己局部轨迹数据构建 R-tree 索引。接着，当该结点第 0 层包围信封时，使用 R-tree 来剪枝。剪枝具体过程如下：(i) 遍历 R-tree，找到跟查询轨迹包围信封相交的叶子节点。(ii) 遍历这些叶子节点所包含的轨迹，并将这些轨迹列为候选。

### 5.3.2 DTW-FLB 算法性能分析

本章节将从时间复杂度和空间复杂度量方面对 DTW-FLB 进行性能分析。在此之前，我们仍使用  $|C_i|$  和  $|CS_i|$  分别表示第  $i$  ( $i \geq 0$ ) 次迭代前候选轨迹的数量和包含候选轨迹的远程结点数量，并假定迭代次数为  $\lambda$  ( $\lambda \leq \log_2 n$ )。由于仅使用下界进行剪枝，因此存在着当概要数据发送完毕后，仍然存在着超过  $k$  个候选的

情况。此时，我们使用  $N'$  来表示迭代计算结束后剩余候选的个数。

**时间复杂度：**不考虑系统初始化时对所有候选进行哈尔小波变换的时间开销，DTW-FLB 算法的时间主要来自两个方面：(i) 算法 3 第 2 阶段的迭代更新的时间；(ii) 对迭代结束后的候选计算真实动态时间卷曲距离的时间。第一方面的时间复杂度为  $\sum_{i=0}^{\lambda} d \cdot |C_i| \cdot 2^i$ ，该值与算法 ED-FTB 的迭代计算复杂度相同。第二方面的计算复杂度为  $O(d \cdot \delta \cdot N' \cdot n^2)$ 。因此，总的时间复杂度为  $O(d \cdot (\sum_{i=0}^{\lambda} |C_i| \cdot 2^i + \delta \cdot N' \cdot n^2))$ 。

**通信复杂度：**DTW-FLB 算法的通信开销跟时间开销一样也是来自迭代发送包围信封和迭代结束后发送轨迹计算真实距离两个阶段。在迭代式发送包围阶段，第  $i$  次迭代中，协调者需要将第  $i$  层包围信封发送到候选结点中。由于，第  $i$  层包围信封已经有一半数据存在于第  $i-1$  层包围信封中且已被候选结点接受。因此，发送的数据量为  $O(d \cdot 2^i \cdot |CS_i|)$ 。远程结点在接受到包围信封后，便更新下界并将局部最小的  $k$  个下界发送给协调者结点。这一过程发送的数据量最多为  $O(|C_i|)$ 。总的来说，迭代过程中产生的通信开销为。假设迭代结束后，所剩  $N'$  ( $N' > k$ ) 个结点分布在  $m'$  个远程结点中。此时最多需要将原始轨迹发送到  $m'$  个结点中。其所对应的开销为  $O(d \cdot n \cdot m')$ 。结合这两个阶段，DTW-FLB 的总通信开销为  $O(\sum_{i=0}^{\log_2 n} (d \cdot 2^i \cdot |CS_i| + |C_i|) + d \cdot n \cdot m')$ 。

通过以上对 DTW-FLB 的分析，我们可以发现该算法对迭代剪枝的效果十分敏感。即剪枝结束后所剩候选的个数即包含这些候选的节点的个数，会严重影响到计算时间和通信开销。由于，DTW-FLB 会在每一轮迭代中都会通过对部分候选轨迹计算真实的 DTW 距离用以更新全局过滤的阈值。该方法能保证该过滤阈值会不断逼近真实的第  $k$  小距离，因此最终会取得较好的剪枝效果。

## 5.4 实验分析

### 5.4.1 实验设置

本章工作在实验中使用北京出租车数据集，该数据集已被广泛应用于轨迹数据分析。该数据集采集了北京市三万多辆出租车在 2013 年 10 月份到 12 月份三个

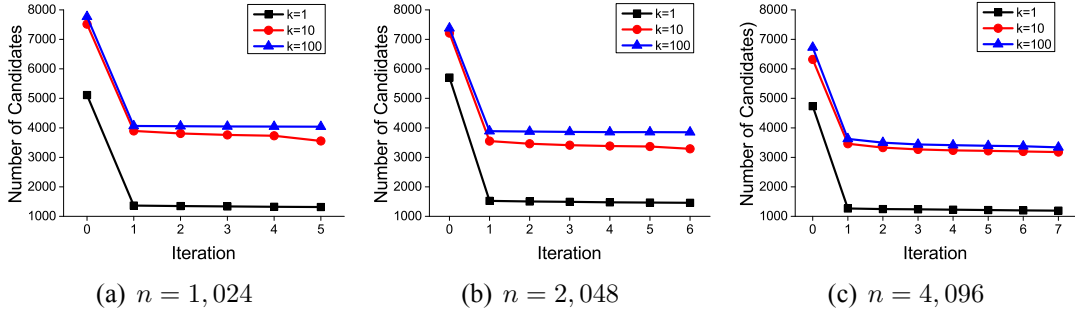


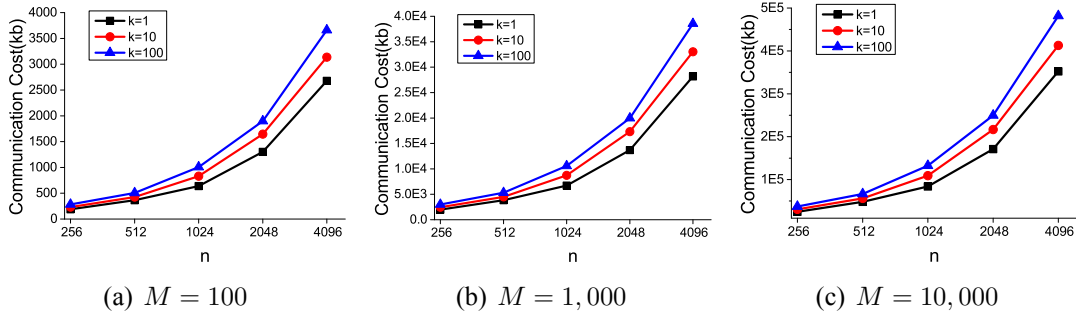
图 5.3: DTW-FLB 剪枝效果

月内的行驶 GPS 轨迹。每个 GPS 轨迹点包含的数据维度较多，我们仅选取了位置(经度和纬度)、时间、速度和角度这五个维度的值，并进行了归一化预处理。我们从该数据集中截取了两个子数据集： $T-Small$  和  $T-Big$  以分别验证算法的有效性和可扩展性。 $T-Small$  截取了 10 月 1 至 7 号间上午 8 点到 10 点的轨迹数据，并从中选出最长的 1 万条轨迹进行分析。 $T-Big$  截取了 11 月 1 号至 12 月 31 号间每天上午 8 至 10 点和下午 5 至 7 点两个时间段的 1 百万条轨迹数据。为满足实验需求，这两个数据集的每条轨迹长度均超过 4,096。

在本章的实验中，DTW-FLB 算法使用 JAVA 实现。需要注意的是，在我们介绍中  $\delta$  的取值为整数来表示约束的范围。但为适应不同长度轨迹的需求，我们在实验中使用轨迹长度的百分比来表示约束范围。实验运行在一个包含 12 个结点的 Spark 集群上。每个结点包含 8 核英特尔 E5335 2.0 GHz 中央处理器和 16GB 的内存。

#### 5.4.2 算法有效性

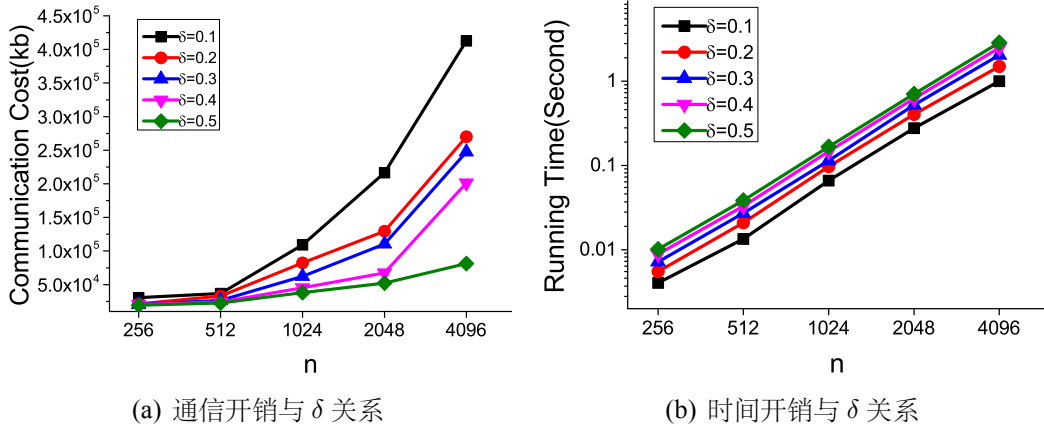
首先，我们通过汇报迭代过程中每轮结束后所剩的候选的个数以展示 DTW-FLB 的剪枝效果。在该组实验中，我们设置轨迹长度  $n$  从 256 变化到 4,096， $k$  的值从 1 变化到 100。图5.3展示了随着  $k$  和  $n$  变化后的剪枝的效果。从该图中我们可以看出在最开始的两轮迭代中，候选的个数急剧下降。此外，两轮迭代后所剩候选数较少。这说明，即使使用粗粒度的概要数据索引和剪枝策略的混合使用能起到较好的剪枝效果。在接下来的迭代中，我们可以发现候选个数降低比较平缓。由

图 5.4: DTW-FLB 通信开销与  $n$ ,  $M$  和  $k$  之间的关系

此可见, 本文提出的由粗到细使用多粒度概要数据进行剪枝的思想具有较高的优越性, 即它能够使用较少的数据剪枝掉大多数候选, 从而达到了降低通信开销的目的。此外, 可以发现随着  $k$  值的降低, 我们能取得更好的剪枝效果。究其原因,  $k$  值较小时, 所选取的全局过滤阈值随之降低即更接近真实的第  $k$  小真实距离值。

接着, 我们研究了  $n$ ,  $k$  和  $M$  对 DTW-FLB 算法通信开销的影响。在该组实验中, 我们将  $M$  值从 100 变化到 10,000, 将  $k$  从 1 变化到 100, 并将  $n$  从 256 变化到 4,096。实验结果如图 5.4 所示, 我们可以看到随着  $n$  和  $M$  的指数增加, 通信开销随之指数增长。这实际反映了通信开销与  $n$  和  $M$  之间存在着线性关系。究其原因, 随着  $M$  的增加, 候选轨迹会存在于更多的远程结点中。所以协调者结点需要将概要数据发送到更多的节点中, 这导致了通信开销的增加。此外, 当  $n$  增加时, 长轨迹需要传递更多的数据才能跟短轨迹达到相同的剪枝效果。最后, 随着  $k$  值的增加, 通信开销也会随之增长。这一结论跟图 5.3 所得到的结果一致, 反映了随着  $k$  的增加, 剪枝阈值越松, 进而每轮迭代中所剩候选越多。

然后, 我们从时间和通信开销两个角度研究了 DTW 距离计算中  $\delta$  的值对 DTW-FLB 算法的影响。在该组实验中, 我们将  $k$  和  $m$  的值分别固定为 10 和 10,000。实验结果如图 5.5 所示。我们可以发现随着  $\delta$  的增加通信开销逐步减小。其原因是  $\delta$  值的增加会导致我们的包围信封更加紧凑, 从而更利于剪枝。同时, 随着  $\delta$  的增加时间开销逐步增加。这是因为  $\delta$  值越大导致最后一步计算 DTW 真实距离时的搜索空间越大。此外, 结合前面对 DTW-FLB 的时间复杂度分析部分, 我们可以发现


 图 5.5: DTW-FLB 性能与  $\delta$  间的关系

这一实验结果跟我们的理论分析是一致的。

#### 5.4.3 算法可扩展性

本小节我们将从时间和通信两个角度研究了 DTW-FLB 算法的可扩展性。首先, 考虑了轨迹数量  $N$  (即数据量大小) 和轨迹长度  $n$  对可扩展性的影响。在该组实验中, 我们将  $k$  设置为 1,  $m$  设置为 10,000。我们将轨迹长度从 256 到 4096 进行指数级变化, 同时将轨迹数量从 10 万到 1 百万进行线性变换。图 5.6 分别对这两个角度的结果进行了介绍。其中图 5.6(a) 介绍了时间性能受  $n$  和  $N$  的影响。我们可以, 看到对于任意长度的轨迹数据集, DTW-FLB 算法的运行时间都随着轨迹数据量的增加而线性增加。此外, 随着轨迹长度的指数增加, 算法的运行时间也指数增加, 即运行时间与轨迹长度呈一定的线性关系。这一结果反映了由于 DTW-FLB 的剪枝效果较好, 导致迭代结束后, 所剩候选数较少。即我们对 DTW-FLB 算法时间复杂度分析部分的  $N'$  较少。结合以上两点, 我们可以看出 DTW-FLB 算法运行时间随着数据集的大小而线性变换, 因此运行效率具有较好的可扩展性。

接着, 我们在图 5.6(b) 中介绍了通信性能受  $n$  和  $N$  的影响。从图中可以看出, 通信量随着轨迹的数据的线性增加而呈指数增加。这是由于 DTW-FLB 具有较好的剪枝效果, 导致迭代结束后, 不仅所剩候选较少, 而且所剩下的候选结点也较少。因此, 迭代结束后的通信较少。而根据我们前面对 DTW-FLB 的通信开销分析



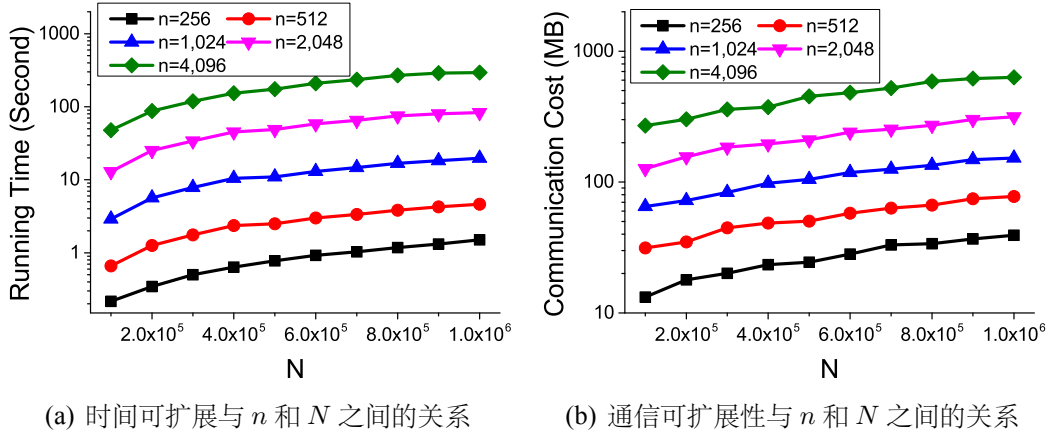


图 5.6: The scalability of DTW-FLB

可知通信开销主要来自迭代式计算部分和迭代后将轨迹发送给所有的候选结点部分。由于后一部分开销较少，故主要开销集中在第一部分，即迭代式通信部分。根据图5.3分析可知，前两次迭代已经过滤掉很过结点，后面的候选变话较小。此时，第一部分的开销主要跟包围盒大小相关。而包围盒大小是呈指数变换（当前大小是上一次的2倍）。所以，我们的实验结果跟我们的理论分析一致。此外，从图中可以看出通信开销随着轨迹长度的指数变化而指数增加，即通信开销与轨迹长度呈线性关系。这反应了对于长轨迹需要更多的概要数据才能达到与短轨迹相同的剪枝效果。

## 5.5 本章小结

本章节介绍了利用 FLB 框架在嵌入动态时间卷曲距离下的具体实现算法 DTW-FLB。本章节，首先提出了针对动态时间卷曲距离的包围信封以作为概要数据。接着，提出了基于包围信封的动态时间距离的下界。该下界能够随着包围信封粒度的增加而逐渐变紧，这使得我们利用将该距离嵌入到 FLB 框架称为可能。为此，我们提出了 DTW-FLB 算法以实现基于动态时间卷曲距离的查询。在我们的查询算法中，我们引入了索引、边计算下界边剪枝等查询优化措施，以提高效率。通过在真实数据集上进行的实验，表明 DTW-FLB 算法剪枝效果较好、算法效率高且具有较好的可扩展性。

## 5.6 附件

**引理 5.2.1.** 给定  $q$ 、 $c$  以及  $q$  的包围矩形  $\{u, l\}$ ，我们有  $SED\_LB(\{u, l\}, c) \leq SED(q, c)$ 。

**证明.** 对于点的任意维度，如第  $j$  维，我们有如下结论

$$\begin{cases} (u^j - c^j)^2 \leq (q^j - c^j)^2 & \text{if } u^j \leq c^j, \\ 0 \leq (q^j - c^j)^2 & \text{if } l^j \leq c^j \leq u^j, \\ (c^j - l^j)^2 \leq (q^j - c^j)^2 & \text{if } c^j \leq l^j. \end{cases} \quad (5.8)$$

由于  $SED\_LB(\{u, l\}, c)$  可看做上式左边部分在各个维度的累加， $SED(q, c)$  是上式右半部分的累加。故累加所有维度的值后，我们可以得到  $SED\_LB(\{u, l\}, c) \leq SED(q, c)$ 。  $\square$

**定理 5.2.2.** 给定轨迹  $C$  和待查询轨迹  $Q$  满足动态时间卷曲约束的包围信封  $\{\mathcal{U}, \mathcal{L}\}$ ，我们有如下结论:  $LB\_Keogh(\{\mathcal{U}, \mathcal{L}\}, C) \leq DTW(Q, C)$ 。

**证明.** 根据  $LB\_Keogh(\{\mathcal{U}, \mathcal{L}\}, C)$  和  $DTW(Q, C)$  的定义，我们的目标即是证明如下不等式成立。

$$\sum_{i=0}^{n-1} SED\_LB(\{u_i, l_i\}, c_i) \leq \sum_{k=1}^K w_k \quad (5.9)$$

此外我们还有  $n \leq K$ ，我们接下来的策略是证明上述不等式左边的每个值都能在右边找到一个比它大或相等的元素：

$$\begin{aligned} SED\_LB(\{u_i, l_i\}, c_i) &\leq w_k \\ \Leftrightarrow SED\_LB(\{u_i, l_i\}, c_i) &\leq \|c_i^j - q_x^j\|^2 \\ \Leftrightarrow \sum_{j=0}^{d-1} SED\_LB(\{u_i^j, c_i^j\}, c_i^j) &\leq \sum_{j=0}^{d-1} (c_i^j - q_x^j)^2 \end{aligned} \quad (5.10)$$

其中  $x$  与  $i$  满足如下不等式约束  $i - \delta \leq x \leq i + \delta$ 。同时，根据  $u_i$  和  $l_i$  的定义，我

们有  $\forall j \ell_i^j \leq q_x^j \leq u_i^j$ . 因此, 我们进一步的得到如下不等式:

$$\begin{aligned} & \begin{cases} (u_i^j - c_i^j)^2 \leq (c_i^j - q_x^j)^2 & \text{if } u_i^j \leq c_i^j, \\ 0 \leq (c_i^j - q_x^j)^2 & \text{if } \ell_i^j \leq c_i^j \leq u_i^j, \\ (c_i^j - \ell_i^j)^2 \leq (c_i^j - q_x^j)^2 & \text{if } c_i^j \leq \ell_i^j. \end{cases} \\ \Leftrightarrow & SED\_LB(\{u_i^j, c_i^j\}, c_i^j) \leq (c_i^j - q_x^j)^2 \\ \Leftrightarrow & \sum_{j=0}^{d-1} SED\_LB(\{u_i^j, c_i^j\}, c_i^j) \leq \sum_{j=0}^{d-1} (c_i^j - q_x^j)^2 \end{aligned}$$

因此, 不等式 5.10 成立。原问题得证。  $\square$

**定理 5.2.3.** 给定轨迹  $\mathcal{Q}$  的第  $l$  层包围信封  $\{\hat{\mathcal{U}}_l, \hat{\mathcal{L}}_l\}$  和候选轨迹  $\mathcal{C}$ , 我们得到如下结论:  $LB\_HPAA(\{\hat{\mathcal{U}}_l, \hat{\mathcal{L}}_l\}, \bar{\mathcal{C}}_l) \leq DTW(\mathcal{Q}, \mathcal{C})$ 。

**证明.** 从定理 5.2.2 可知  $LB\_keogh$  是原始 DTW 距离的下界。因此, 若我们能得到  $LB\_HPAA(\{\hat{\mathcal{U}}_l, \hat{\mathcal{L}}_l\}, \bar{\mathcal{C}}_l) \leq LB\_Keogh(\{\mathcal{U}, \mathcal{L}\}, \mathcal{C})$ , 则原问题得证。根据  $LB\_HPAA$  的定义, 我们可知第  $l$  层包围信封的每个元素对应着  $LB\_Keogh$  的  $s$  ( $s = 2^{L-l}$ ) 个元素。如果  $LB\_HPAA$  的每个元素满足如下不等式, 则原问题得证。

$$s \cdot SED\_LB(\{\hat{u}_{l,i}, \hat{l}_{l,i}\}, \bar{c}_{l,i}) \leq \sum_{p=i \cdot s}^{(i+1) \cdot s - 1} SED\_LB(\{u_p, l_p\}, c_p)$$

根据不等式 5.11, 由于  $\forall j \hat{l}_{l,i}^j \leq l_p^j \leq u_p^j \leq \hat{u}_{l,i}^j$ , 我们有  $SED\_LB(\{\hat{u}_{l,i}, \hat{l}_{l,i}\}, c_p) \leq SED\_LB(\{u_p, l_p\}, c_p)$ 。

$$\begin{cases} (\hat{u}_{l,i}^j - c_p^j)^2 \leq (u_p^j - c_p^j)^2 & \text{if } u_p^j \leq \hat{u}_{l,i}^j \leq c_p^j, \\ 0 \leq (u_p^j - c_p^j)^2 & \text{if } u_p^j \leq c_p^j \leq \hat{u}_{l,i}^j, \\ 0 \leq 0 & \text{if } l_p^j \leq c_p^j \leq u_p^j, \\ 0 \leq (l_p^j - c_p^j)^2 & \text{if } \hat{l}_{l,i}^j \leq c_p^j \leq l_p^j, \\ (\hat{l}_{l,i}^j - c_p^j)^2 \leq (l_p^j - c_p^j)^2 & \text{if } c_p^j \leq \hat{l}_{l,i}^j \leq l_p^j. \end{cases} \quad (5.11)$$

所以,我们的问题变为求证  $s \cdot SED\_LB(\{\hat{\mathbf{u}}_{l,i}, \hat{\mathbf{l}}_{l,i}\}, \bar{\mathbf{c}}_{l,i}) \leq \sum_{p=i \cdot s}^{(i+1) \cdot s-1} SED\_LB(\{\hat{\mathbf{u}}_{l,i}, \hat{\mathbf{l}}_{l,i}\}, \mathbf{c}_p)$  成立。为简化证明,我们只证明第一个轨迹片段(即  $i = 0$  时)成立。此外我们使用  $\hat{\mathbf{u}}, \hat{\mathbf{l}}$  和  $\bar{\mathbf{c}}$  来分别表示  $\hat{\mathbf{u}}_{l,0}, \hat{\mathbf{l}}_{l,0}$  和  $\bar{\mathbf{c}}_{l,0}$ 。对于其他片段,证明方法相同。此时,我们的目标就是证明如下不等式:

$$s \cdot SED\_LB(\{\hat{\mathbf{u}}, \hat{\mathbf{l}}\}, \bar{\mathbf{c}}) \leq \sum_{i=0}^{s-1} SED\_LB(\{\hat{\mathbf{u}}, \hat{\mathbf{l}}\}, \mathbf{c}_i) \quad (5.12)$$

根据  $SED\_LB$  定义,我们展开不等式5.12。此时,若对于任一维度  $j$  有如下不等式成立则不等式5.12也成立。

$$s \cdot SED\_LB(\{\hat{\mathbf{u}}^j, \hat{\mathbf{l}}^j\}, \bar{\mathbf{c}}^j) \leq \sum_{i=0}^{s-1} SED\_LB(\{\hat{\mathbf{u}}^j, \hat{\mathbf{l}}^j\}, \mathbf{c}_i^j) \quad (5.13)$$

对于以上不等式的右半部分,  $\mathbf{c}_i^j$  和  $\{\hat{\mathbf{u}}^j, \hat{\mathbf{l}}^j\}$  之间存在 3 种大小关系。不失一般性,我们假设这 3 种关系均出现在这  $s$  对元组中。为此,我们首先将所有  $\mathbf{c}_i$  进行重新排序,使得排序后的结果满足:(i) 从  $\mathbf{c}_0$  到  $\mathbf{c}_{p-1}$  中的元素,满足  $\mathbf{c}_i^j \leq \hat{\mathbf{u}}^j$ ; (ii) 从  $\mathbf{c}_p$  到  $\mathbf{c}_{q-1}$  中的元素,满足  $\hat{\mathbf{l}}^j \leq \mathbf{c}_i^j \leq \hat{\mathbf{u}}^j$ ; (iii) 从  $\mathbf{c}_q$  到  $\mathbf{c}_{s-1}$  中的元素满足  $\mathbf{c}_i^j \leq \hat{\mathbf{l}}^j$ , 其中  $0 \leq p \leq q \leq s$ 。那么根据  $\bar{\mathbf{c}}$  的定义,我们得到下述结论:

$$\sum_{i=0}^{p-1} (\mathbf{c}_i^j - \bar{\mathbf{c}}^j) = \sum_{i=p}^{s-1} (\bar{\mathbf{c}}^j - \mathbf{c}_i^j) = \sum_{i=p}^{q-1} (\bar{\mathbf{c}}^j - \mathbf{c}_i^j) + \sum_{i=q}^{s-1} (\bar{\mathbf{c}}^j - \mathbf{c}_i^j) \quad (5.14)$$

然后,我们考虑不等式5.13的左半部分。其根据  $\hat{\mathbf{u}}^j$  和  $\bar{\mathbf{c}}^j$  之间的大小关系可分为 3 种情况。(i)  $\hat{\mathbf{u}}^j \leq \bar{\mathbf{c}}^j$  的情况,此时左半部分的值为  $s(\bar{\mathbf{c}}^j - \hat{\mathbf{u}}^j)^2$ 。此时,对于右半部分我们有

$$\begin{aligned} \sum_{i=0}^{s-1} (\{\hat{\mathbf{u}}^j - \hat{\mathbf{l}}^j\}, \mathbf{c}_i^j)^2 &= \sum_{i=0}^{p-1} (\mathbf{c}_i^j - \hat{\mathbf{u}}^j)^2 + \sum_{i=q}^{s-1} (\hat{\mathbf{l}}^j - \mathbf{c}_i^j)^2 \\ &\geq \frac{1}{s-q+p} \left( \sum_{i=0}^{p-1} (\mathbf{c}_i^j - \hat{\mathbf{u}}^j) + \sum_{i=q}^{s-1} (\hat{\mathbf{l}}^j - \mathbf{c}_i^j) \right)^2 \quad (AM - GM \square \square \square) \\ &\geq \frac{1}{s-q+p} \left( \sum_{i=p}^{q-1} (\bar{\mathbf{c}}^j - \mathbf{c}_i^j) + p(\bar{\mathbf{c}}^j - \hat{\mathbf{u}}^j) + \sum_{i=q}^{s-1} (\hat{\mathbf{l}}^j - 2\mathbf{c}_i^j + \bar{\mathbf{c}}^j) \right)^2 \end{aligned}$$

$$\begin{aligned}
 &\geq \frac{1}{s-q+p} \left( \sum_{i=p}^{q-1} (\bar{\mathbf{c}}^j - \hat{\mathbf{u}}^j) + p(\bar{\mathbf{c}}^j - \hat{\mathbf{u}}^j) + \sum_{i=q}^{s-1} (\bar{\mathbf{c}}^j - \hat{\mathbf{l}}^j) \right)^2 \\
 &= \frac{s^2}{s-q+p} (\bar{\mathbf{c}}^j - \hat{\mathbf{u}}^j)^2 \geq s(\bar{\mathbf{c}}^j - \hat{\mathbf{u}}^j)^2
 \end{aligned} \tag{5.15}$$

因此，第一种情况下原问题得证。(ii)  $\bar{\mathbf{c}}^j \leq \hat{\mathbf{l}}^j$  的情况，此时证明过程与第一种情况类似。(iii)  $\hat{\mathbf{l}}^j \leq \bar{\mathbf{c}}^j \leq \hat{\mathbf{u}}^j$  的情况，此时不等式5.13成立，因其左半部分值为0，而有半部分是在大于0。结合以上3种情况，原问题得证。  $\square$

**定理 5.2.4.**  $LB\_HPAA$  下界能随着查询轨迹包围信封层次的增加而逐渐变紧。也就是说： $LB\_HPAA(\{\hat{\mathcal{U}}_l, \hat{\mathcal{L}}_l\}, \bar{\mathcal{C}}_l) \leq LB\_HPAA(\{\hat{\mathcal{U}}_{l+1}, \hat{\mathcal{L}}_{l+1}\}, \bar{\mathcal{C}}_{l+1})$ 。

**证明.** 根据多粒度包围信封的计算法方式我们可知，当包围信封从第  $l$  层转到第  $l+1$  层时，其每个元素分为两个元素用于分别表示左右两边的最值。所以，我们只要证明如下不等式即可：

$$\begin{aligned}
 2 \cdot SED\_LB(\{\hat{\mathbf{u}}_{l,i}, \hat{\mathbf{l}}_{l,i}\}, \bar{\mathbf{c}}_{l,i}) &\leq SED\_LB(\{\hat{\mathbf{u}}_{l+1,2i}, \hat{\mathbf{l}}_{l+1,2i}\}, \bar{\mathbf{c}}_{l+1,2i}) \\
 &\quad + SED\_LB(\{\hat{\mathbf{u}}_{l+1,2i+1}, \hat{\mathbf{l}}_{l+1,2i+1}\}, \bar{\mathbf{c}}_{l+1,2i+1})
 \end{aligned}$$

为简化问题，我们分别使用  $\hat{\mathbf{u}}$ ,  $\hat{\mathbf{u}}_L$  和  $\hat{\mathbf{u}}_R$  来分别表示  $\hat{\mathbf{u}}_{l,i}$ ,  $\hat{\mathbf{u}}_{l+1,2i}$  和  $\hat{\mathbf{u}}_{l+1,2i+1}$ 。并且使用  $\bar{\mathbf{c}}$ ,  $\bar{\mathbf{c}}_L$  和  $\bar{\mathbf{c}}_R$  来分别代表  $\bar{\mathbf{c}}_{l,i}$ ,  $\bar{\mathbf{c}}_{l+1,2i}$  和  $\bar{\mathbf{c}}_{l+1,2i+1}$ 。那么不等式可重新表示为如下形式：

$$\begin{aligned}
 2 \cdot SED\_LB(\{\hat{\mathbf{u}}, \hat{\mathbf{l}}\}, \bar{\mathbf{c}}) &\leq SED\_LB(\{\hat{\mathbf{u}}_L, \hat{\mathbf{l}}_L\}, \bar{\mathbf{c}}_L) \\
 &\quad + SED\_LB(\{\hat{\mathbf{u}}_R, \hat{\mathbf{l}}_R\}, \bar{\mathbf{c}}_R)
 \end{aligned} \tag{5.16}$$

此时如果我们能证明对任一维度  $j$ ，都能满足如下不等式，则不等式5.16成立。

$$\begin{aligned}
 2 \cdot SED\_LB(\{\hat{\mathbf{u}}^j, \hat{\mathbf{l}}^j\}, \bar{\mathbf{c}}^j) &\leq SED\_LB(\{\hat{\mathbf{u}}_L^j, \hat{\mathbf{l}}_L^j\}, \bar{\mathbf{c}}_L^j) \\
 &\quad + SED\_LB(\{\hat{\mathbf{u}}_R^j, \hat{\mathbf{l}}_R^j\}, \bar{\mathbf{c}}_R^j)
 \end{aligned} \tag{5.17}$$

对于不等式5.17左半部分， $\bar{\mathbf{c}}^j$  和  $\{\hat{\mathbf{u}}^j, \hat{\mathbf{l}}^j\}$  之间存在以下三种关系：

(i)  $\bar{\mathbf{c}}^j \geq \hat{\mathbf{u}}^j$ ，此时左半部分等价于  $2 \cdot (\bar{\mathbf{c}}^j - \hat{\mathbf{u}}^j)^2$ 。而右半部分，我们首先假设  $\bar{\mathbf{c}}_L^j \leq \bar{\mathbf{c}}^j \leq \bar{\mathbf{c}}_R^j$ 。此时我们得到如下结论  $SED\_LB(\{\hat{\mathbf{u}}_L^j, \hat{\mathbf{l}}_L^j\}, \bar{\mathbf{c}}_L^j) \geq (\bar{\mathbf{c}}_L^j - \hat{\mathbf{u}}^j)^2$ 。因此，不等式5.17 左半部分等于  $2 \cdot SED(\bar{\mathbf{c}}, \hat{\mathbf{u}})$ 。对于其右半部分，我们有

$$\begin{aligned}
 & SED\_LB(\{\hat{\mathbf{u}}_L^j, \hat{\mathbf{l}}_L^j\}, \bar{\mathbf{c}}_L^j) + SED\_LB(\{\hat{\mathbf{u}}_R^j, \hat{\mathbf{l}}_R^j\}, \bar{\mathbf{c}}_R^j) \\
 &= SED\_LB(\{\hat{\mathbf{u}}_L^j, \hat{\mathbf{l}}_L^j\}, \bar{\mathbf{c}}_L^j) + (\bar{\mathbf{c}}_R^j - \hat{\mathbf{u}}_R^j)^2 \quad (\hat{\mathbf{u}}_R^j \leq \hat{\mathbf{u}}^j \leq \bar{\mathbf{c}}^j \leq \bar{\mathbf{c}}_R^j) \\
 &\geq (\bar{\mathbf{c}}_L^j - \hat{\mathbf{u}}^j)^2 + (\bar{\mathbf{c}}_R^j - \hat{\mathbf{u}}_R^j)^2 \\
 &\geq (\bar{\mathbf{c}}_L^j - \hat{\mathbf{u}}^j)^2 + (\bar{\mathbf{c}}_R^j - \hat{\mathbf{u}}^j)^2 \quad (\bar{\mathbf{u}}_R^j \leq \hat{\mathbf{u}}^j \leq \bar{\mathbf{c}}_R^j) \\
 &\geq \frac{1}{2} \cdot (\bar{\mathbf{c}}_L^j - \hat{\mathbf{u}}^j + \bar{\mathbf{c}}_R^j - \hat{\mathbf{u}}^j)^2 \quad (AM - GM \square \square \square) \\
 &= \frac{1}{2} \cdot (2\bar{\mathbf{c}}^j - 2\hat{\mathbf{u}}^j)^2 \quad (\bar{\mathbf{c}}_L^j + \bar{\mathbf{c}}_R^j = 2\bar{\mathbf{c}}^j) \\
 &= 2 \cdot (\bar{\mathbf{c}}^j - \hat{\mathbf{u}}^j)^2
 \end{aligned}$$

对于  $\bar{\mathbf{c}}_R^j \leq \bar{\mathbf{c}}^j \leq \bar{\mathbf{c}}_L^j$  的情况，我们通过相同的方法得到同样的结论。因此，此种情况下不等式 5.16 成立。(ii)  $\bar{\mathbf{c}}^j \leq \hat{\mathbf{l}}^j$ ，可通过跟第一种相同的方法证明不等式 5.17 成立。(iii)  $\hat{\mathbf{l}}^j \leq \bar{\mathbf{c}}^j \leq \hat{\mathbf{u}}^j$ ，此时不等式5.17 的左半部分为 0，其右半部分始终大于 0。因此，结论也成立。综合以上几种情况，不等式5.16 始终成立。原问题得证。  $\square$



## 参考文献

- [1] DENG Z, HU Y, ZHU M, et al. A scalable and fast OPTICS for clustering trajectory big data[J]. Cluster Computing, 2015, 18(2): 549–562.
- [2] COSTA G, MANCO G, MASCIARI E. Dealing with trajectory streams by clustering and mathematical transforms[J]. Journal of Intelligent Information Systems, 2014, 42(1): 155–177.
- [3] YU Y, WANG Q, WANG X, et al. Online clustering for trajectory data stream of moving objects[J]. Computer Science and Information Systems, 2013, 10(3): 1293–1317.
- [4] MAO J, SONG Q, JIN C, et al. TScluWin: Trajectory Stream Clustering over Sliding Window[C] // Proceedings of the 21st DASFAA, Part II. 2016: 133–148.
- [5] NEHME R V, RUNDENSTEINER E A. SCUBA: Scalable Cluster-Based Algorithm for Evaluating Continuous Spatio-temporal Queries on Moving Objects[C] // Proceedings of the 10th EDBT. 2006: 1001–1019.
- [6] SACHARIDIS D, PATROUMPAS K, TERROVITIS M, et al. On-line discovery of hot motion paths[C] // Proceedings of the 11th EDBT. 2008: 392–403.
- [7] ZHENG K, ZHENG Y, YUAN N J, et al. On discovery of gathering patterns from trajectories[C] // Proceedings of the 29th ICDE. 2013: 242–253.
- [8] TANG L A, ZHENG Y, YUAN J, et al. On Discovery of Traveling Companions from Streaming Trajectories[C] // Proceedings of the 28th ICDE. 2012: 186–197.
- [9] LI X, CEIKUTE V, JENSEN C S, et al. Effective Online Group Discovery in Trajectory Databases[J]. TKDE, 25(12): 2752–2766.
- [10] DUAN X, JIN C, WANG X, et al. Real-Time Personalized Taxi-Sharing[C] // Proceedings of the 21st DASFAA Part II. 2016: 451–465.
- [11] ZHANG Z, WANG Y, MAO J, et al. DT-KST: Distributed Top-k Similarity Query on Big Trajectory Streams[C] // Proceedings of the 22nd DASFAA, Part I. 2017: 199–214.
- [12] HSU C-C, KUNG P-H, YEH M-Y, et al. Bandwidth-efficient distributed k-nearest-neighbor search with dynamic time warping[C] // Proceedings of the 2015 ICBD. 2015: 551–560.



- [13] ZEINALIPOUR-YAZTI D, LAOUDIAS C, COSTA C, et al. Crowdsourced trace similarity with smartphones[J]. TKDE, 2013, 25(6): 1240–1253.
- [14] COSTA C, LAOUDIAS C, ZEINALIPOUR-YAZTI D, et al. SmartTrace: Finding similar trajectories in smartphone networks without disclosing the traces[C] // Proceedings of the 27th ICDE. 2011: 1288–1291.
- [15] CHEN L, ÖZSU M T, ORIA V. Robust and fast similarity search for moving object trajectories[C] // Proceedings of the 2005 ACM SIGMOD. 2005: 491–502.
- [16] ZHU H, LUO J, YIN H, et al. Mining Trajectory Corridors Using Frechet Distance and Meshing Grids[C] // Proceedings of the 14th PAKDD Part I. 2010: 228–237.
- [17] GUO N, MA M, XIONG W, et al. An Efficient Query Algorithm for Trajectory Similarity Based on Frechet Distance Threshold[J]. International Journal of Geo-Information, 2017, 6(11): 326.
- [18] LIN B, SU J. One Way Distance: For Shape Based Similarity Search of Moving Object Trajectories[J]. GeoInformatica, 2008, 12(2): 117–142.
- [19] LIU H, SCHNEIDER M. Similarity measurement of moving object trajectories[C] // Proceedings of the 3rd ACM SIGSPATIAL Workshop on GeoStreaming. 2012: 19–22.
- [20] ZHAO X, XU W. A New Measurement Method to Calculate Similarity of Moving Object Spatio-Temporal Trajectories by Compact Representation[J]. International Journal of Computational Intelligence Systems, 2011, 4(6): 1140–1147.
- [21] ZHENG B, YUAN N J, ZHENG K, et al. Approximate keyword search in semantic trajectory database[C] // Proceedings of the 31st ICDE. 2015: 975–986.
- [22] MAGDY N, SAKR M A, MOSTAFA T, et al. Review on trajectory similarity measures[C] // Proceedings of the 7th ICICIS. 2016: 613–619.
- [23] TOOHEY K, DUCKHAM M. Trajectory similarity measures[J]. Proceedings of the 2015 SIGSPATIAL Special, 2015, 7(1): 43–50.
- [24] VERNICA R, CAREY M J, LI C. Efficient parallel set-similarity joins using MapReduce[C] // Proceedings of the 16th ACM SIGMOD. 2010: 495–506.
- [25] KIM Y, SHIM K. Parallel top-k similarity join algorithms using MapReduce[C] // Proceedings of the IEEE 28th ICDE. 2012: 510–521.

- [26] ZEINALIPOUR-YAZTI D, LIN S, GUNOPULOS D. Distributed spatio-temporal similarity search[C] // Proceedings of the 2006 CIKM. 2006 : 14–23.



## 附录 主要缩写符号对照表

|      |   |
|------|---|
| ALS  | 交替最小二乘法 (Alternating Least Square)                      |
| AUC  | ROC 曲线下面积 (Area under the ROC Curve)                    |
| BPMF | 贝叶斯概率矩阵分解 (Bayesian Probabilistic Matrix Factorization) |
| BPR  | 贝叶斯个性化排序 (Bayesian Personalized Ranking)                |
| ERR  | 期望排序倒数 (Expected Reciprocal Rank)                       |
| LDA  | 潜在狄利克雷分布 (Latent Dirichlet Allocation)                  |
| MF   | 矩阵分解 (Matrix Factorization)                             |
| MRR  | 平均排序倒数 (Mean Reciprocal Rank)                           |
| NDCG | 归一化的贴现累计收益 (Normalized Discounted Cumulative Gain)      |
| PLSA | 概率潜语义分析 (Probabilistic Latent Semantic Analysis)        |
| PMF  | 概率矩阵分解 (Probabilistic Matrix Factorization)             |
| PITF | 成对相互张量分解 (Pairwise Interaction Tensor Factorization)    |
| RMSE | 均方根误差 (Root Mean Square Error)                          |
| ROC  | 受试者工作特征曲线 (Receiver Operating Characteristic Curve)     |
| SGD  | 随机梯度下降 (Stochastic Gradient Descent)                    |
| SVD  | 奇异值分解 (Singular Value Decomposition)                    |
| TF   | 张量分解 (Tensor Factorization)                             |
| WMF  | 加权矩阵分解 (Weighted Matrix Factorization)                  |



## 致 谢

四年本科、五年硕博，回首这九年的华东师大求学时光，历经本科好友、硕士同门的陆续毕业，此刻我也将博士毕业，感慨良多。在此期间，有幸得到老师和亲友们的指导与帮助，在此谨对他们表示衷心的感谢！

首先，我要郑重地感谢我的本硕博导师王晓玲教授。每个人生命中都会遇到贵人，我想她就是我的贵人。她为我们创造了良好的学习环境氛围，在学习方法上的细心指导，在生活上的关怀支持。我依稀记得刚王老师在我刚读研时为我指明了研究的方向，读博迷茫期时对我的耐心开导和鼓励，以及即将毕业之际传授于我未来需要的宝贵的工作经验。在以后的人生道路上，我都会一直铭记她的教诲。

其次，我十分感谢复旦大学的沙朝锋副教授。沙老师深厚的数学功底和精彩地模型介绍，让我看到了数据挖掘技术的魅力。感谢华东师范大学金澈清教授在关于 LBS 相关研究中给予的大量帮助和指导，金老师对待问题的严谨性让我印象深刻。然后，我还要感谢中国人民大学的赵鑫老师。赵老师花费许多精力和时间与我讨论研究问题和研究方法，修改学术论文，让我学习了大量的学术论文写作技巧。也特别感谢华东师范大学周傲英副校长、钱卫宁教授、宫学庆教授、何晓丰研究员、张蓉教授、高明副教授和周敏奇副教授在日常学习与研究生课程中给予的多方面帮助和指导。

另外，感谢所有读研期间陪伴我的同学和朋友，你们是我的美好记忆。感谢已经毕业的林煜明博士、王立博士、徐辰博士、王朝勇、胡颢继等师兄，以及马建松、江俊文、段小艺等师弟师妹，感谢你们在我研究生前期给予的学习上的帮助和生活上的快乐。感谢张凯、彭宏伟、靳远远，你们为我的论文提出了宝贵的修改意见。特别感谢陪我度过研究生时光的朱涛和张新洲，谢谢你们几年来对我的关心和包容。感谢一起毕业的纪文迪博士、房俊华博士、孔超博士、张浪博士和孟丹博士，与你们一起毕业是我的荣幸。也祝尚在奋斗的朱涛博士、庞艳霞博士、毛嘉莉博士、章志刚博士、周欢博士顺利完成学业，早日毕业。感谢在 109 实验室一起学习的梁磊、赵大鹏、刘志、宋光旋、李财政、夏得伦、张颖、吕晓强、刘小捷、屈稳稳、贺韵宇、周纯依、刘文焱等师弟师妹们。此外，我还想感谢本科室友邱星星、吴超凡、李博，以及硕士同学张磊、李勇峰、董绍婵和顾玲，谢谢你们当年的一起

玩耍以及对我找工作时的帮助和关心。

最后，着重感谢我的父母，对我攻读博士学位的大力支持。感谢未婚妻的一直陪伴，你的理解和奉献使我能够无忧地学习，感谢这份许多年来历久弥坚的爱恋。

王科强

二零一七年五月

## 攻读博士学位期间发表的学术论文、科研情况以及奖项

### ■ 已公开发表论文

- [1] **Keqiang Wang**, Wayne Xin Zhao, Hongwei Peng, Xiaoling Wang. Bayesian Probabilistic Multi-Topic Matrix Factorization for Rating Prediction. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 3910-3916, 2016.
- [2] **Keqiang Wang**, Hongwei Peng, Yuanyuan jin, Chanfeng Sha, Xiaoling Wang. Local Weighted Matrix Factorization for Top-n Recommendation with Implicit Feedback. In *Data Science and Engineering (DSE)*, volume 1, no. 4, pages 252-264, 2016.
- [3] **Keqiang Wang**, Xiaoyi Duan, Jiansong Ma, Chaofeng Sha, Xiaoling Wang, Aoying Zhou. Local Weighted Matrix Factorization for Implicit Feedback Datasets. In *Proceedings of the 21th Database Systems for Advanced Applications (DASFAA)*, pages 381-395, 2016.
- [4] **Keqiang Wang**, Chaofeng Sha, Xiaoling Wang, Aoying Zhou. Based on Citation Diversity to Explore Influential Papers for Interdisciplinarity. In *Proceedings of the 16th Asia-Pacific Web Conference (APWeb)*, pages 343-354, 2014. (Best Student Paper)
- [5] Chaofeng Sha, **Keqiang Wang**, Xiaoling Wang, Aoying Zhou. Optimizing Top-k Retrieval: Submodularity Analysis and Search Strategies. In *Frontiers of Computer Science (FCS)*, volume 10, no. 3, pages 477-487, 2016.
- [6] 马建松, **王科强**, 宋光旋, 张凯, 王晓玲, 金澈清. 面向 MAX/MIN 优化的 SQL Window 函数处理. *计算机学报*, 39 卷, 10 期, 2149-2160 页, 2016.
- [7] Kai Zhang, **Keqiang Wang**, Xiaoling Wang, Aoying Zhou. Hotel recommendation based on user preference analysis. In *Proceeding of the 31th IEEE International Conference on Data Engineering Workshops (ICDEW)*, pages 134-138, 2015.
- [8] Chaofeng Sha, **Keqiang Wang**, Dell Zhang, Xiaoling Wang, Aoying Zhou. Optimizing Top-k Retrieval: Submodularity Analysis and Search Strategies. In Pro-



ceedings of the 15th International Conference on Web-Age Information Management (WAIM), pages 18-29, 2014.

- [9] Chaofeng Sha, **Keqiang Wang**, Xiaoling Wang, Aoying Zhou. Ensemble Pruning: A Submodular Function Maximization Perspective. In Proceedings of the 19th Database Systems for Advanced Applications (DASFAA), pages 1-15, 2014.
- [10] Chaofeng Sha, **Keqiang Wang**, Kai Zhang, Xiaoling Wang, Aoying Zhou. Diversifying Top-k Service Retrieval. In Proceedings of the 11th IEEE International Conference on Services Computing (SCC), pages 227-234, 2014.
- [11] 张凯, **王科强**, 王晓玲, 金澈清, 周傲英. 基于评论分析的酒店推荐系统. 计算机研究与发展, 增刊 I, 51 卷, 372-376 页, 2014.
- [12] Hongwei Peng, Haojie Hu, **Keqiang Wang**, Xiaoling Wang. Time-Aware and Topic-Based Reviewer Assignment. In Proceedings of the 22th Database Systems for Advanced Applications Workshops (DASFAA Workshops), pages 145-157, 2017.

#### ■ 参加的科研项目

- [1] 海量众包数据管理的关键技术(国家自然科学基金面上项目, 编号: 61472141)
- [2] XML 个性化协作搜索及其在社会网络服务中的应用(国家自然科学基金面上项目, 编号: 61170085)
- [3] 数据密集型计算环境下的数据管理方法与技术(国家自然科学基金面上项目, 编号: 61033007)
- [4] 海量音频数据管理系统, 核高基项目《非结构化数据管理系统》(编号: 2010ZX01042-002-003-004)

#### ■ 获得奖项

- [1] 上海市优秀毕业生, 2017
- [2] 研究生国家奖学金, 2016
- [3] 校优秀学生, 2016
- [4] 智慧研究生奖学金, 2015
- [5] CCF 数据竞赛三等奖, 2014