中国科学院大学

University of Chinese Academy of Sciences

博士学位论文

基于深度学习的轨迹数据挖掘关键技术研究

作者姓名：姚 迪

指导教师：毕经平 研究员

中国科学院计算技术研究所

学位类别：工学博士

学科专业：计算机系统结构

培养单位：中国科学院计算技术研究所

2019 年 06 月

Research on Deep Learning-based Techniques for Trajectory Data Mining

A dissertation submitted to

The University of Chinese Academy of Sciences

in partial fulfillment of the requirement

for the degree of

Doctor of Philosophy

in

Computer Systems Organization

by

Di Yao

Supervisor: Professor Jingping Bi

Institute of Computing Technology,

Chinese Academy of Sciences

June 2019

中国科学院大学

研究生学位论文原创性声明

本人郑重声明：所呈交的学位论文是本人在导师的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，本论文不包含任何其他个人或集体已经发表或撰写过的研究成果。对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或致谢。

作者签名：

日　　期：

中国科学院大学

学位论文授权使用声明

本人完全了解并同意遵守中国科学院有关保存和使用学位论文的规定，即中国科学院有权保留送交学位论文的副本，允许该论文被查阅，可以按照学术研究公开原则和保护知识产权的原则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、汇编本学位论文。

涉密及延迟公开的学位论文在解密或延迟期后适用本声明。

作者签名：　　导师签名：

日　　期：　　日　　期：

摘 要

随着移动互联网技术的高速发展，带有位置采集功能的设备被广泛应用，产生了海量的轨迹数据。轨迹数据挖掘可以发现如人类、船舶、汽车等移动目标的移动规律和行为特征，对目标识别、情报分析、城市规划等任务有着重要的意义。轨迹数据挖掘已逐渐成为数据挖掘中的热点研究领域。与其他类型数据不同，轨迹数据具有异频采样性、时空序列性及目标差异性等特点，这些特点使得轨迹数据挖掘面临许多挑战：一是由于数据采样频率不一致，导致轨迹数据中记录时间的分布不均匀。在轨迹分析中，采样稀疏的轨迹面临目标位置不确定的挑战；二是由于轨迹的时空序列属性，使得计算轨迹相似度的复杂度至少为平方级。随着数据规模逐渐增大，轨迹相似度计算面临复杂度高的挑战；三是由于不同目标的轨迹在移动行为上存在差异，造成轨迹数据库中存在行为异常的轨迹。这些异常轨迹的标签需要人工标记，导致轨迹异常检测面临数据标签稀少的挑战。近些年来，深度学习技术的兴起为处理上述挑战提供了新的有力手段。

本文利用深度学习技术在隐含特征提取以及复杂函数近似方面的优势，重点研究了稀疏轨迹补全、轨迹相似度计算和轨迹异常检测三个问题，着力解决轨迹数据挖掘面临的数据不确定、相似度计算复杂度高和异常轨迹标签少等挑战。本文的创新性主要体现在：

（1）提出上下文信息感知的稀疏轨迹补全方法

针对稀疏轨迹中静默时间内目标位置缺失和不确定的问题，本文提出了上下文信息感知的稀疏轨迹补全方法 TrajCom。该方法从目标轨迹的上下文信息入手，自动地从历史轨迹库筛选出有用信息，补全目标轨迹中缺失的位置。TrajCom 分为相关轨迹筛选、深度轨迹编码及缺失位置估计三个步骤。首先，基于目标轨迹时空约束和目标偏好的信息，利用神经协同过滤 NCF（Neural Colleberitive Filtering）筛选出与目标轨迹相关的轨迹来解决数据稀疏问题。然后，基于循环神经网络 RNN（Recurrent Neural Network）设计轨迹编码模型并提出时间间隔感知的循环神经网络单元 tGRU（time-aware GRU）用于捕捉动态变化的时间间隔信息。最后，TrajCom 利用注意力机制估计目标轨迹中的缺失位置。我们使用四个真实轨迹数据集来评估 TrajCom 的效果。其结果表明，对比现有的轨迹补全方法，TrajCom 在准确率上提升了 25%。另外，TrajCom 可以通过追溯相关历史轨迹来解释位置缺失的原因。

（2）提出基于深度度量学习的轨迹相似度计算方法

针对轨迹相似度计算复杂度高的问题，本文提出了基于深度度量学习的轨迹相似度计算方法 NeuTraj。该方法是一种轨迹相似度计算的近似方法，利用种子轨迹两两之间相似度的精确值作为监督信息，NeuTraj 训练基于深度神经网络的轨迹编码器，将轨迹相似度计算复杂度降低到线性。NeuTraj 整体基于深度度量学习框架，可以分为数据准备、轨迹编码和度量学习三个步骤。首先，对于给定轨迹数据库，该方法从数据库中采样轨迹作为种子轨迹集合，计算种子轨迹两两之间的相似度，生成种子轨迹相似度矩阵；然后，基于循环神经网络，NeuTraj 将轨迹序列编码成固定长度的表示向量。在此步骤中，本文提出了空间注意力记忆机制，用于解决传统 RNN 不能对序列之间关系进行建模的问题。最后，在度量学习中，为了更快、更准确地学习 NeuTraj 参数，本文提出了加权排序损失，该损失可以将模型优化的重点聚焦在更有区分力的轨迹对上。在实验部分，我们使用两个真实轨迹数据集评估了 NeuTraj 在四种相似性度量方法上的准确率和效率。结果表明 NeuTraj 在 Top-10 相似轨迹检索上的准确率达到 80%，该结果优于已有近似方法。而且，与已有近似算法相比 NeuTraj 的计算时间缩短了 3 倍以上。

（3）提出基于移动行为特征的半监督轨迹异常检测方法

针对轨迹数据异常标签稀少的问题，本文提出了基于移动行为特征的半监督异常检测方法 Traj2Vec。该方法是一个半监督学习的框架，分为无监督学习和有监督学习两个步骤。在无监督学习中，Traj2Vec 利用轨迹自身移动行为的特点，学习轨迹的表示向量。首先，Traj2Vec 利用滑动窗口和移动行为特征抽取算法，解决了数据采样率不均的问题。然后，基于序列自编码模型，将轨迹移动行为特征序列映射为表示向量。在有监督学习中，Traj2Vec 利用少量数据标签，基于分歧学习技术训练轨迹异常分类模型。首先，以无监督学习中的轨迹编码器为基础，利用异常标签，学习初始分类器并用该分类器对无标签轨迹进行分类；然后，根据分类器生成的伪标签，使用协同训练的方法优化初始分类器。重复上述过程直至模型收敛。训练过程中，Traj2Vec 的有监督部分和无监督部分共享轨迹编码器的参数，共同优化异常检测模型。本文通过仿真数据和真实数据来评估 Traj2Vec 性能。其结果表明，与已有轨迹异常检测方法相比，Traj2Vec 在异常检测准确率略有提升（3%）的前提下，将异常轨迹的召回率提高了 10%。

关键词：轨迹数据挖掘，稀疏轨迹补全，轨迹相似度计算，轨迹异常检测

Abstract

With the rapid development of mobile Internet, devices with location acquisition function have been widely used and result in a large amount of trajectory data. Trajectory data mining can be used for investigating the moving pattern and behaviors of objects, which is of great significance to object recognition, intelligence analysis, urban planning *etc.* Consequently, trajectory data mining has become an increasingly popular research field. Different from other data, trajectory data has the characteristics of varying sampling frequency, spatio-temporal correlation and different moving behavior which makes trajectory data mining challenging. Because of the inconsistent sampling frequency, locations of moving objects are sparsely sampled in some trajectories. Location uncertainty is a serious problem in these sparse trajectories. Despite that, the spatio-temporal character of trajectory leads to the quadratic complexity of trajectory similarity measures. With the increase of data, the computation cost would become a bottleneck in trajectory similarity calculation. In addition, since the moving behaviors of object are various, trajectories which have anomalous behaviors are widely distributed in trajectory database. Anomaly labeling requires marking manually, which causes scarce labels in trajectory anomaly detection. In recent years, the development of deep learning has provided a new powerful platform to address these challenges.

Taking the advantages of deep learning technology in implicit feature extraction and complex function approximation, this dissertation focuses on three key problems: sparse trajectory completion, trajectory similarity computation and trajectory anomaly detection, to solve the challenges of data uncertainty, high similarity computation complexity and scarce data labels in trajectory data mining. The novelty of this dissertation is mainly embodied in:

(1) A context-aware neural filtering-encoding method is proposed for completing sparse trajectory data.

Aiming at the problem of data missing and location uncertainty in sparse trajectory, we propose an effective method for completing sparse and irregular trajectories. Our method, named TrajCom, is a two-step filtering-and-encoding method. In the first step, it leverages rich context information to filter a set of reference

trajectories based on spatio-temporal constraints and collaborative filtering. Such reference trajectories, which correlate strongly with the target trajectory, address data sparsity by providing complementary information and emphasizing local patterns around the query point. In the second step, TrajCom learns time-aware encodings for the target trajectory and its reference trajectories. The encoding procedure is underpinned by a novel time-aware recurrent unit, which handles time gap irregularity and makes the completion accurately. Experiments on four real-world datasets show that TrajCom significantly outperforms competitive baselines for the trajectory completion task, with up to 25% relative improvements over the closest competitor. Moreover, TrajCom provides intuitive explanations that rationalize its completions.

(2) A seed-guided neural metric learning method is proposed to reduce the complexity of computing trajectory similarity.

In order to reduce the high computation cost of existing trajectory similarity measures, we propose a Neural metric learning based Trajectory similarity computation method, short for NeuTraj. NeuTraj is an approximate method which is generic to accommodate any existing trajectory measure and fast to compute the similarity of a given trajectory pair in linear time. Furthermore, NeuTraj is elastic to collaborate with all spatial- based trajectory indexing methods to reduce the search space. NeuTraj samples a number of seed trajectories from the given database, and then uses their pair-wise similarities as guidance to approximate the similarity function with a neural metric learning framework. NeuTraj features two novel modules to achieve accurate approximation of the similarity function: (1) a spatial attention memory module that augments existing recurrent neural networks for trajectory encoding; and (2) a distance-weighted ranking loss that effectively transcribes information from the seed-based guidance. With these two modules, NeuTraj can yield high accuracies and fast convergence rates even if the training data is small. Our experiments on two real-life datasets show that NeuTraj achieves over 80% accuracy on Fre´chet, Hausdorff, ERP and DTW measures, which outperforms state- of-the-art baselines consistently and significantly. It obtains 50x- 1000x speedup over brute-force methods and 3x-500x speedup over existing approximate algorithms, while yielding more accurate approximations of the similarity functions.

(3) A semi-supervised trajectory anomaly detection method is proposed for dealing with scarce labeled data.

Aiming at the problem of scarcity of labeled trajectories, we propose a semi-supervised anomaly detection method based on mobile behavior characteristics. Our method, namely Traj2Vec, is a semi-supervised learning framework, which embeds trajectories to vectors for anomaly detection. Traj2Vec is consists of unsupervised learning procedure and supervised learning procedure. In unsupervised learning, Traj2Vec utilizes the moving behavior features to learn the embedding vector of trajectory. Firstly, Traj2Vec uses a sliding window with a moving behavior feature extraction algorithm to solve the uneven sampling rate. Then, based on the sequence autoencoder, the moving behavior sequences of trajectory are mapped to representation vectors. In supervised learning, Traj2Vec utilizes a small number of labeled trajectories to train anomaly classifier in a disagreement-based learning approach. Firstly, based on the trajectory encoder, an initial classifier is learnt by using the labeled data and used to classify the unlabeled trajectories. Then, the initial classifier is further optimized according to the pseudo labels generated by the classifier. Finally, the classifier repeats the process until converges. In the experiment, the performance of Traj2Vec is evaluated by synthetic data and real trajectory data. The results show that, compared with the existing trajectory anomaly detection methods, Traj2Vec improves the recall of anomaly trajectory by more than 10%, and the overall accuracy of anomaly detection by 3%.

Keywords: Trajectory Data Mining, Sparse Trajectory Completion, Trajectory Similarity Computation, Trajectory Anomaly Detection

目 录

图目录

表目录

第 1 章　绪论

1.1　研究背景与意义

轨迹数据可以看作是移动目标随时间变化在空间中留下的痕迹。随着移动互联网、位置服务等技术的高速发展和 GPS 设备的普及，在运营出行服务和监控遥感目标的过程中，会产生大量的轨迹数据。例如，大部分城市中的出租车都装配有定位设备，这些设备会以相对固定的频率向数据中心上报位置，因此城市数据中心汇集了大量的出租车轨迹数据；装配自动识别系统(AIS)的船舶，每隔 3-15 分钟就会上报一条带有位置、航行状态和目的地信息的记录，整个系统每天收集到的船舶位置数据高达数亿条；飞机飞行过程中，需要实时向地面指挥中心发送包含位置、高度和速度等信息的记录用于飞行监管；遥感卫星也可通过电子侦查等手段获得所监控区域内移动目标装配雷达的位置和参数信息。多源丰富的轨迹数据，从不同的粒度、层面和视角记录了目标的活动信息，为轨迹数据的应用提供了有利条件。为了了解移动目标行为，迫切需要系统地研究轨迹数据挖掘方法。因此，轨迹数据挖掘正在成为数据挖掘中的一个热点研究领域。轨迹数据挖掘涉及许多学科包括计算机科学、生物学、社会学、地理学和气候学等，其研究成果可以为用户提供更人性化、更广阔的服务。

与其他类型的数据不同，轨迹数据具有以下三方面的特点：

- 异频采样性：由于轨迹采集不同、目标的行为差异等原因，造成了轨迹中位置记录的采样间隔差异显著。这种差异性增加了轨迹分析的难度。
- 时空序列性：轨迹数据由包含有位置、时间信息的记录序列组成。每一条记录都描述了目标位置动态变化，因此时空序列性是轨迹数据最基本的特点。
- 目标差异性：轨迹数据是目标连续运动的离散化表示。受到目标偏好、移动习惯和环境变化的影响，造成不同目标轨迹移动行为上的差异显著。

针对轨迹数据特点，已有研究人员从各个方面开展轨迹数据挖掘研究。Zheng（2015）、高强等（2017）根据现有研究成果对轨迹数据挖掘的问题和应用做了系统的总结，提出轨迹数据挖掘的研究框架。如图 1.1 所示，本文借鉴上述工作

中轨迹数据挖掘的分层方法，将轨迹数据挖掘分为三层，即轨迹预处理层、轨迹检索层和轨迹应用层。

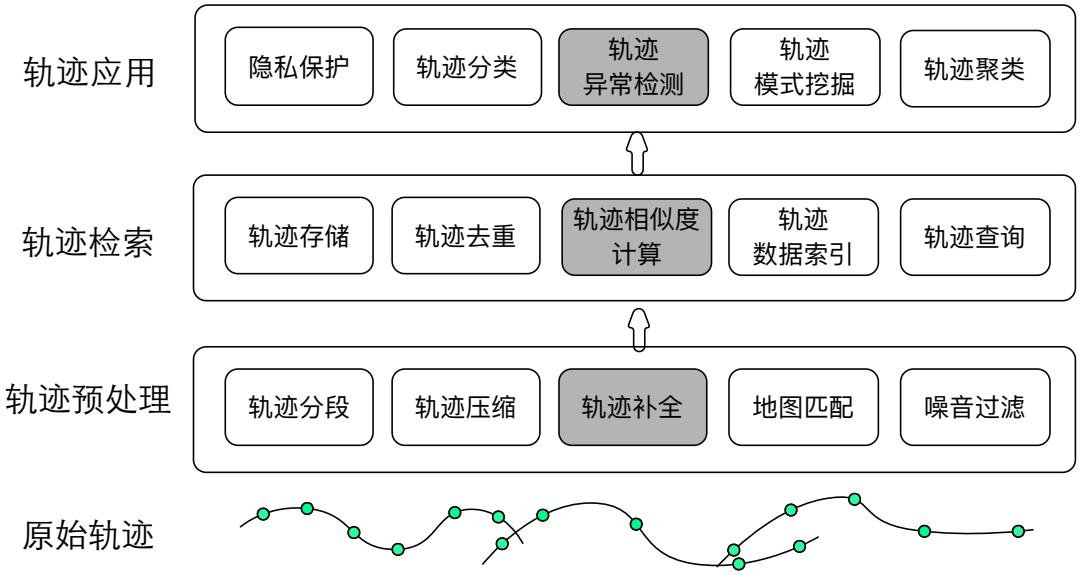

图 1. 1 轨迹数据挖掘研究框架

Figure 1.1 Archtricture of trajectory data mining

- 轨迹预处理层。轨迹预处理层的任务是将原始位置记录转换为可用的轨迹，并存入轨迹数据库。由于位置采样频率差异、信号传输渠道不畅以及信息对抗等原因，造成轨迹数据“空洞”，即轨迹中存在没有位置记录的静默时间。因此轨迹数据的缺失和不确定是该层面临的重要挑战之一。
- 轨迹数据检索层。轨迹检索层为上层轨迹应用提供基础的轨迹检索服务。轨迹检索中的核心算法是轨迹相似度计算。而现有轨迹相似性度量方法的计算复杂度至少为平方级。随着历史轨迹数量的增加，相似度计算的复杂度将成为轨迹检索的瓶颈。
- 轨迹数据应用层。轨迹数据应用层面向不同的应用场景，研究有针对性的轨迹挖掘算法。轨迹异常检测是轨迹数据挖掘的重要应用之一，其目标在于从数据库中发现移动目标的异常轨迹。现实应用中，异常轨迹的数据标签需要依靠专家经验人工标记。这使轨迹异常检测面临数据标签稀少的挑战。

针对上述问题，目前，对于不同类型的轨迹数据国内外已有许多研究人员从

多个角度提出解决方案。

为了解决轨迹缺失和不确定问题，已有研究工作通过轨迹补全来推断目标在数据缺失时间段的位置。该类工作可分为基于单体目标轨迹的方法（Serrano 等.，2017；Li 等.，2016；Long，2016；Li 等.，2015；Chandler 等.，2015；Su 等.，2015）和基于集体移动模式的方法（Yao 等.，2017；Feng 等.，2018；Liu 等.，2016）。基于单体轨迹的补全方法利用单个目标的移动轨迹，训练单体移动模型并利用该模型补全轨迹。Chandler 等（2015）利用数据缺失前后的轨迹数据，通过线性插值的方法对轨迹的缺失部分进行平滑达到补全缺失轨迹的目的。Long（2016）提出了运动学插值的方法补全轨迹，该方法利用速度与加速度估计目标的缺失位置。Serrano 等（2017）将运动学插值的方法应用到移动机器人控制领域用于跟踪机器人的轨迹。另外，Li 等（2015），Li 等（2016）和 Su 等（2015）通过引入城市路网将缺失轨迹映射到道路上并利用道路位置补全轨迹。基于集体移动模式的方法利用历史轨迹数据建模目标的行为习惯，训练一个普适的目标移动模型，并用该模型补全缺失轨迹。Silva 等（2015），Monreale 等（2009）和 Yi 等（2016）通过聚类的方法生成历史轨迹的聚类簇。在补全缺失轨迹时首先将轨迹映射到轨迹簇中，然后根据簇内的轨迹补全当前轨迹。Yao 等（2017），Feng 等（2018）和 Liu 等（2016）利用矩阵分解和序列模型训练位置推荐或预测模型，此类方法也可以应用于缺失轨迹补全。

为了加速轨迹相似度计算，已有工作可分为基于轨迹索引的方法和基于近似算法的方法。基于轨迹索引的方法（Chen 等.，2005；Yazti 等.，2006；Gong 等.，2015；Gowanlock 和 Casanova，2016；Shang 等.，2017）的思路是基于轨迹相似性度量，如 Hausdorff、DTW、Fre´chet 等构造轨迹索引。在轨迹相似性检索中，通过减小涉及计算的轨迹数目来降低检索响应时间。此类方法大多采用 K-D 树或 R 树的索引结构，层次化地组织轨迹数据。检索轨迹时，上述索引可以通过树剪枝的方法快速找到与检索轨迹最相似的轨迹。基于近似算法的方法（Colton 和 Indyk，1999；Backurs 和 Sidiropoulos，2016；Agarwal 等.，2016；Rakthanmanon 等.，2012）针对某一种特定的相似性度量，设计近似算法，降低其计算复杂度。针对 Hausdorff 度量，Colton 和 Indyk（1999），Backurs 和 Sidiropoulos（2016）

提出了基于表示向量的方法，将 Hausdorff 距离转化为 lp 空间中的向量距离。针对 DTW，Rakthanmanon（2012）提出可以省略计算过程中平方操作的方法加速 DTW 计算。近期，Driemel 和 Silvestri（2017）提出了一种基于局部敏感哈希（LSH）的算法，用于快速计算 Fre´chet 和 Hausdorff 距离。

为了解决轨迹异常检测中数据标签稀少的问题，已有研究工作提出利用无标签轨迹数据，提升轨迹异常检测准确率的方法。按照异常检测技术不同，上述方法分为基于模式的轨迹异常检测方法和基于半监督学习的轨迹异常检测方法。基于模式的异常检测方法（Bao 和 Du，2018；Lei 等.，2016；Guo 等.，2014；Laxhammar 和 Falkman，2014；Laxhammar 和 Falkman，2012）通过轨迹聚类、模式挖掘等技术建模目标移动的正常模式，并通过与正常模式对比发现异常轨迹。Laxhammar 和 Falkman（2012）利用滑动窗口划分子轨迹，提出局部异常因子对同一滑动窗口的轨迹聚类，最后将不属于任何一簇的子轨迹判定为异常轨迹。Guo 等（2014）基于轨迹的形状和速度，首先利用核密度估计技术表示轨迹，并通过最大化聚类簇之间互信息的方式得到轨迹聚类，最后基于香侬熵判断轨迹是否为异常轨迹。Lei 等（2016）通过模式挖掘发现同类型目标在空间、序列和行为上的正常移动模式并基于此检测异常轨迹。Laxhammar 和 Falkman（2014）基于 Hausdorff 距离定义轨迹异常因子并通过该因子检测轨迹流中的异常轨迹。Bao 和 Du（2018）首先通过 DBSCAN 得到轨迹的模式，然后利用点到轨迹的 Haversine 距离判断轨迹是否为异常。基于半监督学习的异常检测方法目前研究成果较少，Sillito 和 Fisher（2008），Arnaud 等（2013）通过人为交互的方法获取监督信息并推断无标签轨迹的标签，最后结合所有轨迹信息检测异常轨迹。

解决轨迹数据缺失不确定、加速轨迹相似度计算、实现数据标签稀少下的轨迹异常检测对轨迹数据挖掘至关重要，但现有工作在稀疏轨迹补全、轨迹相似度计算、轨迹异常检测上仍然存在如下问题：

（1） 已有轨迹补全方法不适用于稀疏轨迹，且无法对静默时间段内目标位置的动态变化进行建模。

现有轨迹挖掘方法对于稀疏轨迹大多直接丢弃或简单插值，其主要原因是目前已有的轨迹补全方法大多针对稠密轨迹如出租车轨迹，而不适用于稀疏轨迹。

补全稀疏轨迹可以降低目标移动的不确定性，增加可用轨迹数量，进一步为目标监控识别等应用提供数据支撑。然而，已有的基于单目标的轨迹补全方法（Serrano 等.，2017；Li 等.，2016；Long，2016；Chandler 等.，2015；Li 等.，2015；Su 等.，2015）通过目标的运动特性，如速度、加速度等，对目标位置动态变化进行建模。由于稀疏轨迹静默时间较长，这类方法在处理稀疏轨迹时会产生很大误差。已有基于多目标的轨迹补全方法（Li 等.，2016；Li 等.，2015； Su 等.，2015）利用目标移动行为推荐或预测未来可能访问的位置，但该类方法仅基于缺失时间段前部分轨迹推断若干个可能访问的位置，且该类方法无法对目标在长段静默时间内位置的动态变化进行建模。

（2） 已有加速轨迹相似度计算的工作没有降低计算复杂度，或仅适用于特定相似性度量。

计算轨迹相似度是一项基础且重要的任务，可以广泛应用于轨迹聚类、模式发现、异常检测、目标识别等领域。轨迹相似度计算方法多种多样，不同方法关注的角度不同但有至少平方级的计算复杂度。另外，在某些应用中，如目标识别，需要综合利用多种相似度达到识别的目的，这就需要一个可以适用于多种相似性度量的加速计算方法。而现有基于轨迹索引的方法（Chen 等.，2005；Yazti 等.，2006；Gong 等.，2015；Gowanlock 和 Casanova，2016；Shang 等.，2017）通过构建轨迹索引减小涉及计算的轨迹数量，该类方法仅能加速 Top-K 轨迹相似性检索并没有降低轨迹相似度计算复杂度。另外，基于近似算法的工作（Colton 和 Indyk，1999；Backurs 和 Sidiropoulos，2016；Agarwal 等.，2016；Rakthanmanon 等.，2012），仅针对特定的相似性度量设计加速算法，对于不同的度量加速效果也不同，且对于某些相似性度量没有可用的近似算法。

（3） 已有轨迹异常检测方法依赖先验知识确定异常阈值，且难以解决异常轨迹稀少的问题。

由于轨迹数据中异常轨迹的标签需要依赖人工手动标记，造成了现有轨迹数据集数据标签稀少且难以获得。在这种情况下，仅利用少量有标签数据无法支撑异常检测模型的训练。因此，如何利用无标签轨迹数据中蕴含的信息来解决异常轨迹稀少的问题对轨迹异常检测显得尤为重要。虽然现有基于轨迹模式的异常检

测方法（Bao 和 Du，2018；Lei 等.，2016；Guo 等.，2014；Laxhammar 和 Falkman，2014； Laxhammar 和 Falkman，2012）可以通过聚类获得无标签轨迹的分布特征后，利用预定义的规则和模式筛选异常轨迹，但此类方法依赖先验知识，需要领域专家定义异常阈值。另外，此类方法完全基于无标签数据，难以利用轨迹数据标签。基于半监督学习的方法（Sillito 和 Fisher，2008；Arnaud 等.，2013）分为无监督学习和有监督学习两部分，其中无监督部分大多基于轨迹聚类，而有监督部分基于人机交互获得轨迹异常标签信息，调整无监督聚类的结果。然而传统聚类算法利用轨迹统计特征生成轨迹聚类，该类特征表示无法与轨迹异常建立直接联系，导致该类算法异常检测准确率较低。深度学习技术能够自动地抽取轨迹中与异常检测相关的特征，可以用于解决上述问题。

针对上述问题，本文围绕轨迹数据挖掘的现实需求，利用深度学习方法在隐含特征提取和复杂函数近似方面的优势，研究并提出可行的解决方案：（1）针对稀疏轨迹难以补全问题，本文提出上下文信息感知的稀疏轨迹补全方法，降低稀疏轨迹中目标的不确定性，为目标监控和识别提供数据支撑；（2）针对轨迹相似度计算复杂度高的问题，本文提出基于深度度量学习的轨迹相似度计算方法，将相似度计算复杂度降低到线性，并适用于加速多种相似性度量；（3）针对轨迹异常标签稀少的问题，本文提出基于移动行为特征的半监督轨迹异常检测方法，利用无标签轨迹辅助有标签轨迹训练模型，提高异常检测准确率。

1.2 论文的贡献

本文围绕轨迹数据挖掘的关键问题展开研究。在国家自然基金 61702470、61472403 等项目的支持下，本文深入分析了轨迹数据挖掘的国内外研究现状。并在此基础上，围绕轨迹补全、轨迹相似度计算和轨迹异常检测中亟待解决的关键问题展开研究，提出了上下文信息感知的稀疏轨迹补全方法、基于深度度量学习的轨迹相似度计算方法和基于移动行为特征的半监督轨迹异常检测方法。论文的研究成果弥补了已有相关成果在稀疏轨迹补全、轨迹检索效率和异常检测准确率上的不足，保证了轨迹挖掘算法的有效运行。

本文的主要研究成果体现在以下几个方面：

（1） 深入调研了轨迹数据挖掘的背景及研究现状，分析了轨迹数据挖掘不同层次面临的挑战

本文对轨迹数据挖掘背景、内容和研究现状进行了深入的分析和探讨。首先简要介绍了轨迹数据的定义、特点及分类，分析了导致轨迹数据存在数据不确定、计算复杂度高、异常标签少的原因；其次，介绍了轨迹数据挖掘的基本框架，并深入调研了已有研究在轨迹补全、轨迹相似度计算和轨迹异常检测上已经取得的成果，分析了这些方法存在的不足。这些方法的不足促成了本文后续章节的研究工作。

（2） 针对稀疏轨迹补全的问题，提出了上下文信息感知的稀疏轨迹补全方法

由于采集模式和设备功耗的限制，现有轨迹数据中存在缺失的时间段。这使得轨迹数据面临目标位置不确定的挑战。稀疏轨迹中缺失的时间段较长，使得现有轨迹补全方法中的目标运动特性假设误差变大，不适用于补全稀疏轨迹。因此，需要进一步深入研究稀疏轨迹补全方法。

针对上述问题，本文提出了上下文信息感知的稀疏轨迹补全方法 TrajCom。该方法从目标轨迹的上下文信息入手，自动地从历史轨迹库筛选出有用信息，补全目标轨迹中缺失的位置。TrajCom 分为相关轨迹筛选、深度轨迹编码及缺失位置估计三个步骤。首先，基于目标轨迹时空约束和目标偏好的信息，利用神经协同过滤 NCF（Neural Colleberitive Filtering）筛选出与目标轨迹相关的轨迹来解决数据稀疏问题。然后，基于 RNN 设计轨迹编码模型并提出时间间隔感知的循环神经网络单元 tGRU（time-aware GRU）用于捕捉动态变化的时间间隔信息。最后，TrajCom 利用注意力机制估计目标轨迹中的缺失位置。我们使用四个真实轨迹数据集来评估 TrajCom 的效果。实验结果表明，对比现有的轨迹补全方法，TrajCom 在准确率上提升了 25%。另外，TrajCom 可以通过追溯相关历史轨迹来解释位置缺失的原因。

（3） 针对轨迹相似度计算复杂度高的问题，提出了基于深度度量学习的轨迹相似度近似计算方法

轨迹相似度计算是各种轨迹数据分析应用的基础。然而，轨迹相似度计算方法多种多样且复杂度为平方级，难以适用于大规模数据集。尽管已有研究人员提

出近似算法可以用于降低相似度计算的复杂度，但它们仅能加速某一种特定的相似度计算方法，无法适用于其他相似性度量。

针对上述问题，本文提出了基于深度度量学习的轨迹相似度近似计算方法NeuTraj。该方法能适用于多种已有的轨迹相似度计算方法，并能将相似度计算的复杂度降低到线性。首先，NeuTraj 从给定的数据库中采样种子轨迹集合，然后计算种子对之间相似度的值，并使用它们作为监督信息，学习模型参数；然后，NeuTraj 通过 RNN 将轨迹编码为固定长度的表示向量。其中轨迹编码模型的循环神经网络单元基于本文提出的空间注意记忆机制。该机制可以增强现有的RNN 使其具有捕捉轨迹空间邻近特性的能力；最后，本文提出了加权排序损失，用于训练 NeuTraj。该损失函数可以有效地将模型学习的重点聚焦在更有区分力的种子轨迹对上。通过以上步骤，NeuTraj 即使在小规模训练数据上，也可以做到高近似程度和快速收敛。基于两个真实的轨迹数据集的实验结果表明，NeuTraj 在 Fre´chet，Hausdorff，ERP 和 DTW 计算方法上 Top-10 相似轨迹检索的准确率能够达到 80％以上，该结果优于现有加速轨迹相似度计算的近似方法。另外，与已有近似算法相比 NeuTraj 的计算时间缩短了 3 倍以上。

（4） 针对轨迹异常检测中数据标签稀少的问题，提出了基于移动行为特征的半监督轨迹异常检测方法

在现有轨迹数据中，异常轨迹标签需要由专家手工标记，这使得异常检测面临数据标签稀少的问题。尽管已有工作利用聚类或模式挖掘的方法，提取无标签轨迹中的信息检测异常轨迹，但此类方法不但无法与有标签数据结合而且需要先验知识设定异常阈值，并没有解决数据标签稀少的问题。

针对上述问题，本文提出了基于移动行为特征的半监督异常检测方法Traj2Vec。该方法是一个半监督学习的框架，分为无监督学习和有监督学习两个步骤。在无监督学习中，Traj2Vec 利用轨迹自身移动行为的特点，学习轨迹的表示向量。首先，Traj2Vec 利用滑动窗口和移动行为特征抽取算法，解决了数据采样率不均的问题。然后，基于序列自编码模型，将轨迹移动行为特征序列映射为表示向量。在有监督学习中，Traj2Vec 利用少量数据标签，基于分歧学习技术训练轨迹异常分类模型。首先，以无监督学习中的轨迹编码器为基础，利用异常标

签，学习一个初始的分类器并用该分类器对无标签轨迹进行分类；然后，根据分类器生成的伪标签，进一步优化初始分类器。最后，在有标签数据上再次优化该分类器。重复上述过程直至模型收敛。训练过程中，Traj2Vec 的有监督部分和无监督部分共享轨迹编码器的参数，共同优化异常检测模型。本文通过仿真数据和真实数据来评估 Traj2Vec 性能。其结果表明，与已有轨迹异常检测方法相比，Traj2Vec 在异常检测准确率略有提高（3%）的前提下，将异常轨迹的召回率提高了 10%。

1.3　论文的组织

第 2 章对轨迹数据挖掘的相关背景、研究问题和研究现状进行了综合阐述，包括：简要介绍轨迹数据的定义、特点、分类、轨迹数据挖掘的基本框架和挑战，这为后续研究提供指导和方向；深入分析了已有研究在轨迹补全、轨迹相似性检索和轨迹异常检测上已经取得的成果，并分析了这些方法存在的不足。

第 3 章针对稀疏轨迹难以补全的问题，本文提出上下文信息感知的轨迹补全方法 TrajCom。首先，分析了不同轨迹数据集移动目标时空约束性的特点。然后，详细介绍了 TrajCom 中相关轨迹筛选、轨迹编码、缺失位置估计三大模块。最后，通过真实数据集上的实验证明了轨迹补全算法的有效性。

第 4 章针对轨迹相似度计算复杂度高的问题，本文提出基于深度度量学习的轨迹相似度计算方法 NeuTraj。首先，分析了造成轨迹相似度计算复杂度高的原因；然后，详细介绍了 NeuTraj 中用于建模轨迹空间邻近特性的 SAM（Spatial Attention Memory）模块和用于度量学习距离加权排序误差；最后，通过效率和准确率实验证明了 NeuTraj 算法的有效性。

第 5 章针对带标签异常轨迹数据稀少的问题，本文提出基于移动行为特征的半监督轨迹异常检测方法 Traj2Vec。首先，详细介绍了基于移动行为特征的无监督轨迹表示学习模型架构，详细说明了每一层的操作步骤；然后，阐述了基于轨迹表示的半监督异常检测方法；最后，通过在仿真数据集和真实数据集上的实验证明上述方法的有效性。

第 6 章总结了本文的研究工作，并讨论未来的研究方向。

第 2 章　轨迹数据挖掘相关研究综述

本章首先介绍了轨迹数据的定义和分类；然后简单概述了轨迹数据挖掘的研究框架及问题；最后详细介绍了轨迹补全、轨迹相似度计算及轨迹异常检测的已有相关研究工作，并分析了它们存在的不足。

2.1　轨迹数据概述

本节首先介绍了轨迹数据的定义，然后详细阐述了轨迹的分类方式，并总结了各类轨迹数据的产生方式、主要采集及应用场景、数据特点及开源数据集。

2.1.1　轨迹数据定义

时空环境下，通过对一个或多个移动目标的运动过程进行采样可以获得包括采样点位置和采样时间的记录，这些记录的序列构成了轨迹。轨迹数据通常可表示为目标位置记录的序列，如 $T = [r_1, r_2, \dots, r_k]$。序列中的每一条记录$r_i$都包含时间和空间两个维度的属性，即$r_i = (l_i\ ,\ t_i)$。

2.1.2　轨迹数据分类

轨迹数据有来源丰富，采集模式多种多样，如智能手机中的定位模块获取到的用户移动轨迹数据、车载 GPS 设备采集到的车辆位置信息、装有 RFID 标签的目标产生的移动路线信息等。目前，研究人员关注的轨迹数据主要包括人类活动产生的数据、动物活动产生的数据、交通工具产生的数据和自然现象产生的数据。

表 2.1 公开轨迹数据集及分类

Table 2.1　The classification of public trajectory datasets

数据种类	产生方式	主要场景	数据特点	典型数据集
人类活动	被动收集：人类佩戴位置及其他传感器设备在人类活动中持续采集的数据	健康监控、运动监控、导航、	数据密集；误差受位置采集设备精度影响大；	Geolife
	主动发布：人类主动公开发布的，表现人类活动位置及状态的数据	位置社交网络、银行卡记录、RFID	数据密度低，特定用户数据稀疏；伴随其他语义信息	Twitter、Flickr

（续表）

交通工具	安装在交通工具上的采集设备为了监控其运行状态采集得到的数据	车联网、船舶监控、飞机航线监控	数据采集频率高；某些时段数据缺失；数据格式规整；	AIS 、T-Drive
动物活动	生物研究人员为了研究动物活动规律，为动物安装传感器，这些传感器回传得到的数据	鸟类活动、草原动物活动	受功率限制，数据稀疏且质量不高；数据随机性强	候鸟迁徙、斑马活动
自然现象	气象学研究人员需要监控自然现象的活动，通过卫星遥感等方式可以获得某些自然现象（台风、洋流）的产生变化数据	台风预报、空气污染监控、	采集成本高；数据稀疏；数据应用实效性强	台风、空气质量等数据

表 2.1 总结了不同数据种类、产生方式、主要场景下的数据特点和典型数据集（姚迪 等. 2018）。可以看出，不同类型轨迹数据的定位精度、采样频率、采集模式各不相同。

2.2 轨迹数据挖掘国内外研究现状

本节首先简要介绍了轨迹数据挖掘的研究框架及框架每部分的主要研究工作。然后针对本文重点研究的轨迹补全、轨迹相似度计算和轨迹异常检测问题，详细介绍了其相关工作以及现有方法存在的缺陷。

2.2.1 轨迹数据挖掘

轨迹数据挖掘是指从位置轨迹序列中提取或挖掘出用户感兴趣内容的过程。与其他类型的数据不同，轨迹数据具有异频采样性、时空序列性和目标差异性的特点。根据以上数据特点，现有轨迹数据挖掘的研究框架可以分为三层。研究框架如图 1.1 所示，

第一层为数据预处理层，该层主要对原始位置记录进行预处理，目的是将原始位置记录转化为可以直接利用的轨迹数据。该层主要研究问题包括轨迹压缩、轨迹分段、地图匹配和轨迹补全等。下面将分别介绍轨迹压缩、轨迹分段和地图匹配的相关工作。另外，轨迹补全为本文研究内容之一，该部分相关工作将在小

节 2.2.2 中详细介绍。

轨迹压缩是指通过降采样、路径匹配等方式，在尽可能减少信息损失的前提下缩短轨迹，降低轨迹的存储和通信开销。根据应用场景不同轨迹压缩方法分为离线压缩和在线压缩两类。离线压缩是指在轨迹生成完成之后再进行压缩。该类方法与计算机图形学中的线段化简问题相似，可以通过 Douglas-Peucker 算法（B,1986）解决。在线压缩是指在轨迹未生成完成就开始压缩过程。该类方法通常基于滑动窗口(Keogh 等.，2001)或开放窗口(Meratnia 和 de，2004)机制将窗口内的轨迹看作离线轨迹进行压缩。另外近期有研究工作通过轨迹语义压缩轨迹。该类方法（Richter 等.，2012；Chen 等.，2009）仅记录轨迹的语义信息而非轨迹位置记录。

轨迹分段是指将长轨迹划分成信息独立的子轨迹段。常用轨迹分段方法包括基于时间间隔的分段方法、基于轨迹形状的分段方法和基于轨迹语义的分段方法。基于时间间隔的方法利用相邻两条记录的时间间隔分段轨迹，将时间间隔大于某个阈值的记录作为分段端点。基于轨迹形状的方法（Yuan 等.，2013；Lee 等.，2008）提取轨迹中目标行为发生改变的点如停泊点、转弯点、加速点等作为分段端点。基于语义的方法（Zheng 等.，2008）首先对轨迹进行语义划分，然后将语义信息发生改变的记录点作为轨迹段的端点。

地图匹配把轨迹数据与地图中的地图信息结合，目标是将位置记录映射到路网中。现有地图匹配算法分为基于附加信息的方法和基于增量匹配的方法。基于附加信息的方法分别从轨迹的几何特征（S，2002）、拓扑特征（Yin 和 Wolfson，2004；Yi 等.，1998）、概率角度（Pink 和 Hummel，2008；Quddus 等.，2006）出发设计路网匹配算法。基于增量匹配的方法（Civilis 等.，2005）利用贪心策略顺序将轨迹记录中的位置匹配到路网中。

第二层为轨迹检索层，该层主要将轨迹数据存入数据库并实现面向不同应用的轨迹检索。对该层的研究围绕轨迹检索展开，包括轨迹时空相似性检索和轨迹空间-关键字(spatial-keyword)检索。研究问题包括轨迹索引、轨迹相似度计算等。下面将分别简要介绍轨迹时空相似性检索和轨迹空间-关键字(spatial-keyword)检索的相关工作。另外，轨迹相似度计算为本文研究内容之一，该部分相关工作将

在小节 2.2.3 中详细介绍。

轨迹时空相似性检索方面，已有研究工作大多基于 R 树(R-tree)（Jensen，1999）构建轨迹索引，给定一条轨迹或一个位置，检索与其相似的轨迹。STR 树(Spatio-Temporal R-tree)（Jensen 和 Theodoridis，2000）是三维 R 树的扩展，专门用于轨迹数据检索。与 R 树不同，STR 树在插入节点是不仅考虑空间邻近性，还考虑轨迹完整性，即把代表同一轨迹中的轨迹片段节点插入到一起。具体来说，STR 树提出了一个算法寻找当前轨迹片段的前驱节点。如果这个节点有空间，就把片段插入；否则就分裂前驱节点。另一个可用于轨迹检索的索引结构是 TB(Trajectory Bundle tree)树（Li 等.，2007），其思想是根据 R 树建立轨迹片段的索引，然后利用链表，将同一轨迹对应的叶节点按顺序连接起来。Xie 等（2017）提出了一个基于 R 树构建分布式索引的方法，实现分布式的轨迹相似性检索。Wang 等（2018）提出了一个轨迹数据的查询引擎 Torch，提出一整套索引结构支持轨迹的 Boolean 查询和相似性查询。

与轨迹相似性检索的索引不同，轨迹空间-关键词检索的索引被用于解决空间-关键词(spatial-keyword)检索，即给定位置坐标和一系列关键词，返回与给定距离相近且语义相似的轨迹。Spaccapietra 等（2008）首次提出面向轨迹数据的空间-关键词检索问题，并提出基于 R 树的索引结构，构建位置到轨迹数据的索引。Rakthanmanon 等（2012）和 Yan（2013）分别从计算效率和语义相似角度改进 Spaccapietra 等（2008）提出的模型，使之支持更大规模的数据集。Liu 等（2017）利用主题模型，将检索输入的关键词和描述轨迹语义的文本信息都映射到高维主题空间中的一个点，并基于四叉树(Quadtree)和 B+树分别构建空间和语义的索引；最终设计 LSH 结构的混合检索算法加速检索过程。

第三层为轨迹应用层，该层面向不同的轨迹应用任务提出不同的挖掘算法。该层主要研究问题包括轨迹隐私保护、轨迹分类、轨迹模式挖掘和轨迹异常检测。下面分别介绍轨迹隐私保护、轨迹分类和轨迹模式挖掘的相关工作。轨迹异常检测是本文的研究内容之一，其相关工作将在小节 2.2.4 中详细介绍。

轨迹隐私保护的目的是在使用用户轨迹时防止用户隐私的泄露。面向在线和离线两类的应用场景，所用到的轨迹隐私保护技术也不相同。在线应用场景中，

用户通过上报当前位置获得应用服务，如导航应用中，获取与当前位置 1 公里以内的路况信息。针对这种应用场景，Xue 等（2013），Chow 等（2011）和 Hoh 等（2010）通过加密或混淆用户发布位置的方法，在保证正常应用的前提下防止用户的隐私信息泄露。离线应用场景中，通过累积大量与同一用户相关的历史轨迹可能暴露用户的隐私，如推断用户家、工作地点等。针对这种应用场景的隐私保护方法包括基于聚类的方法（Abul 等.，2008）、基于泛化的方法（Nergiz 等.，2009）、基于抑制的方法（Terrovitis 和 Mamoulis，2008）和基于网格的方法（falvi 等.，2007）。

轨迹分类的目标是发现轨迹与轨迹间的不同之处，如不同的运动状态、交通模式、用户活动等。面向运动状态分类，Krumm 和 Horvitz（2004）提出基于隐马尔科夫模型将 802.11 信号轨迹分类为静止和运动两类。Sohn 等（2006）将 GSM 轨迹信号按用户运动模式分为静止、行走和驾驶三类。Zhu 等（2012）利用路网特征、停泊点特征等将出租车轨迹分类为载客、非载客和停泊三类。面向交通模式分类，Zheng（2008a,b）利用运动特征如速度变化、方向变化等，基于决策树将人类移动轨迹分为开车、骑行、公交和走路四类。用户活动分类方面，Liao 等（2007）、Patterson 等（2003）提出层次化的条件随机场推断模型推断用户行为。Yin 和 Wolfson（2004）提出利用动态贝叶斯网络推断用户行为。

轨迹模式挖掘是指发现轨迹中反复出现的模式，其主要研究对象包括伴随模式、序列模式和周期模式。伴随模式挖掘是指发现轨迹数据库中共同移动的目标群组。已有研究人员面向同向移动（Li 等.，2010）、旅行陪伴（Tang 等.，2011；Tang 等.，2012）和聚集（Zheng 等.，2014）等多种共同移动模式提出挖掘方法。发现伴随模式可以帮助物种迁徙、军事监控和交通事件检测等领域的研究。序列模式挖掘是指发现有限目标轨迹中，访问位置相同的序列的集合，且序列间的时间间隔也相近。Xiao 等（2010）提出了基于图的序列匹配算法用于发现两个用于轨迹的序列模式。Song 等（2014）提出基于 LCSS 距离和前缀树的序列模式挖掘算法，用于发现路网约束下的序列模式。周期模式是指发现轨迹中按周期循环出现的序列。Li 等（2010）提出两步走的方法检测周期模式，首先发现目标经常访问的参考点，并将轨迹转换为进出参考点的 0 或 1 序列；然后基于傅里叶变换发

现轨迹中的周期模式，并基于层次聚类汇总。Li 等（2012）改进 Li 等（2010）中的方法使之适用于不完整的稀疏轨迹。

2.2.2 轨迹补全

根据数据特点和场景的不同，轨迹补全的研究可分为基于单体目标轨迹的方法和基于集体目标轨迹的方法。

在基于单体目标轨迹的轨迹补全方面，研究人员利用单个目标的移动轨迹，建立目标移动模型从而补全目标轨迹。该类方法通常对目标的移动模式做出假设，如速度、加速度、距离之间满足的物理运动规律，基于静默时间段前后位置记录的插值估计目标位置。该类方法包括（Serrano 等.，2017；Long，2016；Chandler 等.，2015）。Chandler 等（2015）利用数据缺失前后的记录，通过线性插值的方法，对轨迹的缺失部分进行平滑，达到补全缺失轨迹的目的。Long（2016）提出了运动学插值的方法补全轨迹，该方法利用速度与加速度，估计目标的缺失轨迹。Serrano 等（2017）将运动学插值的方法应用到移动机器人控制领域，用于跟踪机器人轨迹。

另外，一些基于单体目标轨迹的补全方法，通过引入额外信息将单体目标的轨迹与额外信息结合提高轨迹补全精度。该类方法包括（Li 等.，2015a；Su 等.，2015）。Li 等（2015）将目标轨迹与城市路网信息结合，首先对城市中的轨迹进行路网匹配，对于轨迹中的静默时间段，该方法在路网约束下补全目标缺失位置。Su 等（2015）也利用路网信息，推断目标在静默时间段的位置，实现轨迹在路网上的对齐及补全。该方法首先在城市路网中选取若干个位置作为标准位置，对于整个城市标准位置构成一个无向图；然后将轨迹中的目标位置映射到标准位置上。映射的过程中保证轨迹中相邻标准位置直接联通，从而保证轨迹中目标位置的对齐及完整。

在基于集体移动模式的轨迹补全方面，针对单体轨迹存在的问题，一些研究人员提出利用目标群体的历史轨迹来训练一个普适的模型来挖掘目标的移动模式。该类方法可分为两类，基于模式的方法和基于预测或推荐的方法。基于模式的方法通过聚类或模式挖掘，生成模式相似的轨迹集合及集合的中心轨迹；然后将轨迹映射到相应集合中，并利用集合中心的轨迹补全缺失轨迹。该类方法包括

（Li 等.，2016；Yi 等.，2016；Silva 等.，2015）。Li 等（2016）直接从历史数据发现目标活动规律补全缺失轨迹，在没有引入路网信息的前提下达到与基于路网方法相当的效果。该方法将历史轨迹中的目标位置记录看作 GPS 点集（GPS point cloud），首先利用 L1-骨架特征检测 GPS 点集中的交汇点和交汇路径；然后基于交汇点和路径构建交汇点网络；最后将轨迹与交汇点网络进行匹配，利用交汇点补全缺失轨迹。Silva 等（2015）首先利用 K-Means 算法对所有 GPS 位置进行聚类，得到 K 个 GPS 位置的簇及 K 个标准位置；然后将轨迹对齐到标准位置，即将轨迹中的每个位置用与其最近的标准位置替换；最后利用标准位置的邻近关系，补全轨迹中的缺失位置。Yi 等（2016）提出了一种基于多视角学习的方法补全位置传感器收集到的数据，该方法从整体时间、整体空间、局部时间、局部空间四个视角建模不同位置传感器序列的关系，并利用多视角学习补全序列缺失位置，该方法可应用于轨迹补全。

另外，基于集体移动模式的轨迹补全方法也包括位置预测或推荐的方法。研究人员提出此类方法的目标不是补全缺失轨迹，但此类方法可以应用于轨迹补全问题。该类方法根据利用的技术的不同可分为基于矩阵分解的方法和基于序列模型的方法。基于矩阵分解的方法（Chen 等.，2015；Feng 等.，2015；Zhang 等.，2015；Han 等.，2017；Yang 等.，2017）是静态的方法，该类方法的预测和推荐结果与轨迹信息无关，仅与特定目标相关。该类方法首先利用历史数据构建目标-位置访问矩阵，然后利用矩阵分解分别学习目标和位置的表示向量，最后基于表示推荐或预测下一个位置。Chen 等（2015）提出面向位置种类多样化的位置推荐算法，通过引入多样化正则项，使推荐结果包含更多类型的位置。Feng 等（2015）利用基于排序的度量学习学习位置和目标的表示向量，个性化地为目标推荐下一个访问的位置。另一类基于序列模型的方法（Feng 等.，2018；Liao 等.，2018；Yao 等.，2017；Liu 等.，2016；Monreale 等.，2009）是动态的方法，该类方法通过 RNN 或隐马尔科夫模型建模目标活动的序列特性，从而推荐或预测目标未来可能访问的位置。其中 Feng 等（2018），Liu 等（2016）和 Liao 等（2018）基于 RNN 建模目标活动轨迹，学习目标的移动模型后基于该模型预测目标未来的位置。Yao 等（2017）、Monreale 等（2009）基于隐马尔科夫模型预测目标未

来位置。

现有轨迹补全工作都不适用于稀疏轨迹的补全。基于单体目标轨迹的方法利用运动特征结合路网信息通过插值等方法补全轨迹，当轨迹中静默时间间隔比较长如若干小时时，该类方法的补全结果误差很大。通过集体移动模式补全轨迹方法中，基于模式的方法通过聚类或模式发现来挖掘移动模式相似的轨迹集合，但此类方法也是针对稠密轨迹，对于稀疏轨迹无法获得有意义的移动模式。基于位置推荐和预测的方法可以应用于稀疏轨迹，但此类方法不是针对轨迹补全问题设计，仅能利用缺失轨迹静默时间段之前的数据推断缺失位置。

2.2.3 轨迹相似度计算

本节从两个方面概述与轨迹相似度计算相关的现有研究：（1）轨迹相似度计算传统方法；（2）加速轨迹相似度计算方法。

轨迹相似度计算传统方法。该类方法根据轨迹的时间和空间属性，利用已有的序列相似性度量方法，计算给定两条轨迹相似度的值。与序列相似度计算方法一致，常见的轨迹相似度精确计算方法有 Fre´chet（Alt 等.，1995）、ERP（Chen 和 Ng，2004）、EDR（Yin 和 Wolfson，2004）、DTW（Yi 等.，1998）、LCSS（Vlachos 等.，2002）、Hausdorff（Atev 等.，2010）、CPD 和 SPD，其计算方式和特点见表 2.2。

表 2.2 轨迹相似性度量

Table 2.2 Trajectory similarity measures

名称	英文名	计算方式	特点
Fre´chet	Fre´chet Distance	按时间顺序连接轨迹中两点的最长距离	距离的值有物理意义，但计算开销大
EDR	Edit Distance on Real sequence	计算将一条轨迹编辑为另一条的开销	对噪声鲁棒，但计算量较大
ERP	Edit distance with Real Plenty	将EDR中点与点之间的离散编辑开销，用连续实数量化之后的编辑距离	可根据不同任务调整编辑开销
DTW	Dynamic Time Warping	轨迹对齐后计算所有轨迹对距离之和	可处理变长轨迹，但重排列增加计算噪声

（续表）

LCSS	Longest Common SubSequence	计算两条轨迹的最长子序列记录长度	对噪声鲁棒，但计算量较大，适用于密集轨迹
Hausdorff	Hausdorff distance	将轨迹看作点集，计算最小最大距离	有几何意义，但得到的距离具有方向性和不对称性
CPD	Closet-Pair Distance	计算两条轨迹的最小距离记录	计算简单，但没有考虑整个轨迹分布情况
SPD	Sum of Pairs Distance	计算两条轨迹记录对应点的距离之和	计算简单，但不能处理记录长度不同轨迹

其中最近点对距离（SPD）和点对距离之和（SPD）方法是最基础、直观的轨迹相似度计算方法，但都存在其局限性。SPD 没有考虑轨迹的整体分布，SPD 仅能处理长度固定的轨迹。除此之外，Hausdorff、LCSS、DTW、EDR、ERP 和 Fre´chet 距离等方法均通过匹配-汇总（alignment-summarization）的方式得到轨迹相似度，计算思想相同。但此类方法都面临计算复杂度高的挑战。给定两条长度分别为 m 和 n 的轨迹，这些度量方法的计算复杂度最低为 O(n*m)，很难适用于大规模轨迹数据集。因此需要加速轨迹相似度计算。

加速轨迹相似度计算方法。由于传统轨迹相似性度量方法计算复杂度高，研究人员提出许多技术加速轨迹相似度计算，其可以大致分为轨迹索引和近似算法两类。第一类方法为轨迹索引，该类方法包括（Shang 等.，2017；Gowanlock 等.，2016；Gong 等.，2015；Yazti 等.，2006；Chen 等.，2005），其思路是基于某一个轨迹相似性度量，如 Hausdorff，构造轨迹索引。在 Top-K 轨迹相似性检索中，通过减小涉及计算的轨迹的数目加速检索的响应时间。此类方法大多采用基于 K-D 树或 R 树的索引结构，层次化地组织轨迹数据。检索轨迹时，索引可以通过树剪枝的方法快速寻找与检索轨迹最相似的轨迹。使用索引和修剪技术来减少全局级别的计算数量。Chen 等（2005）提出了一个新的度量方法 EDR，并设计了三种索引结构加速基于 EDR 的相似轨迹检索。Yazti 等（2006）基于 LCSS 检索相似轨迹，该工作提出了一个分布式检索的框架及索引结果用于加速检索过程。Gowanlock 等（2016）提出了一个基于 Bounding Box 的轨迹索引结构，并针对 GPU 计算的特点做了特定优化，使轨迹相似性检索可以利用 GPU 加速。Shang

等（2017）提出了一个基于轨迹相似度合并重复轨迹的方法。该方法首先基于轨迹的空间和时序特性定义了一个新相似度计算方法，然后提出了一个时间优先的检索方法使检索过程可以并行，并利用基于空间网格的索引结构在检索时对不相似轨迹进行剪枝。

第二类方法为近似算法。该类方法针对某一种特定的相似性度量，设计其近似算法，降低其计算复杂度。针对 Hausdorff 度量，Farach-Colton 等人，和 Backurs 等人提出了基于表示向量的方法，将 Hausdorff 距离转化为 lp 空间中的向量距离。针对 DTW，Thanawin 等人提出可以通过省略计算过程中的平方操作，加速 DTW 计算的方法。近期，Driemel 等人提出了一种基于局部敏感性哈希（LSH）的算法，用于快速计算 Fre´chet 和 Hausdorff 距离。

现有加速轨迹相似度计算的方法没有降低计算复杂度或仅适用于特定相似度计算方法。基于轨迹索引的方法专门针对 Top-K 相似性检索问题而设计。该类方法没有降低计算轨迹对相似度的复杂度，不能应用于需要所有轨迹对距离（pair-wise distance）的任务，例如轨迹聚类和异常检测。通过设计近似算法加速相似度计算的工作仅针对一两种具体相似度设计，很难将这些技术用于其他相似度计算方法。

2.2.4 轨迹异常检测

现有轨迹异常检测方法可以分为两类：基于模式的轨迹异常检测方法和基于半监督学习的轨迹异常检测方法。

基于模式的轨迹异常检测方法。该类方法利用已收集的历史轨迹数据，对目标的行为模式建模，发现目标移动的正常模式，并将与目标正常行为模式不同轨迹作为异常轨迹。该类方法依据获得目标正常模式方法不同分为基于轨迹聚类的方法、基于模式挖掘的方法和基于人工先验知识的方法。

基于轨迹聚类的方法通过轨迹聚类发现移动模式相同的轨迹簇，将轨迹簇中心的轨迹作为正常行为模式，并通过与轨迹簇中的轨迹对比检测异常轨迹。该类方法包括（Bao 和 Du，2018；Guo 等.，2014；Laxhammar 和 Falkman，2012；Cheng 和 Li，2006）。Cheng 和 Li（2006）将移动社交网络轨迹异常检测分为聚类、比较、确认的三个阶段，基于主题模型对用户活动轨迹进行聚类，然后监控

同一时空范围内主题的变化，并通过检测该区域内主题与其他轨迹不同的轨迹作为异常轨迹。Laxhammar 和 Falkman（2012）利用滑动窗口划分子轨迹，并提出局部异常因子对统一滑动窗口的轨迹聚类，最后将不属于同一簇的子轨迹检测为异常轨迹。Guo 等（2014）基于轨迹的形状和速度，利用核密度估计表示轨迹，并通过最大化聚类簇之间的互信息得到轨迹聚类，最后基于香侬熵判断轨迹是否为异常轨迹。Bao 和 Du（2018）首先通过 DBSCAN 得到轨迹的模式，然后利用点到轨迹的 Haversine 距离判断轨迹是否为异常。朱燕等（2017）则通过提取相同起止点的轨迹集合，基于轨迹间的相似性度量实现异常轨迹的检测。

基于模式挖掘的方法首先基于历史数据挖掘轨迹中存在的模式，如周期模式、伴随模式、序列模式等，然后基于人工经验设定阈值或基于异常数据训练检测模型检测异常轨迹。Lei（2016）通过模式挖掘的方式，发现同类型目标轨迹在空间、序列和行为上的正常移动模式，并基于此检测异常轨迹。Laxhammar 和 Falkman（2014）基于 Hausdorff 距离定义轨迹异常因子，并通过该因子检测轨迹流中的异常轨迹。Li 等（2006）基于移动目标在时间、空间上的通用移动行为模式 motif，提出 Motion-Alert 方法。该方法首先将轨迹转换成一系列 motif 组成的序列，然后利用 SVM 提取 motif 特征，并学习异常检测分类器。Bu 等（2009）利用滑动窗口技术，持续监控各个时段内明显不同于其空间近邻的轨迹，作为异常子轨迹。Yu 等（2014）提出了一种检测轨迹流中异常轨迹的增量算法 INC。该算法定义了基于点近邻和轨迹近邻的两种异常，并引入了 3 种优化原则，时间意识的检查原则(TAE)、最小支持检查原则(MSE)以及终身触发检测原则(LTD)检测异常轨迹。

基于人工先验知识的方法利用专家提供的先验知识，确定目标移动的正常模式和阈值检测异常轨迹。Ge 等（2011）利用证据理论基于专家知识开发了检测出租车欺诈行为的系统，该系统可以检测基于旅行路线和旅行距离的两类驾驶欺诈行为。Liu 等（2014）提出了基于速度的出租车欺诈行为检测系统，该系统采用基于速度信息的聚类方法建模出租车的正常模式，并基于正常模式检测出租车异常驾驶行为。Zhu 等（2015）提出了依赖时间的异常轨迹检测算法 TPRO，该方法将时间段相近的最热门的 K 条轨迹作为正常轨迹，将每条轨迹与其相关的

正常轨迹进行比较，路径差异较大的轨迹作为异常轨迹。

基于半监督学习的轨迹异常检测方法。该类方法目前工作较少，通过少量的轨迹标签信息，推断无标签轨迹的标签，并利用无标签轨迹中蕴含的信息辅助异常轨迹检测。该类方法包括 Sillito 和 Fisher，（2008）和 Arnaud 等（2013）。Sillito 和 Fisher，（2008）提出一个启发式的半监督异常轨迹检测框架，该框架基于高斯混合模型检测建模轨迹的生成过程，对于一条新的轨迹该框架首先估计其生成概率，并将概率低的轨迹交人工判别是否为异常，然后根据人工判别结果调整混合模型参数。Arnaud 等（2013）关注语义轨迹的异常检测问题，提出基于概念的异常检测系统。该系统首先将原始轨迹切分并进行语义标注，得到轨迹的语义事件和语义行为，然后将与历史行为不一致的轨迹交由人工判别，确定其是否为异常。

现有轨迹异常检测方法没有解决数据标签稀少下的轨迹异常检测问题。基于模式的轨迹异常检测模型大多通过无监督的方式检测异常轨迹，该类方法难以与有标签数据结合，检测异常轨迹。基于半监督学习的轨迹异常检测方法尚处于初步研究阶段，目前还依赖启发式的人工判别检测异常轨迹。

2.3 小结

本章对轨迹数据挖掘的研究工作进行了总结；首先概述了轨迹数据的定义及分类；然后基于轨迹数据挖掘的研究框架，重点阐述了现有轨迹静默时间补全、轨迹相似度计算，轨迹异常检测三类任务的国内外研究现状，分析了现有研究工作已取得的成果和存在的不足。

第 3 章　上下文信息感知的稀疏轨迹补全方法研究

本章针对稀疏轨迹缺失位置难以补全的问题，提出了上下文信息感知的稀疏轨迹补全方法 TrajCom。TrajCom 利用历史轨迹中蕴含的目标移动规律构建补全模型，提高稀疏轨迹补全的准确率。TrajCom 首先根据待补全轨迹的时空约束和目标偏好信息，从历史轨迹数据库中筛选出相关轨迹。然后，利用深度轨迹编码模型得到时间感知的轨迹编码，并基于注意力机制建模相关轨迹与待补全轨迹的相关程度。最后，TrajCom 根据注意力权重估计目标轨迹的缺失位置。

本章的组织如下：3.1 节介绍了本章的研究问题和挑战。3.2 节详细介绍了上下文信息感知的稀疏轨迹补全方法，其中，3.2.1 节介绍了问题定义及方法概述；3.2.2 节介绍了候选位置及相关轨迹筛选模型；3.2.3 节介绍了深度轨迹编码模型，并重点介绍了全新的时间间隔感知的循环神经网络单元 tGRU。3.2.4 介绍了缺失位置估计方法和 TrajCom 的参数优化算法。3.3 节通过真实数据集上的实验验证了 TrajCom 的有效性。3.4 节对本章进行了小结。

3.1　引言

轨迹数据在许多数据驱动任务如自动驾驶、机器人中有着广泛的应用。然而轨迹应用中存在的一个重要问题是轨迹数据的缺失和不确定。这个问题的原因在于原始轨迹数据的采集过程往往是稀疏和不完整的。比如，出租车轨迹数据的采集需要通过基站收集 GPS 的定时播报，这种收集方式中信号不稳定可能会造成轨迹位置的缺失；在社交网络中用户通过定点打卡的方式上报所处的位置，这种方式得到相邻记录间的时间间隔往往是数小时，使得轨迹数据非常稀疏。为了解决这种由不稳定信号或采集模式导致的目标位置缺失、不确定的问题，轨迹补全成为了近几年来轨迹数据挖掘的研究热点。轨迹补全的目标在于推测某条特定轨迹，在特定时刻缺失的轨迹位置。具体来说，如图 3.1 所示，给定了目标u的一条特定轨迹T和一个特定时间点t^*，轨迹补全目标是估计u在时刻t^*最可能所处的位置。

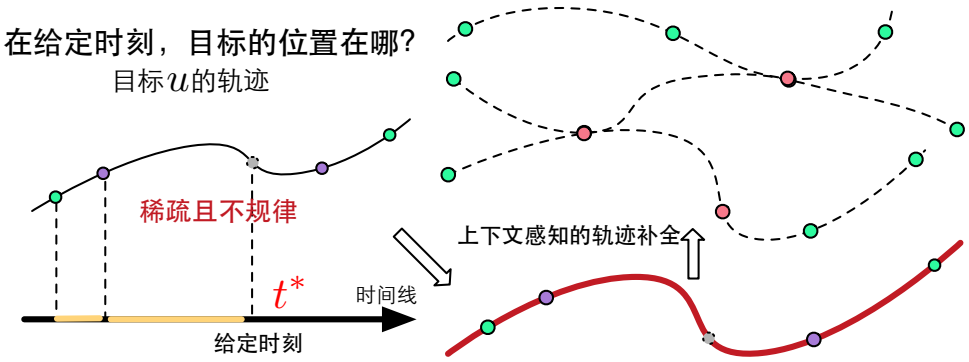

图 3.1 轨迹补全问题示意图

Figure 3.1 Illustration of trajectory completion problem

现有的轨迹补全方法可以分为两类：基于单体目标的补全方法和基于集体移动模式的补全方法。基于单体目标轨迹的补全方法只考虑单条目标轨迹轨迹，通过运动学假设建模轨迹的移动规律从而推断目标缺失位置。这类方法通过线性插值或者运动学插值可以补全轨迹，但此类方法仅适用于时间间隔较短的稠密轨迹。在稀疏轨迹数据中，这类方法往往是无效的。其原因在于稀疏轨迹静默时间间隔较长，使得上述方法中的运动学假设失效。基于集体移动模式的补全方法利用序列模型（如隐马尔科夫模型或者 RNN 模型）来描述目标的移动规律。补全过程分为训练和推断两步。在训练时，将轨迹作为输入，学习目标的移动规律；推断时，基于训练好的移动模型，对目标轨迹进行补全。这类方法假设所有的记录在时间维度上都是均匀的，即所有的连续记录点之间的时间间隔是相等的。然而，由于轨迹的异频采样性，这样的假设在实际应用中很难成立。因此，该类方法无法建模稀疏轨迹中时间间隔的变化情况，不能补全给定时刻的缺失位置。

研究轨迹数据的补全需要重点解决如何在时间、空间等约束条件下，合理利用历史轨迹数据库中与待补全轨迹相关的轨迹集合来对轨迹缺失部分进行补全。该问题涉及三个方面：第一，如何合理地从海量历史轨迹数据库中筛选出候选位置和相关轨迹集合；第二，如何自动地从相关候选轨迹中抽取有效特征来进行轨迹补全，该特征抽取过程不仅需要考虑轨迹的时空序列性，还要考虑不同记录之间时间间隔的不一致性；第三，如何利用抽取到的特征来估计稀疏轨迹中的缺失位置。

针对现有方法存在的问题，本文提出了一种上下文信息感知的轨迹补全方法

TrajCom。该方法属于基于集体移动模式补全方法的一种，采用筛选-编码的框架补全轨迹。该方法分为三个步骤。

首先，候选位置及相关轨迹筛选。TrajCom 采用了基于上下文指导的神经过滤策略(context-guided neural filtering)筛选候选位置和相关轨迹。上下文信息是缺失轨迹在给定时刻前后记录的位置、时间以及目标偏好信息。利用上述信息，TrajCom 从历史轨迹数据库中过滤并筛选了一系列的候选位置和相关轨迹，筛选过程与缺失轨迹密切相关。然后利用这些相关轨迹的移动模式补全缺失轨迹在给定时刻的位置。该过程可以发现历史轨迹中与给定时刻上下文信息相似的轨迹。这些轨迹可以缓解数据稀疏问题，有利于目标轨迹的补全。

其次，上下文感知的轨迹编码。基于循环神经网络，TrajCom 将目标轨迹和相关轨迹中的记录编码为表示向量。本文提出了一个时间间隔感知的循环神经网络单元 tGRU 用于轨迹编码。该单元可以捕捉轨迹记录的时间间隔信息，从而学习到与缺失轨迹给定时刻上下文信息相关的移动规律。另外，基于 tGRU 的编码器采用双向 RNN 进行编码，编码过程融合了位置信息、时间段信息和目标偏好等多种影响因素。这样轨迹中每一条记录的编码都蕴含整条轨迹的信息。

最后，基于轨迹记录编码，本文提出了流行度加权的注意力机制，该机制不仅可以用于相关轨迹的加权从而提高预测的准确率，而且可以追溯补全结果的生成过程，即 TrajCom 不仅可以获得基于哪些历史轨迹补全当前轨迹中的缺失位置，还可以自动推断这些轨迹对最终补全结果的贡献度。TrajCom 的这种性质使得在利用其进行补全时可以方便地与专家交互，并通过人类经验知识获得更有意义的补全结果。

3.2　上下文信息感知的稀疏轨迹补全方法

本节详细介绍了上下文信息感知的稀疏轨迹补全方法 TrajCom。首先定义了轨迹补全问题并概述了 TrajCom 的主要步骤，然后详细阐述了 TrajCom 的模型结构和计算过程，最后介绍了 TrajCom 的参数优化的方法。

3.2.1　问题定义及方法概述

本章首先定义了稀疏轨迹补全问题，然后简要介绍了本文提出的上下文信息

感知的稀疏轨迹补全方法 TrajCom。

3.2.1.1 问题定义

表 3.1 符号及其含义解释

Figure 3.1 Explaination of parameters of models

符号	含义解释
u	某目标 u
$\mathcal{T}_{\mathrm{u}}$	目标 u 的轨迹集合
$\mathcal{T}$	整体轨迹集合
T_u^a	目标u的某条轨迹a
t^*	指定待补全时刻
l^*	待补全时刻的真实位置
$\hat{l}$	待补全时刻的估计位置
$\mathrm{L}_{\mathrm{u}}^{\mathrm{a}}$	关于T_u^a和t^*的候选位置集合
$\mathrm{T}_{\mathrm{u}}^{\mathrm{a}}$	关于T_u^a和t^*的相关轨迹集合

对于 M 个目标$U = \{u_1, \dots, u_M\}$和 N 个地点$L = \{l_1, \dots, l_N\}$，每个目标u都对应一条记录序列$S_u = [r_1, \dots, r_t]$，其中每条记录$r_i = (l_i, t_i)$是一个二元组包含了位置$l_i \in L$和时间戳t_i。

轨迹定义：给定目标u的记录序列S_u以及时间间隔阈值$\delta > 0$，目标u的一段记录$T_u = [r_i, r_{i+1}, \dots, r_{i+k}] \in S_u$是一条轨迹，需要满足以下条件：

$$\forall j, 1 < j \leq k: t_j - t_{j-1} \leq \delta \tag{3.1}$$

上述条件约束了轨迹中相邻记录的时间间隔不能太长。根据轨迹定义，每个目标u的记录序列可以被分割并构建轨迹集合$\mathcal{T}_u = \{T_u^1, T_u^2, \dots\}$。通过合并 M 个用户的轨迹集合可以得到整体轨迹集合$\mathcal{T}$。给定目标u的一条轨迹$T_u^a \in \mathcal{T}_u$和时间点$t^* \in (t_i, t_{i+k})$，轨迹补全是估计u在t^*时刻的位置，即：

$$\hat{l} = \arg max P(l | T_u^a, t^*, \mathcal{T}) \tag{3.2}$$

其中t_i和t_{i+k}是轨迹T_u^a的起始和终止时间。

3.2.1.2 方法概述

如图 3.1 所示，轨迹补全的目的是估计目标u在时刻t^*所处的位置。利用 TrajCom 进行轨迹补全需要解决三个子问题：第一，如何合理地从海量轨迹数据库中筛选候选位置和相关轨迹；第二，如何自动地从相关轨迹中抽取与轨迹补全相关的有效特征；第三，如何利用抽取出的特征估计缺失位置。

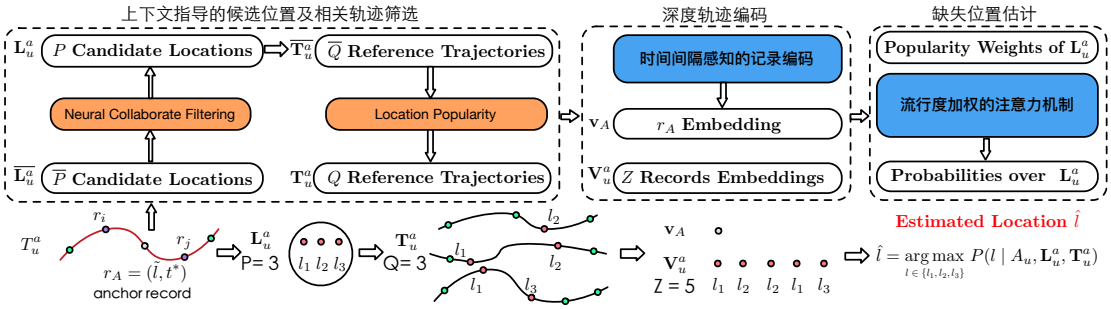

图 3. 2 TrajCom 示意图

Figure 3. 2 Architecture of TrajCom

为了解决上述三个问题，本文基于神经网络过滤-编码框架提出了 TrajCom。如图 3.2 所示，该方法分为三个步骤：

- 第一步，TrajCom 利用上下文指导的协同过滤(context-guided neural filtering)建模目标移动的时空约束和目标倾向性约束，筛选候选位置和相关轨迹。
- 第二步，由于特征抽取过程不仅需要考虑轨迹的时空序列性，也要考虑相邻记录时间间隔的不一致性，因此本文提出了时间间隔感知的编码模型。该模型将目标轨迹和相关轨迹的每一条记录映射到向量空间中。编码模型以 RNN 为基础，提出时间感知的循环神经网络单元 tGRU，该单元可以显式地建模时间间隔信息。
- 第三步，根据第二步得到的轨迹记录表示向量，本文提出了一个流行度加权的注意力机制，该机制不仅可以对表示向量进行加权，更好地提高预测的准确率，而且可以追溯 TrajCom 利用哪些轨迹生成补全结果，进一步解释轨迹数据缺失的原因。

以上三个步骤共同构成了一个端到端的轨迹补全方法。候选位置和相关轨迹筛选模块，可以有效地筛选出与目标相关的候选位置和相关轨迹序列。结合第二步和第三步中的轨迹编码和位置估计方法，使得 TrajCom 适用于补全稀疏轨迹。

3.2.2 候选位置及相关轨迹筛选

本节首先介绍了 TrajCom 中的候选位置筛选的过程，然后介绍了利用候选位置生成相关轨迹的方法。

3.2.2.1 候选位置筛选

候选位置筛选过程利用给定轨迹T_u^a在特定时间点t^*的上下文信息，对目标u在时刻t^*可能访问的位置进行筛选。筛选过程基于了移动目标u的时空约束和目标倾向性约束。下面将分别介绍这两种约束的建模方式。

首先，目标的移动行为会受到时间-空间约束。对于处于某个具体位置的目标来说，其下一条记录的位置受限于当前时刻位置以及与下一条记录的时间间隔。本文对目标的时间-空间限制进行如下的建模：

$$f(x) = \alpha \cdot log\ (\beta \cdot x + 1) \tag{3.3}$$

其中x是相邻记录的时间间隔。该函数表示在时间间隔为x的情况下，序列上相邻两条记录之间的最大距离。其中的参数α和β是与轨迹数据集有关的参数。对于不同类型的轨迹数据，参数的值也不同。

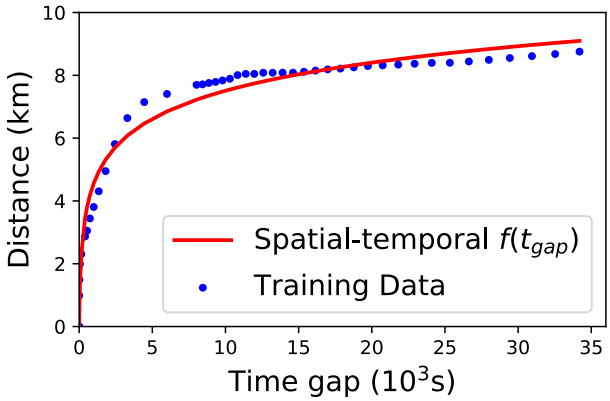

图 3.3 Foursquare 中时间-空间约束函数示意图

Figure 3.3 Distance statistics of data in Foursquare dataset

对于特定数据集，参数α和β的选取过程如下：首先，统计每条轨迹中相邻记录间的距离及时间间隔。然后，计算距离 95%分位数随时间变化的样本点，并用它们来拟合上述的函数。图 3.3 是公开数据集 Foursquare 上的拟合结果，其中蓝点代表着真实的轨迹记录间隔值，红线是函数拟合曲线。结果表明，拟合函数与轨迹相邻记录距离的分布相对一致。

得到轨迹相邻记录距离的分布趋势后，给定任意一条轨迹T_u^a，以及该轨迹中的任意两个记录$r_i = (l_i, t_i), r_j = (l_j, t_j)$，可以通过以下约束条件初步筛选记录间的候选位置：

如图 3.4 所示，候选位置集合$\overline{\mathrm{L_u^a}} = \{l_1, \ldots, l_p\}$是全体位置的子集，对于任意候选位置$l_p \in \overline{\mathrm{L_u^a}}$：$dis(l_i, l_p) < f(t^* - t_i)$且$dis(l_j, l_p) < f(t_j - t^*)$，本文使用 Haversine 距离作为$dis(.,.)$度量空间距离。该约束可以解释为候选位置与前后位置的间距保持在一定范围内。

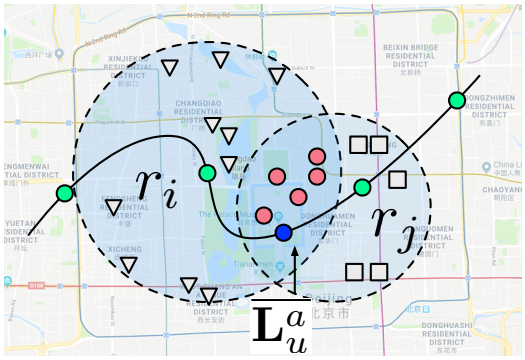

图 3.4 候选位置示意图

Figure 3. 4 Illustration of initial candidate locations

然而，仅利用时空约束将会生成大量的候选位置，不利于后续计算过程。目标的移动行为还受到目标倾向性的约束，即该目标在移动过程中更倾向于访问一些特定的位置。这种倾向性带有目标本身的语义信息，可以利用目标的历史轨迹数据获得。对于这种约束，本文采用神经协同过滤模型 NCF(Neural Collaborative Filtering)建模，并将其作为限制条件加入候选位置筛选的过程中。NCF 首先构建了一个二值矩阵B，矩阵中的每个值$\mathrm{b}_{u,l}$取值为 0 或者 1 来表示了目标u是否访问过位置l。然后利用矩阵分解的方式将矩阵B分解为目标特征矩阵E_u^{NCF}以及位置特征矩阵E_l^{NCF}。其中矩阵分解利用矩阵 B 进行优化，即：

$$\mathrm{b}_{u,l} \approx \mathrm{logistic}(\mathrm{MLP}(e_l^{NCF}, e_u^{NCF})) \tag{3.4}$$

给定用户u，TrajCom 利用 NCF 可以得到与u相关的位置排序列表并将该列表作为目标倾向。假设限制候选位置集合$\mathrm{L_u^a}$的最大数量为 P 个，那么$\mathrm{L_u^a}$可以由如下方式得到：顺序检查用户u位置排序列表中的位置，如果该位置处于$\overline{\mathrm{L_u^a}}$中且$\mathrm{L_u^a}$中位

置数量小于 P，那么将该位置加入到L_u^a中。依照此方式直至L_u^a含有 P 个位置或位置列表遍历完成。

3.2.2.2 相关轨迹筛选

利用候选位置集合L_u^a，TrajCom 从历史数据库中检索经过候选位置的轨迹作为相关轨迹。具体来说，初始相关轨迹集合$\overline{T_u^a}$为$\mathcal{T}$的子集合，$\overline{T_u^a}$中的轨迹需要至少经过一个L_u^a中的位置。但上述过程将会产生大量的相关轨迹。因此，TrajCom 通过候选位置的流行度对相关轨迹的数量进行限制，候选位置流行度计算方法定义如下：

$$g(l)=\begin{cases}1, & n_l \le \gamma \\ \min\left\{\frac{n_l}{\gamma}, 3\right\}, & n_l > \gamma\end{cases} \tag{3.4}$$

γ是人为指定的超参数，其物理意义是每个候选位置对应相关轨迹的最大数目，而n_l则是经过指定位置l的所有相关轨迹数目。给定缺失时间点的前后位置，TrajCom 利用上述方法筛选相关轨迹集合T_u^a。而$g(l)$也将作为后续估计缺失位置的权重。

3.2.3 深度轨迹编码

利用 RNN 建模轨迹的序列特性，TrajCom 中的深度轨迹编码模块将轨迹序列中的每一条记录编码为表示向量。由于轨迹补全需要估计t^*时刻的位置，因此轨迹编码需要考虑时间间隔的变化，而传统 RNN 模型不能建模这种变化。本文提出了一个时间感知的循环神经网络单元 tGRU 用于建模空间特征和时间间隔的变化。以 tGRU 为基础的深度轨迹编码器分为两层：输入编码层和序列编码层。

3.2.3.1 输入编码层

对于目标u的一条轨迹$T_u=[r_1,\dots,r_k]$，TrajCom 首先对其中的记录$r_k=(l_k,t_k)$进行输入层编码，并将编码后的向量e_k作为 RNN 每个时刻的输入。编码记录时，TrajCom 考虑目标类别、位置和时间三方面的信息。其中目标u和位置l的都采用 one-hot 向量表示，向量的维度是目标个数或位置个数，其中与记录相关的目标和位置对应的维度为 1，其他维度为 0。类似的，时间戳被按小时和工作日情况划分，得到 2*24 种时间段。这样，时间信息t_k也可以由长度为 48 维的

one-hot 向量表示。t_k中对应时间段的维度为 1，其他维度为 0。于是，对于任意一条记录r_k，其输入编码e_k可由如下计算得到：

$$e_k^u = u \cdot E_u;\ e_k^l = l_k \cdot E_l;$$
$$e_k^t = t_k \cdot E_t;\ e_k = [e_k^u; e_k^l; e_k^t] \quad (3.5)$$

其中$[\cdot;\cdot]$为向量间的连接操作，而E_u, E_l, E_t则分别是目标、位置和时间的表示矩阵。

3.2.3.2　序列编码层

为了捕捉轨迹的序列关系，同时也为了建模位置与变化时间间隔下的关系，本文提出了一种时间感知的循环神经网络单元 time-aware GRU，简称为 tGRU。tGRU 对轨迹序列特征的编码机制与传统 GRU 相同，其不同之处在于对时间间隔变化的编码。

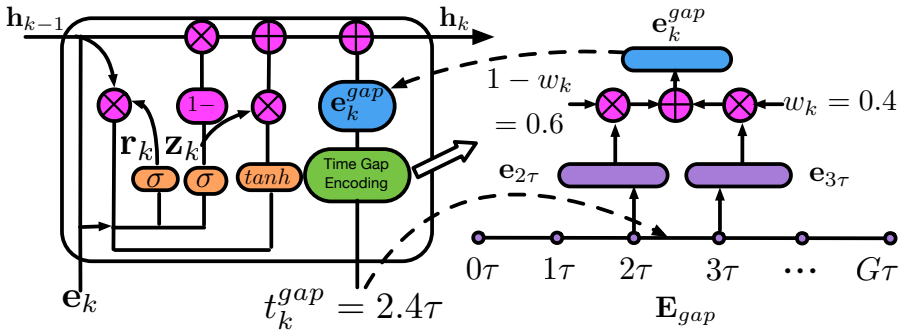

图 3.5 tGRU 示意图

Figure 3.5 Architecture of tGRU

对于目标u的一条轨迹记录序列$T_u = [r_i, r_{i+1}, \ldots, r_{i+k}] \in S_u$，在给定的时间间隔阈值$\delta > 0$前提下，$T_u$满足$\forall j, 1 < j \leq k \ \ s.t.\ t_j - t_{j-1} \leq \delta$。该约束保证轨迹中相邻记录的时间间隔都小于$\delta$。由于轨迹的异频采样性，相邻记录的时间间隔不一，tGRU 采用如下方式对时间间隔进行编码。tGRU 首先将时间间隔$[0, \delta]$划分为G个等长的时间片段$\tau = \delta / G$,从而就获得到了$G+1$个断点，并为每个端点维护一个轨迹表示向量，得到端点的表示向量矩阵$\mathbf{E}_{\text{gap}}$；对于每一个记录r_k，时间间隔的值为 $t_k^{gap} = t_k - t_{k-1}$。为了建模时间间隔上的先后语义关系，对于任意的时间间隔值t_k^{gap}，它将由其前后两个时间间隔点的向量表示通过加权相加得到。计算方式如下：

$$\boldsymbol{i_k^p} = \left\lfloor \frac{t_k^{gap}}{\tau} \right\rfloor;\ \boldsymbol{i_k^s} = \left\lceil \frac{t_k^{gap}}{\tau} \right\rceil;\ w_k = \frac{t_k^{gap}}{\tau} - i_k^p;$$

$$e_k^{gap} = (1 - w_k) \cdot \boldsymbol{i_k^p} \cdot \mathbf{E_{gap}} + w_k \cdot \boldsymbol{i_k^s} \cdot \mathbf{E}_{gap} \tag{3.6}$$

其中，$\boldsymbol{i_k^p}$和$\boldsymbol{i_k^s}$分别是它前后两个时间断点的索引，向量$\boldsymbol{i_k^p}$和$\boldsymbol{i_k^s}$则是索引所对应的 one-hot 编码。$\mathbf{E}_{\text{gap}} \in \mathrm{R}^{(G+1)\times d_{gap}}$是$\mathrm{G}+1$个时间断点的向量表示矩阵。图 3.5 举例说明了 tGRU 对于时间间隔的编码，假设$t_k^{gap} = 2.4\tau$，那么时间间隔编码就是$e_{2\tau}$和$e_{3\tau}$的带权线性拼接，而这里的权重$w_k = 0.4$。

时间间隔的编码与基于 GRU 的时序关系编码融合得到第 k 条记录的隐含状态，即融合 GRU 输出h'_k和时间间隔编码向量e_k^{gap}。tGRU 循环神经网络单元在处理轨迹中的第 k 条记录r_k时，执行如下计算生成其隐含状态h_k:

$$h'_k = \mathrm{GRU}(e_k, h_{k-1});$$
$$h_k = h'_k + e_k^{gap} \tag{3.7}$$

对于一条记录r_k，TrajCom 采用双向 RNN 来获得这两个隐含状态表示$\overrightarrow{h_k}$和$\overleftarrow{h_k}$。假设 tGRU 的隐含状态的大小为h，那么对于记录r_k的表示向量v_k=[$\overrightarrow{h_k}$，$\overleftarrow{h_k}$]是长度为$2h$的向量，该向量包含当前处理轨迹的序列信息及局部的时间间隔信息。

3.2.4 缺失位置估计

本文基于上述轨迹记录表示向量，提出了流行度加权的注意力机制（popularity-weighted attention）。该机制将表示向量作为输入，通过注意力机制从候选位置中选取估计结果。受 Vaswani 等（2017）启发，本文采用了多头注意力机制（multi-head attention）来提升位置轨迹精度。流行度加权的注意力机制计算过程如下：

$$s_z = g(l_z) \cdot (f_A \otimes \mathrm{F}_u^a[z]);$$
$$s = [s_1, \ldots, s_Z];$$
$$\mathrm{P_R} = \sigma([s_1, \ldots, s_Z]);$$
$$\mathrm{P'_R} = mean\_by\_heads(\mathrm{P_R});$$

$$\mathrm{P_L} = sum\_over\_locations(\mathrm{P_R'}); \tag{3.8}$$

其中，$\mathrm{F}_u^a[z]$是第z条记录的表示向量所对应的第F个 head 的张量，$g(l_z)$则是l_z的流行度权重，⊗操作则是矩阵点乘操作，σ(.)是softmax操作，$\mathrm{P_R}$则是 multi-head 的概率分布，$\mathrm{P_R'}$是Z条记录的总概率，$\mathrm{P_L}$是到达某个地点$\mathrm{L_u^a}$的访问概率。最终选择$\mathrm{P_L}$中概率值中最高的位置作为对缺失位置的估计。

3.2.5　参数优化

算法 3.1：TrajCom 训练算法

输入：

轨迹训练集 D，筛选模型初始参数Θ_F，编码器初始参数Θ_E，

输出：

优化后的参数Θ_F及Θ_E

for $\forall \mathrm{T_u^a}, t^* \in D$ **do**
　产生$\mathrm{L_u^a}$和$\mathrm{T_u^a}$；设定计数器$\mathrm{count} = 0$
　while l^*不在$\mathrm{L_u^a}$ **and** $\mathrm{count} < \eta$ do
　　$\Theta_\mathrm{F} \leftarrow \Theta_\mathrm{F} - lr_s \cdot \nabla \mathrm{J_s}$
　　更新$\mathrm{L_u^a}$和$\mathrm{T_u^a}$
　　计数器$\mathrm{count} = \mathrm{count} + 1$
　end while
　$\Theta_\mathrm{E} \leftarrow \Theta_\mathrm{E} - lr_m \cdot \nabla \mathrm{J_m}$
end for
return Θ_F, Θ_E

TrajCom 中，给定任意一个训练样本$(T_u^a, t^*) \rightarrow l^*$，方法训练的损失包含了两个部分：（1）位置过滤损失$\mathrm{J_s}$用来鼓励目标位置l^* 位于选择的候选位置$\mathrm{L_u^a}$中；（2）位置估计损失$\mathrm{J_m}$可以保证l^*在$\mathrm{P_L}$中的概率高。TrajCom 使用了交叉熵损失（cross-entropy loss）来优化这两部分。其中$\mathrm{J_s}$和$\mathrm{J_m}$的形式定义如下：

$$\mathrm{J_s} = -\log\left(\mathrm{P}_{\overline{L}}[l^*]\right) - \sum_{l \in \overline{L_u^a} \setminus l^*} \log\left(1 - \mathrm{P}_{\overline{L}}[l]\right) + \frac{\lambda_1}{2}\left|\left|\Theta_F\right|\right|^2$$

$$\mathrm{J_m} = -\log(\mathrm{P_L}[l^*]) - \sum_{l \in L_u^a \setminus l^*} \log(1 - \mathrm{P}_L[l]) + \frac{\lambda_2}{2}\left|\left|\Theta_E\right|\right|^2 \tag{3.8}$$

其中，$\mathrm{P}_{\overline{L}}$是有 NCF 模型产生的$\overline{\mathrm{L_u^a}}$的概率分布，$\mathrm{P_L}$则是$\overline{\mathrm{L_u^a}}$的概率分布。$\lambda_1$和$\lambda_2$则是两个正则项参数。损失函数中$\Theta_\mathrm{F}$和$\Theta_\mathrm{E}$表示在 NCF 和 RNN 编码器中所有

可以训练的参数。为了使得 TrajCom 中的所有参数可以以端到端的方式训练，本文提出了如下的参数学习算法：

如算法 3.1 所示，对于每个训练样本，首先判断地点l^*是否在 NCF 的前 P 个位置。如果l^*在L_u^a中，TrajCom 仅仅需要优化J_m。否则，TrajCom 首先优化J_s并且更新Θ_F，直到$l^* \in L_u^a$或者到达了最大迭代数目η，然后再优化J_m。TrajCom 使用 RMSprop（Kurbiel 和 Khaleghian. 2017）优化位置筛选模型，并基于 BPTT 算法学习参数Θ_E。通过上述学习算法，TrajCom 可以实现端到端（end-to-end）地训练模型参数。

3.3 性能评估

为了评估 TrajCom 的准确率，我们在四个真实轨迹数据集进行实验，并对比了八种已有的轨迹补全方法。本节首先介绍了实验设置，然后展示并分析了实验结果。

3.3.1 实验设置

本节首先介绍了用于评测 TrajCom 方法的数据集，然后介绍了对比方法及评价指标，最后介绍了实验中用的到的参数设置。

3.3.1.1 实验数据

用于实验的四个公开数据集分别是：Foursquare in Tokyo(TKY)、Foursquare in United States(US)、GeoTweets、Geolife。图 3.6 和图 3.7 展示了这些数据集的数据分布和统计信息。

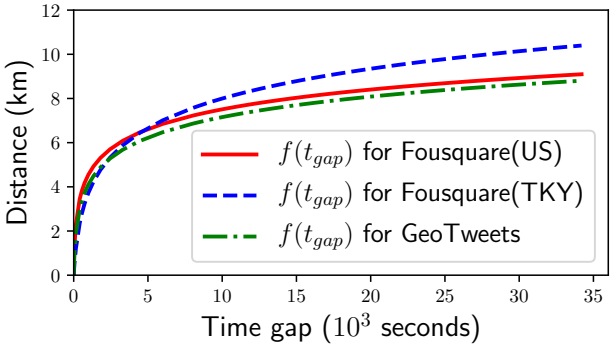

图 3. 6 数据集时间-空间约束函数

Figure 3. 6 Distributions of different datasets

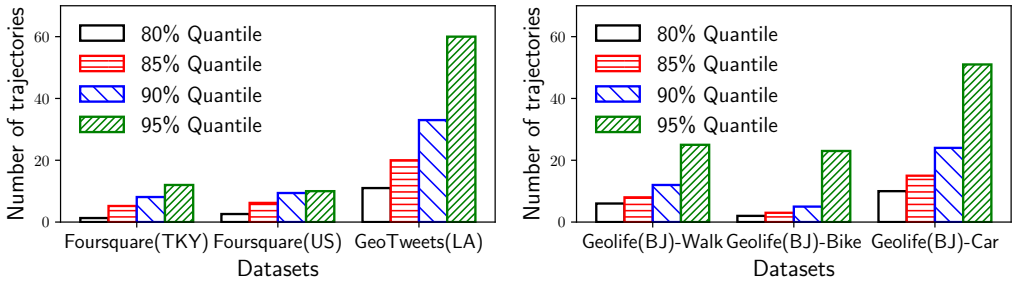

图 3. 7 数据集轨迹数量分布

Figure 3. 7 Quantiles of different datasets

表 3. 2 数据集统计数据

Table 3. 2 Statistics of four datasets

数据集	目标数量	位置数量	轨迹数量	平均长度	平均间隔
Foursquare（东京）	1083	38333	10k	12	2.3h
Foursquare（美国）	50813	501900	126k	8	3.1h
GeoTweets（洛杉矶）	3401	67210	13k	11	2.7h
Geolife（北京）	69	40000	8.5k	24	0.1h

Foursquare 数据集带有用户访问的 POI 信息，因此可以直接利用 POI 作为位置补全轨迹。对于 GeoTweets 和 Geolife 数据集，本文将轨迹所在空间区域划分为 500 米×500 米的网格，每个网格作为一个位置。预处理过程中，我们删除了上述所有数据集中位置记录低于 3 条的轨迹和记录少于 5 条的用户。训练数据方面，本文随机逐个删除原始轨迹中的记录，并将删除后的轨迹作为缺失轨迹，训练 TrajCom 补全删除的位置。对于每个数据集，TrajCom 随机选择 70%的轨迹用来构建训练集，并保留 5%和 25%的数据分别用于验证和测试。

3.3.1.2　对比方法及评价指标

TrajCom 仅仅与轨迹数据本身有关，并不依赖于其他的额外信息。因此，本文选择了八个现有的不依额外信息的轨迹补全方法作为对比方法：

- 最近邻方法（Nearest Locations）：该方法选择离目标时间最邻近记录的位置作为补全的位置。
- 最频繁位置（Most Frequent Location）：该方法会将目标最频繁访问的位置作为补全的位置。
- 基于聚类的位置预测（Clustering-based Methods）：该方法利用 K-Means 生成位置的聚类，利用聚类中心位置中离缺失轨迹最近的位置作为补全的位置。

- ST-RNN（Liu 等.，2016）：该方法是一个基于 RNN 的位置预测方法，它可以学习到转移矩阵中的时空信息，用于位置预测。
- SERM（Yao 等.，2017）：该方法是基于 RNN 的位置预测方法，使用了表示学习的技术联合建模不同影响因素对于目标移动的影响，从而预测后续位置。
- BPR（Rendle 等.，2009）：该方法是广泛使用的位置推荐方法，基于矩阵分解建模目标与位置之间的关系，并基于此模型推荐位置。
- 基于上下文的协同过滤方法（Context-guided Neural Filtering）：该方法利用了 TrajCom 位置筛选部分模型，通过协同过滤的方法来评估缺失点的位置。
- RankGeoFM（Li 等.，2015b）：该方法通过在因子分解中加入地理位置信息，基于排序损失优化模型进行位置推荐。

评价指标方面，本文使用两种最常用的评价指标来评价 TrajCom 的实验效果。第一种就是Hit Ratio @k (HR@k)，它用来评价目标点位置是否在预测的位置点序列的前k个位置中。第二个是评价指标是误差距离ϵ，它利用 Haversine 距离来计算估计地理位置的误差。

3.3.1.3 参数设置

TrajCom 中需要设置的参数分为两部分，第一部分位于候选位置和相关轨迹筛选中包括α，β，γ，P，这些参数可以决定筛选候选位置的精度；第二部分位于深度轨迹编码模型，包括各个影响因素表示向量的维度、时间间隔δ、时间间隔分割段的数目 G、多头注意力机制中 head 的数目和学习率。下面将分别介绍这两部分参数的设置。

表 3. 3 参数设置表

Table 3. 3 Hyperparameter values in TrajCom

数据集	α	β	γ	P	筛选 Acc
Foursquare（东京）	1.97	5.68	12	200	0.673
Foursquare（美国）	1.39	33.09	10	200	0.684
GeoTweets（洛杉矶）	1.34	22.45	60	50	0.932
Geolife（北京）-步行	固定距离 1.814km		25	—	0.907
Geolife（北京）-骑车	固定距离 2.489km		23	—	0.932
Geolife（北京）-开车	固定距离 4.227km		51	—	0.948

在候选位置及相关轨迹筛选中，α和β决定每个数据集的时空约束函数。对于社交网络数据集，如 Foursquare 和 GeoTweets，上述参数可以按照小节 3.2.2.1 中的方法得到。图 3.6 展示了不同数据集时空约束函数的差别。在人类活动数据集 Geolife 中，用户偏好对位置的影响很弱，且位置点之间时间间隔的分布比较均匀，本文采用固定的距离建模时空约束。γ和P可以被用来平衡涉及计算的相关轨迹数量和候选位置筛选准确率之间的关系。经过反复测试，实验所需参数设置情况汇总在表 3.3 中。

深度轨迹编码模型中，对于输入编码层的不同影响因素，TrajCom 采用相同维度的向量进行编码，将用户偏好、时间因素、空间因素和 NCF 中用户及位置表示向量的维度都设置为 50；时间间隔上，本文设置社交网络数据集中的$\delta = 10\text{h}$，并在编码时间间隔时将时间间隔分成 G＝20 个轨迹段；轨迹段编码的维度和 tGRU 中隐含状态的维度都被设置为 128；多头注意力机制及模型优化中，head 数目被设置为 8，最大优化次数$\eta = 20$，对于位置筛选和模型估计的学习率都设置为 0.01。

3.3.2　实验结果分析

本节分别介绍了准确率实验、变种实验和可解释分析实验的结果。

3.3.2.1　准确率实验结果

表 3. 4 Foursquare 和 GeoTweets（LA）上实验结果

Table 3. 4 Experimential results on Foursquare and GeoTweets(LA) datasets

方法	Foursquare（东京）		
	HR@1	HR@10	HR@20
NL	0.0094	0.0242	0.0692
MFL	0.0231	0.0732	0.1421
Cluster	0.0161	0.0330	0.0760
ST-RNN	0.0676	0.1400	0.2668
SERM	0.0716	0.1653	0.2962
BPR	0.0537	0.1280	0.2657
CNF	0.0507	0.1120	0.2311
RankGeoFM	0.0680	0.1409	0.2730
TrajCom	**0.0984**	**0.2231**	**0.3219**

（续表）

方法	Foursquare（美国）		
	HR@1	**HR@10**	**HR@20**
NL	0.0097	0.0180	0.0637
MFL	0.0220	0.0736	0.1401
Cluster	0.0140	0.0308	0.0723
ST-RNN	0.0641	0.1358	0.2528
SERM	0.0688	0.1592	0.2873
BPR	0.0542	0.1234	0.2607
CNF	0.0471	0.0980	0.2037
RankGeoFM	0.0618	0.1354	0.2702
TrajCom	**0.0880**	**0.2131**	**0.3169**

方法	GeoTweets（洛杉矶）			
	HR@1	**HR@10**	**HR@20**	$\epsilon(km)$
NL	0.3270	0.4368	0.4539	4.132
MFL	0.3480	0.4582	0.4772	3.721
Cluster	0.3482	0.4690	0.4921	3.980
ST-RNN	0.4328	0.6096	0.6502	2.498
SERM	0.4591	0.6267	0.6930	2.292
BPR	0.3719	0.5117	0.6893	2.599
CNF	0.3813	0.5320	0.6931	2.611
RankGeoFM	0.3950	0.5506	0.7122	2.512
TrajCom	**0.5670**	**0.7161**	**0.8614**	**2.130**

表 3. 5 Geolife 上实验结果

Table 3. 5 Experimential results on Geolife dataset

方法	Geolife（北京）-步行			
	HR@1	**HR@10**	**HR@20**	$\epsilon(km)$
NL	0.7054	0.8541	0.8894	0.678
MFL	0.1910	0.2539	0.3321	2.936
Cluster	**0.7376**	**0.8600**	**0.8981**	**0.645**
ST-RNN	0.6176	0.7300	0.8268	1.780
SERM	0.5716	0.6653	0.7962	1.845
BPR	0.2571	0.3879	0.5630	2.763
CNF	0.2693	0.4107	0.5828	2.663

（续表）

RankGeoFM	0.2761	0.4279	0.5930	2.459
TrajCom	0.6372	0.8169	0.8543	0.680

方法	Geolife（北京）-骑车			
	HR@1	**HR@10**	**HR@20**	$\epsilon(km)$
NL	0.6297	0.7802	0.8597	0.708
MFL	0.2020	0.2736	0.3012	2.960
Cluster	0.6941	0.8308	0.8582	0.671
ST-RNN	0.6423	0.7382	0.8518	1.703
SERM	0.5608	0.6529	0.7896	1.803
BPR	0.2542	0.3960	0.5873	2.937
CNF	0.3073	0.4569	0.5981	2.708
RankGeoFM	0.3189	0.4747	0.6239	2.402
TrajCom	**0.7140**	**0.8553**	**0.8961**	**0.648**

方法	Geolife（北京）-汽车			
	HR@1	**HR@10**	**HR@20**	$\epsilon(km)$
NL	0.4231	0.6211	0.7200	1.364
MFL	0.1839	0.2317	0.2875	2.985
Cluster	0.6236	0.7731	0.8129	0.937
ST-RNN	0.5928	0.6423	0.7129	1.916
SERM	0.5133	0.5942	0.6537	2.035
BPR	0.2197	0.3518	0.5140	2.649
CNF	0.2394	0.3760	0.5637	2.478
RankGeoFM	0.2732	0.4180	0.6074	2.304
TrajCom	**0.6705**	**0.7931**	**0.8511**	**0.893**

表 3.4 和表 3.5 统计了不同数据集下的实验结果，下面将按照数据集不同分别分析实验结果。

对于 Check-in 类型的数据集，如表 3.4 所示，TrajCom 的补全准确率高于其他对比的方法。以 Foursquare (US)数据集为例，TrajCom 在HR@1上可以取得将近 25%左右的提升。所有对比方法中，TrajCom 与 Naïve 方法如 NL 和 NFL 在 HR@1 上的差距最大。这是因为目标的活动是复杂的，影响目标位置的因素有很多， Naïve 方法没有考虑这些因素，因此难以得好的效果。对比聚类的方法，其

结果比 Naïve 方法好一些，但准确率仍比 TrajCom 差。基于位置预测的方法（ST-RNN、SERM）比 POI 推荐的方法（BPR、CNF、RankGeoFM）效果好但也都次与 TrajCom。这是因为基于预测的方法考虑到了轨迹的序列信息，这使得其准确率高于位置推荐的方法。与基于预测的方法相比，TrajCom 同样使用 RNN 建模轨迹序列特性，但补全准确率有较大的提升。这是因为 TrajCom 基于 tGRU 建模了时间间隔信息，并通过双向 RNN 建模了目标在给定时刻前后的移动模式。另外，所有方法在 GeoTweets 数据集上的效果都高于 Foursquare 数据集。这是因为 Foursquare 上的位置分布更加的稀疏且不均匀，造成其补全难度大于其他数据集。

对于 Geolife 数据集，从表 3.5 可以看出，TrajCom 在大多数情况下都能取得优于对比方法的效果。所有对比方法中效果最好的是基于聚类的方法。这是因为人们的移动模式受到路网的限制，而基于聚类的方法可以捕捉这类特点。由于 TrajCom 可以联合建模多种影响因素，因此 TrajCom 可以在自行车（Bike）和汽车（Car）轨迹上取得最好的效果。但在行人（Walk）数据集上 TrajCom 与 NL 方法效果相当，这是由于人在路上行走的速度较慢，NL 方法可以很容易地发现正确的位置。但也因为如此，在速度更快的自行车和汽车轨迹数据上，NL 方法的精确度会急速下降。

3.3.2.2 变种实验结果

表 3. 6 方法的有效性验证

Table 3. 6 Validation of model effectiveness

方法	HR@1	HR@10	HR@20
TrajCom-vanilla	0.0710	0.1601	0.2321
TrajCom-pwAtten	0.0751	0.1621	0.2539
TrajCom-tGRU	0.0813	0.1869	0.2980
TrajCom	0.0880	0.2131	0.3169

为了验证 TrajCom 中 tGRU 单元和多头注意力机制单元的有效性，本实验中的对比方法中分别将 TrajCom 中的相关部分替换为标准的 GRU 单元和标准注意力机制，得到两个变种方法 TrajCom-tGRU 和 TrajCom-pwAtten。有效性实验的结果如表 3.6 所示，可以发现 tGRU(TrajCom − tGRU)和流行度权重注意力机制(TrajCom − pwAtten)都对于 TrajCom 都有积极的影响，可以提升轨迹补全准确

率。

3.3.2.3　可解释性分析

本实验通过两个实例来展示 TrajCom 的可解释性。本实验分别在 Foursquare（US）数据集和 Geolife 数据集上随机选择了 200 多个轨迹样本进行补全，并在地图上展示了补全结果及其相关轨迹。结果如图 3.8 所示。

图 3.8 可解释性示意图

Figure 3. 8 Explanation of TrajCom

第一条轨迹来自 Foursquare（US）数据集。可以观察到其相关轨迹与缺失轨迹具有相似的移动模式。此外，最相关的位置记录具有类似的语义信息，例如 Bar 和 Asian Food 都为餐馆。第二条轨迹来自 Geolife 数据集的轨迹。基于其 Top-5 的相关位置和概率，可以得到以下两点结论：第一，这些相关轨迹具有很高的语义相似性；其二，这些记录的时间间隔几乎与目标轨迹相同。这意味着在使用了 tGRU 模块之后，TrajCom 可以有效地对上下文信息和时间间隔信息进行建模。

3.4 小结

本文提出了一种基于神经过滤和编码的方法 TrajCom 用于补全稀疏和轨迹。TrajCom 的创新性在于两个方面：其一，设计了一个上下文指导的协同过滤模型，它能够从历史轨迹中筛选出与目标轨迹相关的轨迹。其二，设计了一个深度轨迹编码器，该编码器利用时间感知的神经网络单元 tGRU 为基础学习轨迹表示向量。最后，TrajCom 利用目标轨迹和相关轨迹中记录的表示向量推断缺失的位置。实验证明了 TrajCom 在补全稀疏轨迹上的效果优于目前已有的轨迹补全方法。

第 4 章　基于深度度量学习的轨迹相似度计算方法研究

本章针对轨迹相似度计算复杂度高的问题，提出了基于深度度量学习的轨迹相似度计算方法 NeuTraj。该方法可以将轨迹相似度计算的复杂度降低到线性。NeuTraj 基于深度度量学习将计算轨迹序列之间的相似度转化为计算轨迹表示向量间的相似度。本章首先介绍了轨迹相似度计算复杂度高的原因并分析了利用深度度量学习加速相似度计算的可行性。然后介绍了空间注意力记忆机制 SAM。该机制被用于解决传统 RNN 模型不能捕捉轨迹空间邻近特性的问题。最后介绍了 NeuTraj 的优化过程，提出了加权排序损失并将其用于模型训练。实验利用两个真实的轨迹数据集验证了 NeuTraj 的有效性。

本章的组织如下：4.1 节详细阐述了研究问题的内容和挑战。4.2 节详细介绍了基于深度度量学习的轨迹相似度计算方法的主要内容，其中，4.2.1 节定义了轨迹相似度计算加速问题，并概述了 NeuTraj 的流程；4.2.2 节详细介绍了 SAM 模块及基于 SAM 提出的循环神经网络单元；4.2.3 节详细介绍了加权排序损失和 NeuTraj 的优化过程。4.3 节通过准确率和效率两方面的实验验证了 NeuTraj 算法的有效性和可行性。4.4 节对本章进行了小结。

4.1　引言

轨迹相似度定义了轨迹与轨迹之间的基本关系，是轨迹分析和检索的基础。目前，已经有许多研究工作提出轨迹相似度计算方法，包括 Fre´chet（Alt 等.，1995）、ERP（Chen 和 Ng，2004）、EDR（Yin 和 Wolfson，2004）、DTW（Yi 等.，1998）、LCSS（Vlachos 等.，2002）、Hausdorff（Atev 等.，2010）等等。这些相似度计算方法在异常检测和轨迹聚类等任务中发挥了关键的作用。然而，随着轨迹数量的增加，现有相似度计算方法复杂度高成为了大规模轨迹分析的瓶颈。为了计算两条轨迹之间的相似度，现有方法采用对齐-汇总（alignment-summarization）的框架，首先对齐两条轨迹中的所有记录，然后汇总所有记录对之间的信息计算相似度。这样的过程使得现有度量方法的计算复

杂度至少为平方级，很难应用于大规模轨迹数据集。例如，在一台高配置服务器上仅计算 8000 条 GPS 轨迹两两之间的 Hausdorff 距离就需要花费超过 6.5 小时的时间。随着轨迹数量的增加所需的计算量也会急剧提升。因此，研究如何加速轨迹相似度计算是轨迹数据挖掘的一项重要任务。

加速轨迹相似度计算的难点有两方面。一方面在于轨迹相似度计算本身的复杂性。输入的轨迹具有不同的长度，不能从头至尾进行对齐。因此，大多相似性度量方法需要计算轨迹记录之间的最佳匹配，仅这一过程的复杂度就为平方级。另一个难点在于轨迹相似性度量的定义多种多样。常用的轨迹相似性度量在定义和计算机制方面有着很大不同，难以设计出适用于多种相似性度量的通用加速方法。

针对上述难点，已有研究人员提出了加速轨迹相似度计算的方法，这些方法可分为两类。第一类是基于轨迹索引的方法，这类方法利用轨迹的空间结构对轨迹数据库构建索引从而加速轨迹检索的过程。然而，这些方法专门针对轨迹 Top-K 相似性检索任务，通过为给定的轨迹数据库设计索引和剪枝策略减少检索中的涉及计算的轨迹数量，却无法降低轨迹对相似度计算的复杂度。因此，该类方法不能应用于如轨迹聚类和异常检测等需要计算轨迹对相似度的任务中。第二类是基于近似计算的方法。针对不同的相似性度量，研究人员提出了不同的近似算法，用于降低轨迹相似度计算的复杂度，如利用局部敏感哈希（LSH）计算轨迹的 Fre´chet 和 DTW 相似度。然而，这类方法仅针对某一种特定的相似性度量，并不适用于其他相似性度量方法。

为了解决上述问题，本文提出了基于深度度量学习的轨迹相似度计算方法 NeuTraj。该方法是一种近似算法，可以将轨迹相似度计算复杂度降低到线性，且可以适用于不同相似性度量。NeuTraj 首先从轨迹数据库中采样若干种子轨迹，形成种子轨迹池，并利用种子与种子之间相似度的值作为指导，训练轨迹编码模型。总的来说，NeuTraj 利用神经网络学习轨迹记录序列到轨迹表示向量的映射函数，使得表示向量之间的相似度接近于原有相似度。NeuTraj 的训练过程包括数据准备、轨迹编码和度量学习三个阶段。

在数据准备阶段，NeuTraj 从轨迹数据库中采样种子轨迹，并计算所有种子

轨迹对之间的相似度。然后，以预计算的种子相似度作为指导，设计损失函数以优化轨迹编码网络来拟合种子轨迹表示向量间的相似度。

在轨迹编码阶段，NeuTraj 利用 RNN 将变长的轨迹序列编码为固定长度的表示向量。然而传统的 RNN 及其变体（GRU，LSTM）只能建模轨迹内部前后部分间的序列关系，但不能捕捉多条轨迹之间的空间邻近特性。由于轨迹相似性度量的定义大部分基于轨迹空间邻近性，导致空间邻近特性对于计算轨迹相似度非常重要。为了解决这一问题，本文提出了空间注意力记忆机制（SAM），该机制通过增加一个外部记忆张量（External Memory Tensor）来存储已处理过轨迹的信息，并利用注意力机制来控制对张量信息的读取和写入，从而捕获训练轨迹之间的空间邻近特性，生成适用于相似度计算的轨迹表示相邻。

在度量学习阶段，NeuTraj 利用种子轨迹之间的相似度优化模型参数。该阶段的难点在于训练效率和准确率之间的权衡。一方面，训练 NeuTraj 需要遍历所有的种子轨迹对。另一方面，将所有种子轨迹对作为训练样本会导致很大的计算和时间开销。例如训练过程仅采样 500 条轨迹作为种子轨迹，种子轨迹对的数量就会达到 12.5 万对。为了解决这一难题，本文提出了一种距离加权排序损失（Weighted Ranking Loss），用于从所有的种子轨迹对中采样更具有区分度的轨迹对作为训练样本。对比已有相似度计算近似算法，NeuTraj 在训练数据规模很小的情况下，也可以在 Top-10 轨迹相似性检索任务上达到 80%以上的准确率和 3 倍以上的计算加速。该结果优于已有加速轨迹相似度计算的近似算法。

4.2　基于深度度量学习的轨迹相似度计算方法

本节详细介绍了基于深度度量学习的轨迹相似性度量方法 NeuTraj。首先定义了轨迹相似度计算问题并概述了 NeuTraj 的主要流程;然后详细阐述了 NeuTraj 各个步骤的计算过程和模型结构，最后分析了基于 NeuTraj 计算轨迹相似度的复杂度。

4.2.1　问题定义及方法概述

本章首先定义了加速轨迹相似度计算的问题，然后简要介绍了本文提出的基

于深度度量学习的轨迹相似度计算方法 NeuTraj。

表 4.1 符号及标记说明

Table 4.1 Notations of TrajCom

符号	含义解释
$\mathcal{T}$，$\mathcal{S}$	总轨迹数据库和由 N 条种子轨迹组成的种子数据库
T	轨迹数据库中的一条轨迹，由一系列坐标点组成
$\mathbf{D}, \mathbf{S}$	种子数据库两两之间的距离矩阵和相似度矩阵
$\mathbf{M}$	储存空间记忆信息的空间记忆张量
$\mathbf{E}_i, \mathbf{E}_j$	由 NeuTraj 生成的 d 维的轨迹表示向量
X_t	在 t 时刻 NeuTraj 的输入，含有坐标信息和空间网格信息
$\mathbf{W}, \mathbf{U}, \mathbf{b}$	SAM 单元中的线性权重和偏差
$\boldsymbol{f_t}, \boldsymbol{o_t}, \boldsymbol{i_t}$	SAM 增强的 LSTM 中的遗忘、输出和输入门
$\boldsymbol{s_t}$	SAM 单元中的创新空间门，控制空间信息的输入输出
$c_t, \hat{c_t}$	SAM 单元中的单元状态和中间状态
h_t, h_{t-1}	SAM 单元在时刻 t 和 t-1 上的隐含状态
$\mathbf{G}_t$	时刻 t，X_t作为输入时的空间信息矩阵
$\mathbf{A}$	时刻 t，X_t作为输入时的注意力权重
$\mathbf{c_t^{cat}}, \mathbf{c_t^{his}}$	SAM 单元中的中间拼接状态和最终的历史状态
T_a	为了训练 NeuTraj，从轨迹数据库中采样的种子锚轨迹
$\mathcal{T}_a^s, \mathcal{T}_a^d$	关于T_a的 N 条相似轨迹和 N 条不相似的轨迹
$\mathbf{S_a^s}, \mathbf{S_a^d}$	精确的关于相似和不相似轨迹对的相似度值
$\mathbf{\hat{S}_a^s}, \mathbf{\hat{S}_a^d}$	由 NeuTraj 计算得到的轨迹对相似度的值

4.2.1.1 问题定义

给定轨迹数据库$\mathcal{T}$和轨迹相似性度量$f(\cdot,\cdot)$，每个轨迹$T \in \mathcal{T}$是运动目标的位置记录序列。不失一般性的情况下，本文考虑二维轨迹相似度计算问题。也就是说，轨迹$T = [X_1^c, \dots, X_t^c, \dots]$的每条记录$X_t^c = (x_t, y_t)$包含二维的经纬度属性。对于轨迹$T_i, T_j \in \mathcal{T}$，$f(T_i, T_j)$是轨迹相似性度量，可以用于计算$T_i$和$T_j$之间的相似度。$f(\cdot,\cdot)$可能是 DTW，Hausdorff，Fre´chet 或其他任何轨迹相似性度量。

大多数情况下，轨迹相似性度量$f(\cdot,\cdot)$的计算复杂度至少是二次的。因此，需要提出近似算法$g(\cdot,\cdot)$来降低$f(\cdot,\cdot)$的复杂度。具体来说，本章的研究问题是：如何学习与$f(\cdot,\cdot)$近似的函数$g(\cdot,\cdot)$，使得$g(T_i, T_j)$ 的计算复杂度低于$f(T_i, T_j)$，

且$|f(T_i,T_j)-g(T_i,T_j)|$最小。

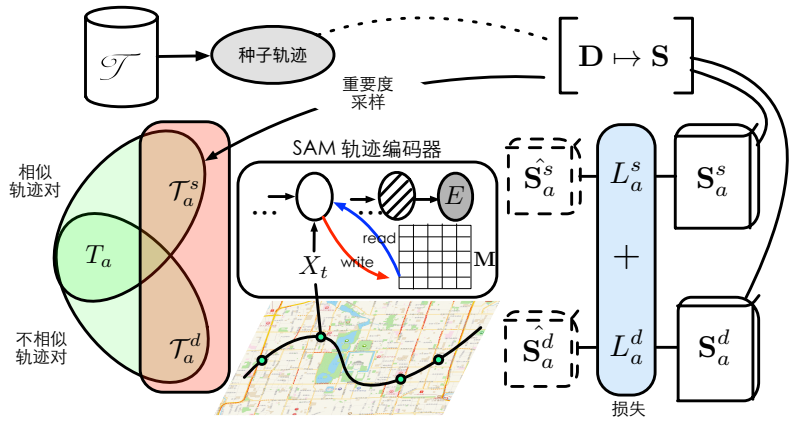

图 4. 1 NeuTraj 示意图

Figure 4. 1 Architecture of NeuTraj

4.2.1.2　方法概述

NeuTraj 是一个基于深度度量学习的方法。它首先从$\mathcal{T}$中随机采样 N 条轨迹作为种子轨迹，形成种子轨迹池$\mathcal{S}$，并计算$\mathcal{S}$中任意两对种子之间的相似度，形成 N×N 的距离矩阵**D**。然后，通过归一化等操作将**D**变换为相似度矩阵**S**。以**S**为指导，NeuTraj 学习一个轨迹编码网络。该网络将任意长度的轨迹编码为固定长度的向量。具体来说，对于任意两条输入轨迹T_i和T_j $(i,j\in[1,\ldots,N])$，NeuTraj 分别将它们投影到两个d维表示向量E_i和E_j。这一编码过程需要保持轨迹的相似性，即$f(T_i,T_j)\approx g(T_i,T_j)$，其中$g(\cdot,\cdot)$是表示向量空间中$E_i$和$E_j$之间的相似度。图 4.1 展示了 NeuTraj 的架构。可以看出 NeuTraj 由两个主要模块组成：空间注意力记忆（SAM）轨迹编码器和种子指导下的深度度量学习方法。

SAM 轨迹编码器。NeuTraj 利用 RNN 对轨迹进行建模，并将 RNN 的最后隐含状态作为轨迹的表示向量。然而，传统 RNN 及其变体（GRU，LSTM）仅能独立地对每个序列前后之间的关系进行建模，不能建模序列与序列之间的关系。由于轨迹相似性度量大多基于空间邻近特性计算轨迹相似度，因此轨迹之间的相关性对轨迹相似度计算来说至关重要。本文在 NeuTraj 中设计了空间注意力记忆（SAM）模块。它采用空间记忆张量来存储已经处理过的轨迹的空间信息，并基于注意力机制控制对张量的读取和写入，使得 RNN 可以按编码和检

索已经处理的轨迹信息。

种子指导下的深度度量学习。基于 SAM 轨迹编码器，NeuTraj 利用种子轨迹的相似度，指导编码器参数的优化。优化 NeuTraj 时，每次以一对种子轨迹作为输入，首先分别对轨迹进行编码得到轨迹表示向量；然后优化编码网络参数，使得轨迹表示向量之间的相似度和矩阵**S**中的相似度近似。现有度量学习方法采用随机采样来产生训练轨迹对，这意味着所有轨迹对都是等权重的。这种采样方式会造成模型收敛变慢的问题。本文提出了加权排序损失将模型的优化重点聚焦在更有区分力的轨迹对上。对于一条种子轨迹来说，NeuTraj 假设与其相似度最大或最小的轨迹比中间部分轨迹更有区分力。因此，NeuTraj 首先利用加权采样得到最相似和最不相似的轨迹来构成训练轨迹对，然后利用排序损失学习网络的参数。

4.2.2 空间注意力记忆机制

本节介绍了空间注意力记忆 SAM（Spatial Attention Memory）机制，该机制可以增强传统 RNN 模型，使其具备建模序列空间邻近特性的能力。下面首先介绍 SAM 的结构及空间记忆张量；然后介绍 SAM 增强的 LSTM；最后，详细介绍 SAM 增强 LSTM 中对于空间记忆张量的读写操作。

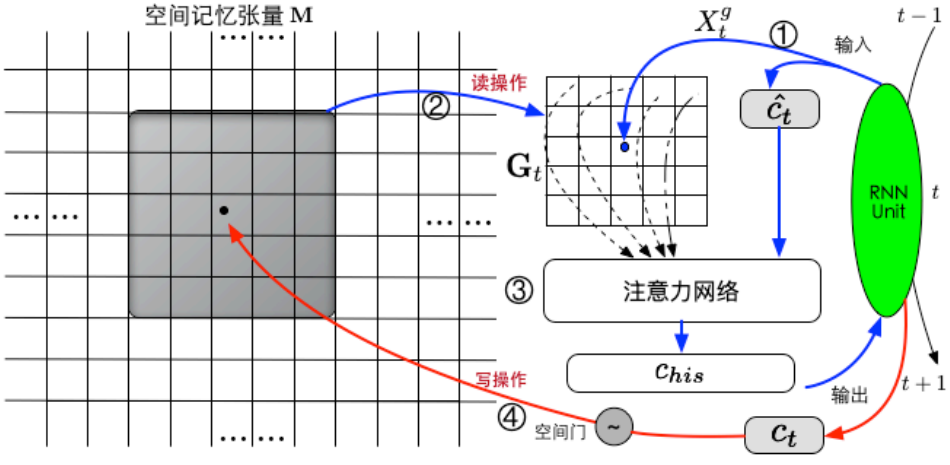

图 4.2 空间注意力记忆（SAM）机制示意图

Figure 4. 2 Illustration of the proposed spatial attention memory(SAM)

4.2.2.1 基于网格的空间记忆张量

SAM 机制首先将空间划分为等大小的空间网格单元，然后定义了一个基于

空间网格的空间记忆张量（Spatial Memory Tensor）。这样，任何轨迹$T = [X_1^c, \ldots, X_t^c, \ldots]$都有与之对应到网格序列$T_g = [X_1^g, \ldots, X_t^g, \ldots]$，其中$X_t^g = (x_t^g, y_t^g)$是指定对应网格在横纵第$x_t^g$和$y_t^g$个位置。图 4.2 展示了 SAM 的结构。如图所示，其核心部分是空间记忆张量$\mathbf{M}$。$\mathbf{M}$存储空间中所有网格单元的向量表示，能够编码和检索已经处理过的轨迹的空间信息。具体来说，假设整个空间被划分为$\mathrm{P} \times \mathrm{Q}$个网格单元，则张量$\mathbf{M}$的维度为$\mathrm{R}^{\mathrm{P}\times\mathrm{Q}\times\mathrm{d}}$，其中 d 是循环单元隐含状态的维度。$\mathbf{M}$中的每个切片$(\mathrm{p}, \mathrm{q}, :)$是存储空间网格 (p, q)的向量表示。所有网格单元表示向量在训练之前都被初始化为 0。随着 NeuTraj 处理轨迹，$\mathbf{M}$将随之更新并对处理轨迹中的空间信息进行编码。

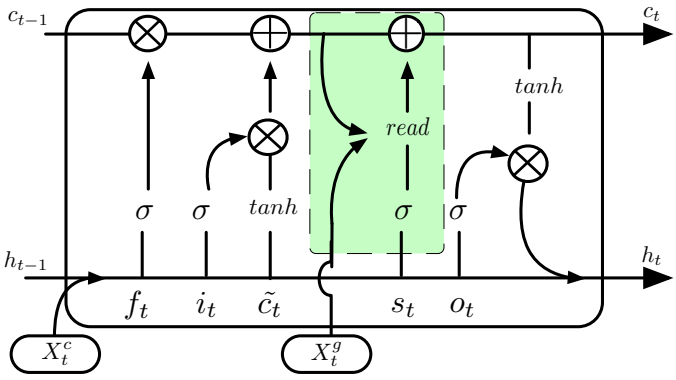

图 4. 3 SAM 增强的 LSTM

Figure 4. 3 SAM-augmented LSTM

4.2.2.2　SAM 增强的 LSTM

图 4.3 展示了 SAM 增强的 LSTM 单元架构，图中绿色部分是新提出的 SAM 模块。如图所示，在每个循环步骤t中，该单元以$\mathrm{X_t} = (X_t^c, X_t^g)$和前一步骤的隐含状态$\boldsymbol{h_{t-1}}$作为输入，输出$\boldsymbol{h_t}$并进行下一个循环。与 LSTM 操作相似，SAM 增强的 LSTM 使用门控机制（gate mechanism）来控制对单元状态$\mathrm{c_t}$的操作。计算过程进行：

$$(\boldsymbol{f_t}, \boldsymbol{i_t}, \boldsymbol{s_t}, \boldsymbol{o_t})^T = \sigma\left(\boldsymbol{W_g} \cdot X_t^c + \boldsymbol{U_g} \cdot \boldsymbol{h_{t-1}} + \boldsymbol{b_g}\right) \tag{4.1}$$

$$\mathbf{c_t^{\sim}} = \tanh(\boldsymbol{W_c} \cdot X_t^c + \boldsymbol{U_c} \cdot \boldsymbol{h_{t-1}} + \boldsymbol{b_c}) \tag{4.2}$$

$$\boldsymbol{c_t^{\wedge}} = \boldsymbol{f_t} \cdot \boldsymbol{c_{t-1}} + \boldsymbol{i_t} \cdot \boldsymbol{c_t^{\sim}} \tag{4.3}$$

$$\boldsymbol{c_t} = \boldsymbol{c_t^{\wedge}} + \boldsymbol{s_t} \cdot read(\boldsymbol{c_t^{\wedge}}, X_t^g, \boldsymbol{M}) \tag{4.4}$$

$$\text{write}\left(\boldsymbol{c_t}, \boldsymbol{s_t}, X_t^g, \boldsymbol{M}\right) \tag{4.5}$$

$$\boldsymbol{h_t} = \boldsymbol{o_t} \cdot \tanh\left(\boldsymbol{c_t}\right) \tag{4.6}$$

在其中$\mathbf{W_g} \in R^{4d\times 2}, \boldsymbol{U_g} \in R^{4d\times d}, \boldsymbol{W_c} \in R^{d\times 2}, \boldsymbol{U_c} \in R^{d\times d}$，d 是隐含状态的大小。所有的门向量$(\boldsymbol{f_t}, \boldsymbol{i_t}, \boldsymbol{s_t}, \boldsymbol{o_t})$、单元状态$(\boldsymbol{c_t^{\sim}}, \boldsymbol{c_t^{\wedge}}, \boldsymbol{c_t})$和隐含状态$(\boldsymbol{h_t}, \boldsymbol{h_{t-1}})$有着相同的维度：$\mathrm{R}^{\mathrm{d}\times 1}$。

为了获得隐含状态$\boldsymbol{h_t}$，该单元执行以下操作步骤：

- 门操作。通过公式 4.1，该单元使用 sigmoid 函数 σ 对输入坐标X_t^c和前一个隐含状态$\boldsymbol{h_{t-1}}$的线性组合进行非线性变换，得到四个门：遗忘门$\boldsymbol{f_t}$，输入门$\boldsymbol{i_t}$，空间门$\boldsymbol{s_t}$和输出门$\boldsymbol{o_t}$；
- 单元状态操作。根据公式 4.2~4.4，该单元根据$\boldsymbol{f_t}$，$\boldsymbol{i_t}$，$\boldsymbol{s_t}$和输入(X_t^c, X_t^g)产生当前循环步骤的单元状态$\boldsymbol{c_t}$。
- 隐含状态操作。通过公式 4.6，该单元产生$\boldsymbol{h_t}$，并将其输出到下一个循环步骤。

SAM 增强 LSTM 的关键操作在于公式 4.4 和 4.5。其中公式 4.4 使用读取操作来获得$\mathbf{M}$中与当前输入相关的历史轨迹信息增加$\boldsymbol{c_t^{\wedge}}$。此外，该单元通过公式 4.5 的写入操作更新记忆张量$\mathbf{M}$。读取和写入操作的细节将在下节中介绍。

4.2.2.3 空间记忆张量的读取和写入操作

SAM 使用注意力机制控制的空间记忆张量$\mathbf{M}$的读写操作：在读取操作中，SAM 从$\mathbf{M}$中搜索与输入网格X_t^g邻近空间网格的表示向量，并用该信息增强轨迹当前输入；在写入操作中，SAM 将当前轨迹输入的信息更新到X_t^g对应的网格表示向量。

读取操作：在每一个循环步骤中，SAM 从空间记忆张量中检索信息并使用该信息来增强对当前轨迹输入的编码。如图 4.2 所示，读取操作在每个循环步骤中会以两部分信息作为输入：网格单元输入X_t^g；中间单元状态$\boldsymbol{c_t^{\wedge}}$。利用这两个输入，该操作输出一个含有部分空间记忆张量$\mathbf{M}$信息的状态向量$\boldsymbol{c_t^{his}}$，这个向量通过共享同一个空间记忆张量$\mathbf{M}$来建模轨迹之间的空间邻近关系。

如图 4.2 所示，读取操作首先提取在空间中接近$X_t^g = (x_t^g, y_t^g)$的网格单

元。具体来说，该操作利用宽度w扫描空间网格，获取(x_t^g, y_t^g)周围的单元格：$scan(x_t^g) = [x_t^g - w, x_t^g + w]; scan(y_t^g) = [y_t^g - w, y_t^g + w]$。经过该步骤，相关的网格表示向量从$\mathbf{M}$中取出并存入到维度为$\mathrm{R}^{(2w+1)^2 \times d}$的张量$\mathbf{G_t}$中。然后，读取操作利用注意力机制，以单元状态$\boldsymbol{c_t^{\wedge}}$查询，以$\mathbf{G_t}$作为内容生成 d 维状态向量$\boldsymbol{c_t^{his}}$。 注意力机制运算如下：

$$\boldsymbol{A} = \mathrm{softmax}(\mathbf{G_t} \cdot \mathbf{c}_t);$$

$$\mathbf{mix} = \mathbf{G_t} \cdot \boldsymbol{A};$$

$$\boldsymbol{c_t^{cat}} = [\mathbf{c}_t, \boldsymbol{mix}];$$

$$\boldsymbol{c_t^{his}} = \tanh\left(\boldsymbol{W_{his}} \cdot \boldsymbol{c_t^{cat}} + \boldsymbol{b_{his}}\right) \quad (4.7)$$

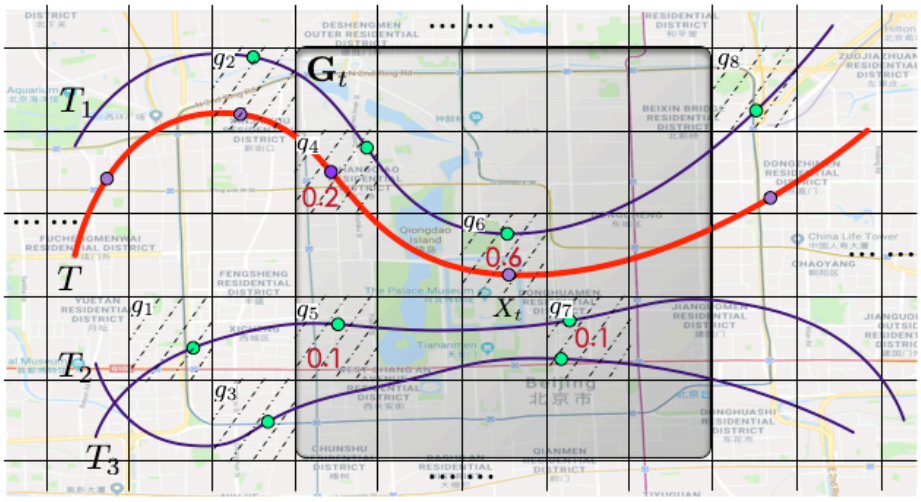

图 4.4 空间读写操作示例

Figure 4. 4 Example to illustrate spatial reads and writes

矩阵$\boldsymbol{W_{his}}$和$\boldsymbol{b_{his}}$是注意力网络中的参数。$\boldsymbol{A} \in \mathrm{R}^{(2w+1)^2 \times 1}$是注意力权重，它反映了当前状态$\mathrm{c_t^{\wedge}}$与已处理轨迹信息张量$\mathbf{G_t}$中网格的相似度。例如，图 4.4 中$T_1$~$T_3$是历史处理过的轨迹，网格单元$g_1$~$g_1$是含有空间信息的非零网格单元。以$X_t^g$作为输入，经过注意力网络后，网格单元$g_4$~$g_7$的注意力权重为 0.2，0.1，0.6，和 0.1，表示当前处理轨迹与 T1 更相似。$\mathrm{mix} \in \mathrm{R}^{d \times 1}$是$\mathrm{G_t}$中表示向量的权重之和。最终，该操作经过全连接层将$\boldsymbol{c_t^{cat}}$转换为 d 为向量并产生包含历史信息的空间单元状态$\boldsymbol{c_t^{his}}$。得到$\boldsymbol{c_t^{his}}$后，经 SAM 增强的 LSTM 将它与$\boldsymbol{c_t^{\wedge}}$结合，然后通过公式 4.7 得到最终的单元状态$\boldsymbol{c_t}$。

写入操作：在处理过程中，当前轨迹输入应更新$\mathbf{M}$中对应网格单元的表示

向量。在每一循环步骤中，写入操作基于$\boldsymbol{s_t}$对$\mathbf{M}$中的相应条目进行稀疏更新操作：

$$\mathbf{M}\left(\boldsymbol{X_g}\right)_{\mathbf{new}} = \sigma(\boldsymbol{s_t}) \cdot \boldsymbol{c_t} + \left(1 - \sigma(\boldsymbol{s_t})\right) \cdot \mathbf{M}\left(\boldsymbol{X_g}\right)_{\boldsymbol{old}} \quad (4.8)$$

由上述公式可看出，网格单元的表示向量是处理过的轨迹状态信息的加权平均。需要注意的是，SAM 增强的 LSTM 对读取操作和写入操作使用相同的空间门$\boldsymbol{s_t}$。这是因为$\boldsymbol{s_t}$不仅反映了$\boldsymbol{c_t^{his}}$对当前输入的置信度，还反映了当前输入中有多少信息适合更新到$\mathbf{M}(X_g)$。另外，共享$\boldsymbol{s_t}$可以减少了 NeuTraj 需要优化的参数数量。

4.2.3 种子引导的深度度量学习

在本节首先概述 NeuTraj 深度度量学习过程，然后提出加权采样和参数优化方法来学习 NeuTraj 参数。

4.2.3.1 NeuTraj 深度度量学习过程

图 4.1 展示了 NeuTraj 方法，该方法使用 RNN 将变长轨迹映射到一个 d 维向量，并使向量间相似度近似等于轨迹相似度。如图所示，NeuTraj 的核心是 SAM 增强的 RNN 编码器。对于输入轨迹，该编码器将 RNN 的最终隐含状态作为轨迹表示。对于任何两个输入轨迹T_i和T_j $(i, j \in [1, \ldots, N])$，RNN 编码器能够将它们投影到两个 d 维向量$\boldsymbol{E_i}$和$\boldsymbol{E_j}$。优化的目标是学习编码器参数，使得$\boldsymbol{E_i}$和$\boldsymbol{E_j}$之间的相似度接近给定相似性度量计算结果$\boldsymbol{f}(\boldsymbol{T_i}, \boldsymbol{T_j})$。然而，直接拟合种子相似度矩阵$\mathbf{S}$中的所有种子轨迹对会导致模型收敛变慢并存在过拟合的风险。因此需要选择合适训练样本用于优化 NeuTraj。本文将 NeuTraj 的损失函数定义为所有训练样本对 MSE（Mean Square Error）的加权之和：

$$\mathbf{min} \sum_{k=1}^{K} \boldsymbol{w_k} \cdot \left(\boldsymbol{f}(\boldsymbol{T_i}, \boldsymbol{T_j}) - \boldsymbol{g}(\boldsymbol{T_i}, \boldsymbol{T_j})\right)^2 \quad (4.9)$$

其中 K 是训练样本对的数量，w_k是对 k 的权重。

为此选取合适的训练样本对，本文设计了加权采样过程。具体来说，NeuTraj 首先顺序地将种子池中的轨迹作为锚轨迹，对于一条锚定轨迹T_a，从种子轨迹中采样一组相似轨迹$\mathcal{T}_{\mathrm{a}}^{\mathrm{s}}$以及一组不相似轨迹$\mathcal{T}_{\mathrm{a}}^{\mathrm{d}}$，以形成关于$T_a$相似和不

相似的训练样本对。然后利用加权排序损失设计目标函数，鼓励 NeuTraj 以更有区分度的轨迹样本对的优化 NeuTraj 参数，用以拟合轨迹相似度矩阵$\mathbf{S}$。

4.2.3.2　加权排序损失和参数优化

NeuTraj 与经典度量学习方法 Siamese 网络相关（Bromley 等.,1993）。Siamese 网络使用随机采样的方法生成训练轨迹对，这意味着所有训练轨迹对在损失函数中有着相同的权重。该假设不适用于轨迹相似度计算任务，因为它忽略了轨迹之间的空间邻近性，会导致模型收敛变慢和精度降低。

为了解决上述问题，本文提出了加权排序损失。给定一个锚定轨迹，NeuTraj 需要重点关注与其最相似的轨迹和最不相似的轨迹，因为它们比其他的轨迹更具有区分度。NeuTraj 首先需要将原始距离矩阵$\mathbf{D}$转换为归一化的相似矩阵$\mathbf{S}$，过程如下所示：

$$\mathbf{S_{i,j}} = \mathbf{exp}\left(-\boldsymbol{\alpha} \cdot \mathbf{D_{i,j}}\right) / \sum_{\boldsymbol{n=1}}^{\boldsymbol{N}} \mathbf{exp}\left(-\boldsymbol{\alpha} \cdot \mathbf{D_{i,n}}\right) \tag{4.10}$$

其中 α 是控制相似度数值分布的参数。原始距离的分布通常服从幂律分布，并且距离的大小可以跨越很大范围。该转换本质上是一种平滑操作，它将相似度值压缩至范围[0,1]。

受 Manmatha 等（2017）的启发，加权采样的工作原理如下。首先将种子池中的 N 条轨迹依次作为锚定轨迹进行操作。对于一个锚轨迹 T_a，NeuTraj 从相似度矩阵$\mathbf{S}$中取相应的行，作为重要度的权重$\mathbf{I_a}$。然后，利用 $\mathbf{I_a}$ 作为权重采样 n 个不同的轨迹作为相似样本：$\mathcal{T}_{\mathrm{a}}^{s} = \{T_1^s, \dots, T_n^s\}$。相反的，使用$\mathbf{1 - I_a}$作为权重，采样 n 个不相似的轨迹样本$\mathcal{T}_{\mathrm{a}}^{d} = \{T_1^d, \dots, T_n^d\}$。最后，按照相似度降低的顺序对 $\mathcal{T}_{\mathrm{a}}^{s}$ 进行排序，并按相似度增加的顺序对 $\mathcal{T}_{\mathrm{a}}^{d}$ 进行排序。经过上述过程，我们获得了 2n 对与T_a相关的训练样本对。

根据训练样本对，NeuTraj 利用轨迹编码器生成轨迹表示向量，并按照如下公式为锚轨迹T_a定义了相似和不相似轨迹对的相似度：

$$\begin{aligned} \mathrm{S}_{\mathrm{a}}^{\hat{}\mathrm{s}} &= S^{\hat{}}(T_a, \mathcal{T}_a^{\mathrm{s}}) = [g(T_a, T_1^s), \dots g(T_a, T_n^s)]; \\ \mathrm{S}_{\mathrm{a}}^{\hat{}\mathrm{d}} &= S^{\hat{}}\left(T_a, \mathcal{T}_a^{\mathrm{d}}\right) = \left[g\left(T_a, T_1^d\right), \dots g(T_a, T_n^d)\right] \end{aligned} \tag{4.11}$$

$g\left(\mathrm{T_i}, T_j\right) = \exp\left(-Euclidean(\boldsymbol{E_i}, \boldsymbol{E_j})\right)$ 用来计算两个轨迹表示向量之间的相似

度，$\boldsymbol{E}$ 是相应轨迹的表示向量。

NeuTraj 利用加权排序损失定义训练的损失函数。受 Cao 等（2007），McFee 和 Lanckriet（2010）启发，给定由 n 个轨迹对组成的排好序的列表，我们将排序权重依次置为$r = (1, \frac{1}{2}, \ldots, \frac{1}{1}, \ldots, \frac{1}{\mathrm{n}})$，并用 $\sum_{l=1}^{n} r_l$ 对权重进行归一化。对于 n 个相似的轨迹对，它们的权重随着排序顺序而减小，即$\mathcal{T}_{\mathrm{a}}^{s}$中最相似的轨迹被认为是最重要的。因此将$T_a$中的相似轨迹对的损失定义为：

$$\mathrm{L}_a^s = \sum_{l=1}^{n} r_l \cdot \left(g(T_a, T_l^s) - f(T_a, T_l^s)\right)^2 \tag{4.11}$$

其中$f(T_i, T_j)$是T_i, T_j的准确相似度。对于不相似的轨迹对，不需关注最不相似的样本对之间距离。因此对于不相似样本对，设计损失函数如下：

$$\mathrm{L}_{\mathrm{a}}^d = \sum_{l=1}^{n} r_l \cdot \left[\mathrm{ReLU}\left(g\left(T_a, T_l^d\right) - f\left(T_a, T_l^d\right)\right)\right]^2 \tag{4.12}$$

其中函数 $\mathrm{ReLU(x)} = \max(0, \mathrm{x})$ 。当 $g\left(T_a, T_l^d\right) - f\left(T_a, T_l^d\right) < 0$时$\mathrm{L}_{\mathrm{a}}^d = 0$，意味着不相似的样本在轨迹表示向量空间中与锚轨迹的距离足够远；当 $g\left(T_a, T_l^d\right) - f\left(T_a, T_l^d\right) > 0$时$\mathrm{L}_{\mathrm{a}}^d > 0$，应该调整轨迹表示向量，以扩大锚轨迹与不相似样本的在轨迹表示向量空间中的距离。最后，给定种子轨迹池$\mathcal{S}$，NeuTraj 的损失是所有 N 个种子上相似和不相似样本的总和。

$$\mathrm{L}_{\mathcal{S}} = \sum_{a \in [1,2,\ldots,N]} (\mathrm{L}_a^s + \mathrm{L}_a^d) \tag{4.13}$$

由于所有模块和损失函数都是可微分的，NeuTraj 中的所有参数都可以用端到端的方式进行学习。在训练过程使用反向传播时间（BPTT）算法更新参数，并使用 Adam 优化器进行参数优化。

4.2.3.3 复杂度分析

利用 NeuTraj 计算轨迹相似度的计算过程包括两部分：轨迹编码部分和距离计算部分。轨迹编码部分的计算开销与轨迹长度是线性关系。在一个循环步骤中，循环神经网络单元处理轨迹中记录的过程仅与当前记录有关，是一个固定的常数。因此，轨迹编码部分的计算复杂度是线性的。距离计算方面，对于一对轨迹，轨迹表示向量维度 d 是常数，表示空间中距离计算的复杂度也是常数。因此，NeuTraj 的总计算复杂度也是线性的。

对于轨迹数据库来说，轨迹编码部分仅需要计算一次。对于轨迹相似度检索来说，当进行新的检索轨迹输入时，仅需要对新轨迹进行编码，并基于轨迹表示向量搜索相似的轨迹。因此，计算开销与轨迹搜索空间大小成线性关系，这使得 NeuTraj 适用于大型数据集。

4.3　性能评估

为了评估 NeuTraj，本文进行两大类实验：Top-K 轨迹相似性检索及轨迹聚类，分别用于评估 NeuTraj 在轨迹检索和轨迹对相似度计算上的准确率和效率。本节首先介绍了实验设置，然后展示并分析了实验结果。

4.3.1　实验设置

实验设置部分主要包括数据集、实验方案、对比方法、评估及参数设置，下面分别介绍上述五个方面

4.3.1.1　实验数据

本实验基于两个公开的城市轨迹数据集：北京和波尔图。第一个数据集（Zheng 等.，2010），简称 Geolife，由 2007 年至 2010 年的 17621 个人类流动轨迹组成。第二个数据集（Matias 等.，2016）包含 2013 年至 2014 年超过 170 万的出租车轨迹。为了节制 M 的大小以节省内存空间，实验数据选择在城市中心区域的轨迹，并将该区域离散为 50 米×50 米的网格单元，然后删除少于 10 条记录的轨迹。经过上述预处理，获得 8203 条 Geolife 轨迹和 601071 条 Porto 轨迹。

4.3.1.2　实验方案

为了评估 NeuTraj 的准确性，本文研究了 Geolife 和 Porto 数据集上的 Top-K 相似性检索问题，并在 Fre´chet 距离、Hausdorff 距离、ERP 和 DTW 四种相似性度量下评估了 NeuTraj 的性能。其中前三个满足度量定义，即相似度是对称的并且满足三角不等式，可以直接利用度量学习近似。由于 DTW 不满足三角不等式，导致 DTW 距离不是标准的度量。DTW 的实验验证了 NeuTraj 在非标准度量上的近似性能。

轨迹 Top-K 相似性检索的准确结果是基于相似度定义得到的 Top-K 相似轨迹。对于 Geolife，本实验计算所有轨迹对的准确相似度，并随机选择 20%轨迹作为训练 NeuTraj 的种子轨迹。另外，10%的轨迹用于调整参数，70%用于测试。对于 Proto，由于轨迹数量巨大，直接计算所有轨迹对的精确相似度是不切实际的。我们随机选择一万条轨迹来计算轨迹对相似度，并按照与 Geolife 相同的实验方案进行实验。Top-K 相似性检索的性能比较，效率研究和参数灵敏度的实验结果分别在小节 4.3.2.1、4.3.2.2 和 4.3.2.3 中讨论。

为了评估 NeuTraj 在计算轨迹对相似度时的有效性，本文对两个数据集进行了轨迹聚类实验，比较了精确相似度和基于轨迹表示向量相似度之间聚类结果的差异。实验首先利用 DBSCAN 分别基于两种相似度获得聚类结果，并根据同质性、完整性、V 度量和调整随机指数四个聚类评估指标评估两个聚类结果下的差异。轨迹聚类的结果显示在小节 4.3.2.4 中展示。

另外，本文还进行零样本学习测试，以验证 NeuTraj 能否适用于没有真实轨迹，仅有道路网络的城市。基于北京的道路网络（Pallotta 等.，2013），本文仿真了 6000 条轨迹作为种子并利用仿真轨迹相似度训练 NeuTraj。然后基于真实的 Geolife 轨迹验证 Top-K 检索结果。零学习的结果在 4.3.2.5 中展示。

4.3.1.3 对比方法及评价指标

在四种轨迹相似性度量方法上，本文将 NeuTraj 与三类方法进行了比较：

- 近似算法：除了 ERP 没有近似算法外，其他三种相似性度量都有近似算法。本文利用 Driemel 和 Silvestri（2017）所提出的方法用于计算 Fre´chet 和 DTW 距离，Backurs 和 Sidiropoulos（2016）提出的方法用于计算 Hausdorff 距离。我们将这些近似算法统称为 AP。
- Siamese Network（Bromley 等.，1993）：此类方法是基于 Siamese 网络的度量学习方法。轨迹编码模型采用标准 LSTM 的 RNN 并利用随机采样的轨迹对学习编码器参数，此类方法统称为 Siamese。
- NeuTraj 变体：最后，本实验测试了 NeuTraj 的两种变体。（1）将 NeuTraj 中的权重采样替换为随机采样测试加权排序损失的有效性。该变体记为 NT-NO-WS。（2）用标准 LSTM 替换 NeuTraj 中的 SAM 增强的 LSTM 单元，

记为 NT-NO-SAM，测试所提出的空间注意力记忆机制的效果。

实验结果使用三种不同的指标进行评估。第一个是 Top-K 命中率，它统计了 NeuTraj 计算的前 k 个轨迹与准确 Top-K 轨迹的重叠百分比。本实验统计了 Top-10（HR @ 10）和 Top-50 检索（HR @ 50）的命中率。第二个是对 Top-10 名准确结果（R10 @ 50）的 Top-50 召回率。该指标评估了通过 NeuTraj 生成的 Top-50 轨迹中包含了多少 Top-10 的精确结果。第三个指标是 Top-10 个结果的平均距离失真，表示为δ_{H10}和δ_{R10}。根据 NeuTraj 结果和准确结果，计算其 Top-10 轨迹与检索轨迹的平均距离之差记作δ_{H10}；δ_{R10}基于 NeuTraj 结果的 Top-50 轨迹比较其与 Top-10 个最相似的轨迹平均距离之差。该距离越小，方法的准确率越强。

4.3.1.4　参数设置

NeuTraj 中的关键参数包括：（1）表示向量维数 d;（2）注意记忆读取器的扫描宽度 w。本实验通过在范围{16,32,64,128,256}中进行网格搜索来确定 d 的值。通常，性能随着 d 增加而在足够大时逐渐稳定。对于 w，本实验在{0,1,2,3,4}中进行了搜索，发现在两个数据集上 w = 2 时实验结果最佳。因此设置 d = 128 和 w = 2。此外，将训练是每批锚轨迹数量设置为 20，将样本大小 n 设置为 10。小节 4.3.2.3 阐述了参数敏感度分析的结果。

4.3.2　实验结果分析

本节从 Top-K 检索准确率、Top-K 检索效率、参数敏感度分析、轨迹聚类和零样本学习五个方面分析了 NeuTraj 的准确率、效率和灵活性。

4.3.2.1　Top-K 检索准确率比较

表 4.2 展示了不同相似性度量下，NeuTraj 在 Top-K 相似度检索任务上的准确率。两个数据集上，NeuTraj 在大多数指标中都明显优于对比方法。以 Fre´chet 距离为例。与近似算法（AP）相比，NeuTraj 及其变体，将 H10@10 提高近两倍，R10 @ 50 增加约 70%，平均距离减少约 69%。NeuTraj 相对于 Siamese 网络的优势也很明显。造成这种现象的原因有两个方面。首先，与 Siamese 网络相比，加权采样和排序损失可以产生区分力更强的训练轨迹样本对

和更有针对性的损失函数。其次，空间注意记忆模块建模轨迹的空间邻近特性，这有利 NeuTraj 产生高质量的轨迹表示向量。在四个相似性度量中，NeuTraj 在 DTW 上的性能比其他三个相似性度量差。其原因是基于轨迹表示向量的距离是标准度量，而 DTW 则不是。这样的系统误差使得 NeuTraj 在 DTW 表现不如其他三种相似性度量方法。

表 4. 2 Top-K 相似性检索中不同方法的性能比较

Table 4. 2 Performance comparison for different methods on Fre´chet, Hausdorff, ERP and DTW distances

数据	方法	Fre´chet			
		HR@10	HR@50	R10@50	$\delta_{H10}/\delta_{R10}$
Geolife	AP	0.2374	0.2542	0.5290	213/87
	Siamese	0.4631	0.6032	0.8121	162/34
	NeuTraj	**0.4947**	**0.6786**	**0.8403**	**84/18**
Porto	AP	0.2542	0.2851	0.5520	208/79
	Siamese	0.4740	0.5802	0.7970	128/27
	NeuTraj	**0.5225**	**0.6351**	**0.8292**	**89/8**

数据	方法	Hausdorff			
		HR@10	HR@50	R10@50	$\delta_{H10}/\delta_{R10}$
Geolife	AP	0.2967	0.3180	0.5363	217/113
	Siamese	0.3120	0.4236	0.6640	199/69
	NeuTraj	**0.3691**	**0.4870**	**0.7416**	**152/42**
Porto	AP	0.2832	0.2966	0.5620	201/86
	Siamese	0.3834	0.4999	0.7760	165/48
	NeuTraj	**0.4372**	**0.5714**	**0.8089**	**101/15**

（续表）

数据	方法	ERP		
		HR@10	HR@50	R10@50
Geolife	AP	——	——	——
	Siamese	0.5787	0.7363	0.8964
	NeuTraj	**0.6137**	**0.7780**	**0.9424**
Porto	AP	——	——	——
	Siamese	0.4982	0.6893	0.9043
	NeuTraj	**0.5427**	**0.7297**	**0.9277**

数据	方法	DTW		
		HR@10	HR@50	R10@50
Geolife	AP	**0.3870**	0.4268	**0.7139**
	Siamese	0.2680	0.4582	0.6172
	NeuTraj	0.3067	**0.4832**	0.6513
Porto	AP	0.3798	0.4160	0.7010
	Siamese	0.3832	0.4804	0.7602
	NeuTraj	**0.4370**	**0.5613**	**0.8396**

表 4. 3 NeuTraj 变体实验结果

Table 4. 3 Results of ablation experiments for different methods on Fre´chet, Hausdorff, ERP and DTW distances

数据	方法	Fre´chet			
		HR@10	HR@50	R10@50	$\delta_{H10}/\delta_{R10}$
Geolife	NT-NO-WS	0.4736	0.6353	0.7996	139/27
	NT-NO-SAM	0.4842	0.6483	0.8198	117/23
	NeuTraj	**0.4947**	**0.6786**	**0.8403**	**84/18**
Porto	NT-NO-WS	0.4990	0.5883	0.7981	102/10
	NT-NO-SAM	0.5154	0.6121	0.8171	92/10
	NeuTraj	**0.5225**	**0.6351**	**0.8292**	**89/8**

（续表）

数据	方法	Hausdorff			
		HR@10	HR@50	R10@50	$\delta_{H10}/\delta_{R10}$
Geolife	NT-NO-WS	0.3338	0.4393	0.6273	169/55
	NT-NO-SAM	0.3574	0.4607	0.7219	157/46
	NeuTraj	**0.3691**	**0.4870**	**0.7416**	**152/42**
Porto	NT-NO-WS	0.4190	0.5628	0.7909	140/33
	NT-NO-SAM	0.4238	0.5691	0.8033	126/16
	NeuTraj	**0.4372**	**0.5714**	**0.8089**	**101/15**

数据	方法	ERP		
		HR@10	HR@50	R10@50
Geolife	NT-NO-WS	0.5880	0.7170	0.8686
	NT-NO-SAM	0.6090	0.7537	0.9291
	NeuTraj	**0.6137**	**0.7780**	**0.9424**
Porto	NT-NO-WS	0.5192	0.6920	0.8917
	NT-NO-SAM	0.5382	0.7111	0.9107
	NeuTraj	**0.5427**	**0.7297**	**0.9277**

数据	方法	ERP		
		HR@10	HR@50	R10@50
Geolife	NT-NO-WS	0.2591	0.4610	0.6260
	NT-NO-SAM	0.2881	0.4792	0.6482
	NeuTraj	**0.3067**	**0.4832**	**0.6513**
Porto	NT-NO-WS	0.3930	0.5013	0.7919
	NT-NO-SAM	0.4238	0.5425	0.8148
	NeuTraj	**0.4370**	**0.5613**	**0.8396**

NeuTraj 变体实验的结果如表 4.3 所示。将 NeuTraj 与其变体进行比较，可以进一步验证 NeuTraj 中两个主要模块的有效性。以 Geolife 数据集上 Fre´chet 距离的结果为例：（1）通过 SAM 模块，NeuTraj 将 NT-NO-WS 的 HR @ 10 从 0.46 提高到 0.47; （2）通过包括加权采样和优化模块，NeuTraj 将 NT-NO-WS 的 HR @ 10 从 0.47 提高到 0.49。Porto 数据集和其他三个相似性度量的趋势相似。另外，两个数据集中 HR @ 10 和 HR @ 50 的值不是很高。其原因在于两个数据集中的轨迹都有很多近似重复的轨迹。该现象可以从δ_{H10}的值上看出。

4.3.2.2　Top-K 检索效率比较

本小节研究了 NeuTraj 在 Top-K 相似性检索上的效率，首先研究了 NeuTraj 在线相似性检索的时间开销，然后研究了 NeuTraj 离线训练时间开销。该实验在配有 Inter Xeon E5 @ 2.20GHz CPU 和 Nvidia P100 GPU 的服务器上进行。

表 4. 4 无索引的在线相似度检索的时间开销

Table 4. 4 Time cost for online similarity search without index

方法	1k	5k	10k	200k
		Fre´chet		
BruteForce	8.712s	41.876s	84.480s	1639.834s
AP	1.840s	11.319s	23.107s	532.652s
NT-NO-SAM	0.461s	0.471s	0.489s	1.576s
NeuTraj	0.461s	0.470s	0.490s	1.574s
		Hausdorff		
BruteForce	0.238s	1.416s	2.981s	51.642s
AP	0.127s	0.154s	0.179s	3.426s
NT-NO-SAM	0.026s	0.046s	0.072s	1.133s
NeuTraj	0.024s	0.047s	0.073s	1.131s
		ERP		
BruteForce	0.409s	1.982s	3.807s	73.054s
NT-NO-SAM	0.027s	0.046s	0.081s	1.154s
NeuTraj	0.026s	0.047s	0.081s	1.152s
		DTW		
BruteForce	0.305s	1.482s	3.070s	59.054s
AP	0.119s	0.142s	0.185s	4.021s
NT-NO-SAM	0.023s	0.044s	0.066s	1.028s
NeuTraj	0.021s	0.043s	0.067s	1.027s

在线相似度检索的时间开销。该部分实验分为无索引相似性检索实验和有索引相似性检索实验。通过无索引实验可以验证 NeuTraj 与已有近似方法之间的加速效果对比；通过有索引的相似性检索实验，证明 NeuTraj 的灵活性，即可以与已有轨迹索引技术相结合，进一步加速相似性检索过程。

无索引的相似性检索。表 4.4 显示了 NeuTraj 在不同大小轨迹数据集上进行 Top-K 相似性检索的时间开销。具体来说，首先从 Porto 数据集中分别随机抽取大小为 1K，5K，10K 和 200K 的四个轨迹子数据库。然后，使用 NeuTraj 为每

个轨迹执行 Top-50 相似性检索，得到 50 条轨迹后计算它们与检索轨迹的准确相似度。表 4.4 统计了处理一个查询的平均时间开销，NeuTraj 与 BruteForce 方法，近似算法（AP），以及基于神经网络的方法（NT-NO-SAM）进行了比较。可以看出，NeuTraj 实现了比 BruteForce 高 50-1500 倍的加速，对比近似算法 AP 具有 3-500 倍的加速。

表 4.5 有索引的在线相似度检索的时间开销

Table 4. 5 Time cost for online similarity search with index

方法	1k	5k	10k	200k
	Bounding Box R-tree Index			
BruteForce	5.526s	27.802s	54.558s	1070.433s
AP	0.438s	1.731s	4.372s	62.853s
NeuTraj	0.005s	0.029s	0.056s	0.868s
涉及轨迹数	675	3377	6736	134051
	Grid-based Inverted Index			
BruteForce	5.633s	27.793s	54.993s	1098.042s
AP	0.460s	1.911s	4.722s	66.072s
NeuTraj	0.006s	0.030s	0.065s	1.173s
涉及轨迹数	685	3424	6834	136201

有索引相似性检索。本实验将 NeuTraj 于两种主流的索引技术结合：（1）边界框 R-tree;（2）基于网格的倒排索引，并随机选择 200 个轨迹查询来测试 NeuTraj 的灵活性。在各种大小的轨迹子数据库上，利用 Fre´chet 计算轨迹相似度，本实验将索引扩展的 NeuTraj 与两类方法进行了对比：BruteForce 方法和近似算法。表 4.5 统计了检索的平均时间开销。分析实验结果后我们发现 NeuTraj 在两个索引结构下的时间开销均低于对比方法。对比近似算法，NeuTraj 实现了超过 30 倍的加速。

表 4.6 训练的时间成本

Table 4. 6 Time cost for offline model training

方法	训练			Embedding
	t_{epoch}	$\#_{epoch}$	t_{total}	200k
Siamese	164s	71	11644s	411s
NeuTraj	285s	15	5130s	639s

Ablations Experiments

（续表）

NT-NO-SAM	168s	15	2520s	412s
NT-NO-WS	283s	20	5660s	636s

离线训练和轨迹编码的时间开销。对比已有近似算法，NeuTraj 存在额外的时间开销，这部分开销包括模型训练和轨迹编码的时间开销。本实验分别统计了离线训练时间和轨迹编码的时间开销。

离线训练时间。表 4.6 统计了在 Fre´chet 距离下对 Porto 数据集的离线训练时间。对于 2,000 个种子轨迹，NeuTraj 在 20 个时期内收敛。对于 NeuTraj，一个时期的训练时间约为 5 分钟，因此 NeuTraj 的整个训练时间不到 2 小时。相比之下，Siamese 网络需要超过 60 个时期才能收敛，这比 NeuTraj 慢约 3 倍。另外，本实验还对比了 NeuTraj 与 NT-NO-SAM 的收敛速度，并观察到 NT-NO-SAM 具有比 NeuTraj 更快的收敛速度，这是因为 SAM 模块引入了额外的计算，增加了处理轨迹的时间开销。

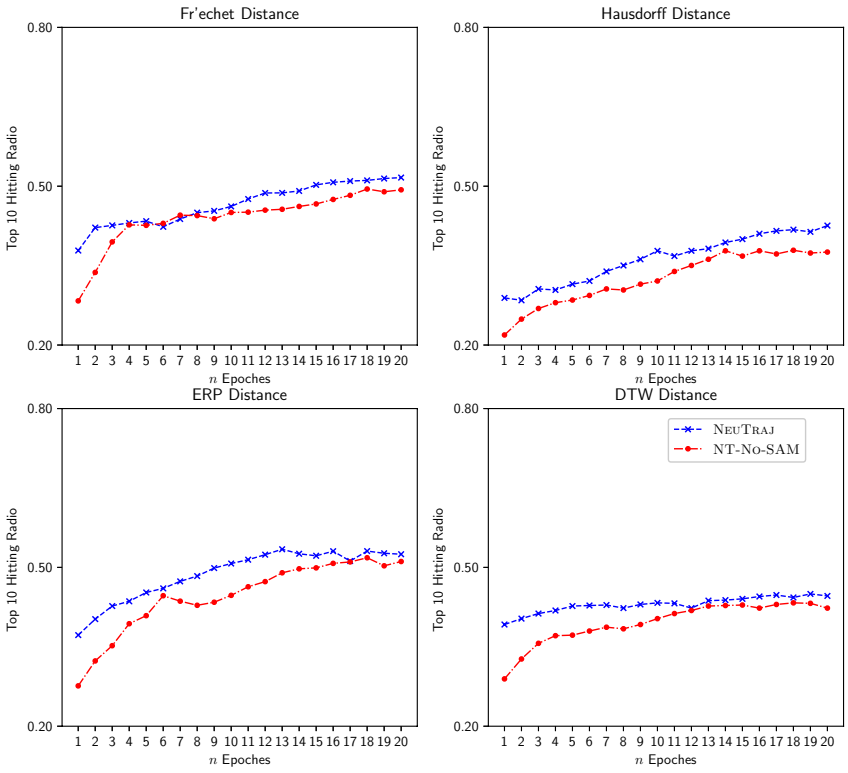

图 4. 5 NeuTraj 的收敛曲线

Figure 4. 5 The convergence curve of NeuTraj and NT-NO-SAM

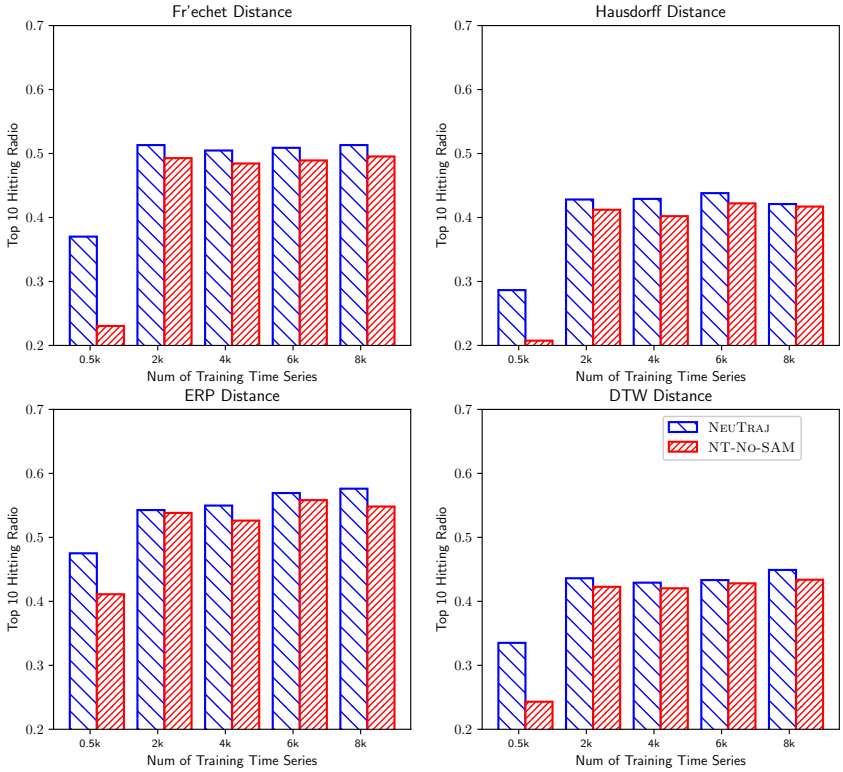

图 4. 6 训练数据大小对 NeuTraj 的影响

Figure 4. 6 HR@10 of NeuTraj and NT-NO-SAM with varying training data size

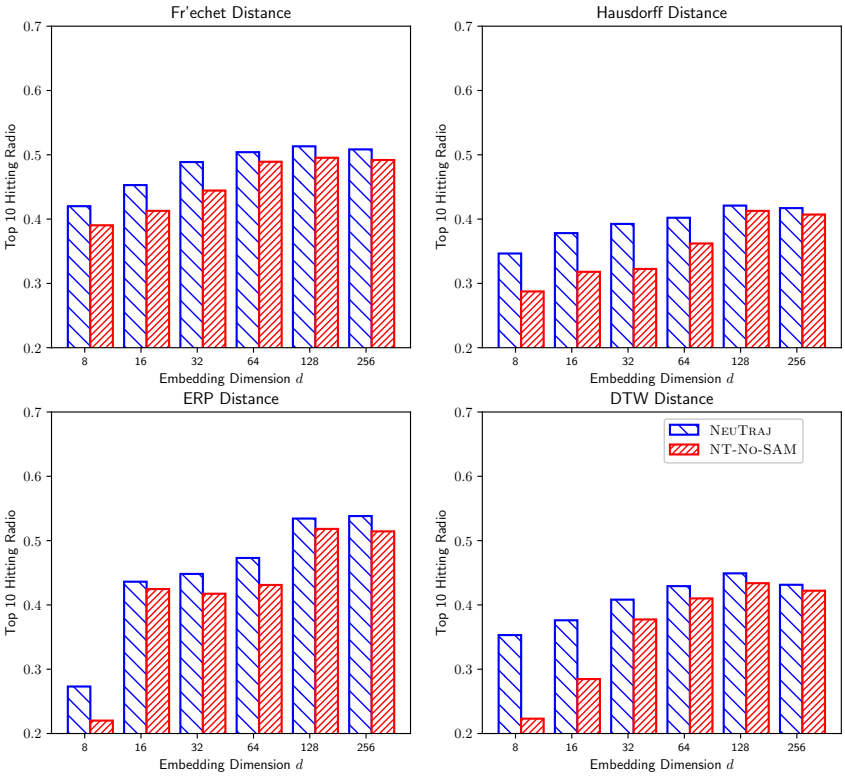

图 4. 7 表示向量维度变化对 NeuTraj 的影响

Figure 4. 7 HR@10 of NeuTrajand NT-NO-SAM on with varying embedding size d

离线轨迹编码时间开销。离线训练的另一部分是轨迹编码，该程序通过使用训练好的模型生成轨迹的表示向量。在本实验中，轨迹编码的批大小被设置为2000 并统计生成 20 万条轨迹表示向量的时间成本。所有统计结果如表 4.6 所示，可以看出基于 SAM 单元的方法（NeuTraj，NT-NO-WS）比基于标准循环神经网络单元的方法（NT-NO-SAM，Siamese）稍慢。其原因在于 SAM 单元需要更多的计算来寻找空间记忆张量中的有用信息。

4.3.2.3　参数敏感度分析

本节评估了 Porto 数据集上 NeuTraj 对三个参数的敏感性：训练数据大小，扫描宽度 w 和轨迹表示向量维度 d。

训练数据大小的敏感性：本实验研究了种子轨迹数量对 NeuTraj 性能的影响。图 4.6 显示了四种相似性度量下，NeuTraj 及 NT-NO-SAM 对训练数据大小的敏感性。将 Porto 的训练数据大小从 500 递增至 8000，如图所示，当训练轨迹超过 2,000 时，NeuTraj 的性能变得相对稳定。另外，在少量训练数据下，NeuTraj 比 NT-NO-SAM 更鲁棒。当训练轨迹的数量仅为 500 时，NeuTraj 和 NT-NO-SAM 之间的性能差异很大。这是因为 SAM 具有对已处理轨迹的空间信息的记忆能力。

轨迹表示向量维度 d 的灵敏度：本实验研究了表示向量维度 d 对 NeuTraj 性能的影响。图 4.7 显示了 d 从 8 逐步增加到 256，NeuTraj 及对比方法在 HR@10 的值。如图所示，NeuTraj 及其变体的性能首先增加然后略有下降。其原因在于参数 d 控制了 NeuTraj 的复杂度。当 d 增加时，模型具有更强的表达能力来捕获轨迹度量空间的内在结构。然而，当 d 太大时，由于训练数据的大小有限，模型容易出现过拟合，导致准确率降低。

扫描宽度 w 的灵敏度：SAM 模块中的扫描宽度 w 是控制哪些历史轨迹对当前轨迹有用的关键参数。如图 4.8 所示，随着 w 的增加，所有方法的 HR@10 均先增加后略微下降。这种现象的原因有两个：（1）随着 w 增加，SAM 读取器往往会访问更多空间网格，这在初始阶段对于编码当前轨迹很有用;（2）当 w 太大时，将不可避免地包含一些不相关轨迹的信息。即使 SAM 中的注意机制可以帮助减少这种影响，NeuTraj 的性能仍会受到损害。

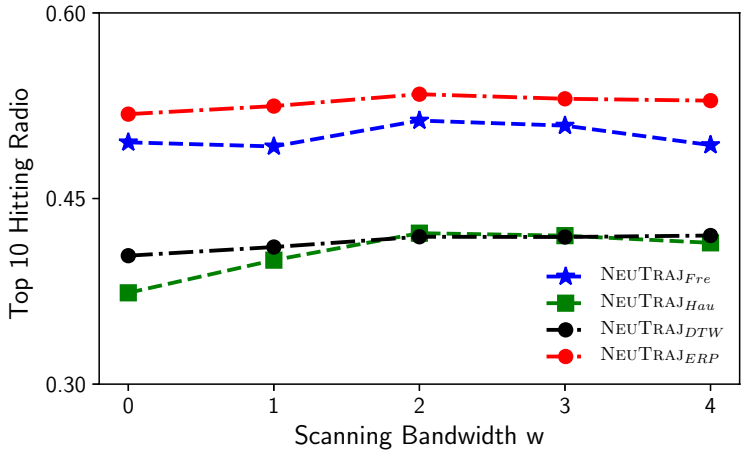

图 4.8 w 变化对 NeuTraj 的影响

Figure 4.8 HR@10 of NeuTraj with varying w

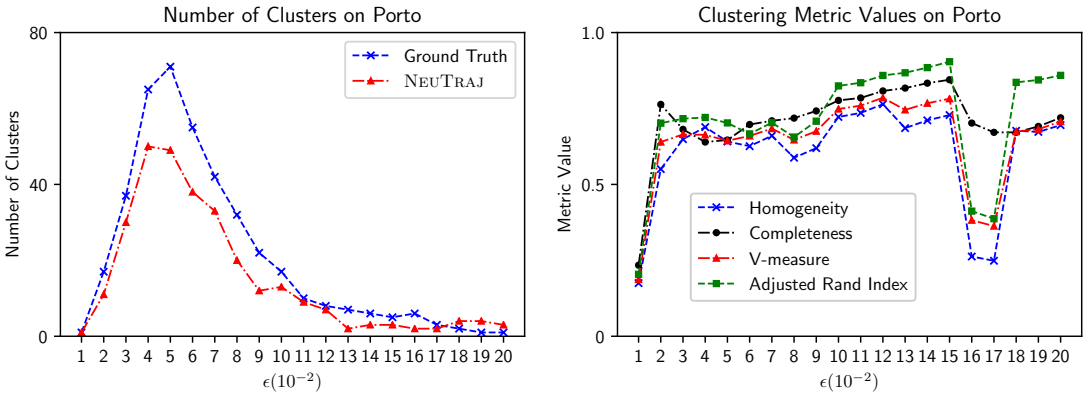

图 4.9 轨迹聚类结果

Figure 4.9 Trajectory clustering result

4.3.2.4 轨迹聚类结果

本实验的目标是通过轨迹聚类探索 NeuTraj 在计算轨迹对相似度任务中的有效性。实验在 Porto 数据集上 Fre´chet 距离下进行。基于准确的轨迹相似度和 NeuTraj 的近似轨迹相似度，本实验利用 DBSCAN 分别产生聚类结果，并比较两个聚类结果之间的差异。如图 4.9 所示，将 DBSCAN 簇内最小样本点固定为 10，随着ε的增加，两个结果的变化趋势相似。另外，本实验使用四个聚类结果评价指标评价两个聚类结果的近似程度，大部分评估指标的最佳值大于 0.8，这表明使用 NeuTraj 相似度得到聚类与使用精确相似度得到的聚类结果相近。

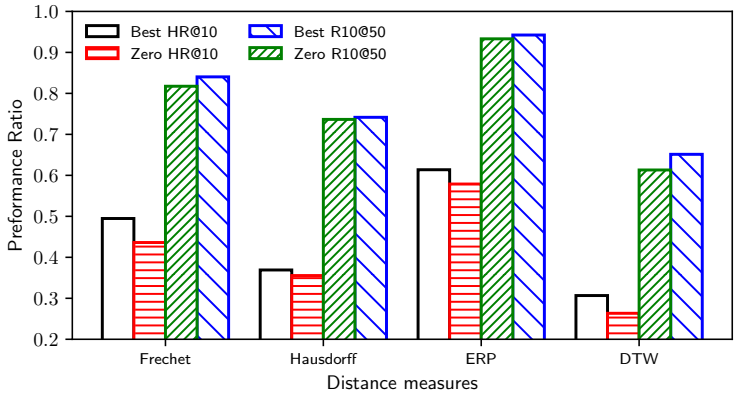

图 4. 10 Geolife 数据集上零样本学习的结果

Figure 4. 10 Illustration of zero-shot learning results on Geolife dataset

4.3.2.5　零样本学习评估

最后，本实验测试了 NeuTraj 在零样本场景下的应用效果。NeuTraj 方法依赖于真实的轨迹数据库并从数据库中采样种子轨迹作为指导。本实验还关注了在没有真实的轨迹作为种子的情况下 NeuTraj 的性能表现。具体而言，假设没有真实轨迹数据库，只有目标区域的道路网络。本实验通过模拟目标移动，产生仿真轨迹作为种子训练 NeuTraj。基于北京的道路网络（Pallotta 等.，2013），通过在道路节点图上的随机游走（Random Walk）及插值生成 6000 条仿真轨迹，利用仿真轨迹训练 NeuTraj，并用 Geolife 中的真实轨迹进行测试。图 4.10 显示了该实验中 NeuTraj 的 HR @ 10 和 R10 @ 50。可以看出，即使使用仿真轨迹及其距离作为指导，NeuTraj 在四个相似性度量上仍然可以达到大约 0.7 的 R10 @ 50。该结果表明即使没有真正的轨迹，NeuTraj 也有很好的效果。

4.4　小结

本章提出了一种种子引导深度度量学习的方法 NeuTraj 用于加速轨迹相似度计算。NeuTraj 具有快速、准确、通用和灵活的特点。它的创新性在于两个方面：（1）空间注意记忆（SAM）模块，可以增强现有的 RNN 架构捕捉轨迹之间的空间邻近特性;（2）加权采样及排序损失，有效利用种子相似度的信息，学习高质量的轨迹表示向量。在两个真实数据集上的实验结果表明，NeuTraj 可以有效地加速各种轨迹相似性度计算，同时产生比现有近似算法更高的准确率。

第 5 章　基于移动行为特征的半监督轨迹异常检测方法

本章针对轨迹异常检测中异常标签稀少的问题，提出了基于移动行为特征的半监督轨迹异常检测方法 Traj2Vec。该方法采用半监督学习的框架，将轨迹异常检测分为无监督学习和有监督异常检测两个阶段，实现了对标签数据稀少的轨迹数据集上的异常检测。在无监督学习阶段，提出了基于移动行为特征的无监督表示学习方法。在有监督异常检测阶段，提出基于分歧学习机制的轨迹异常检测方法，该方法根据数据标签稀少的特点来设计损失函数及优化方式。实验结果证明，Traj2Vec 可以显著提升异常检测综合准确率和异常轨迹的召回率。

本章组织如下：5.1 节详细阐述了研究问题的内容和挑战。5.2 节详细介绍了基于移动行为特征的半监督轨迹异常检测方法 Traj2Vec。其中，5.2.1 节介绍了半监督轨迹异常检测的问题定义和方法概述；5.2.2 节介绍了移动行为特征抽取方法；5.2.3 节详细介绍了基于序列自编码的轨迹表示学习算法；5.2.4 节介绍了基于半监督学习的轨迹异常检测方法及训练过程；5.4 节在仿真数据集和真实数据集上分别验证了上述方法的有效性；5.5 节对本章进行了小结。

5.1　引言

轨迹异常检测可以从轨迹数据库中的异常轨迹中发现高价值的异常目标，其结果可以广泛应用于态势监控、行为分析等领域。现实应用中，难以获得轨迹数据的异常标签。尽管有大量轨迹数据被采集和处理，异常轨迹的标签仍需要依靠专家经验手动标记。与此相对，训练轨迹异常检测模型需要大量的带标签数据，这使轨迹异常检测面临标签数据稀少的挑战。

为了解决轨迹异常检测中数据标签稀少的挑战，现有研究工作提出了许多利用轨迹数据集中的无标签轨迹来提升轨迹异常检测准确率的方法。根据异常检测技术不同将现有方法划分为基于模式的轨迹异常检测方法和基于半监督学习的轨迹异常检测方法。基于模式的异常检测方法利用聚类或模式挖掘的技术，无监督地发现轨迹中隐含的对异常检测有用的规律来检测异常轨迹。基于

模式的方法通常利用相似性度量来量化轨迹间的相似性，例如使用如K-Means、DBSCAN、谱聚类等聚类算法得到相似轨迹簇，并将不属于任何簇的轨迹视为异常轨迹。该类方法中的某些方法利用模式挖掘技术发现轨迹的伴随、序列和周期等模式检测异常轨迹。这类方法依赖轨迹先验知识，需要设定异常轨迹阈值。另外，这类方法虽然可以对固定地理区域、时间内的相似轨迹进行有效地聚类，但对轨迹移动行为相关的异常检测效果较差。基于半监督学习方法的研究刚刚起步，现有方法首先通过无监督的方法检测疑似异常的轨迹，然后将疑似异常轨迹交由人工判别，最后根据人工判别结果调整检测模型。此类方法需要与人交互，仍未能解决仅有少量数据标签情况下的轨迹异常轨迹检测问题。

针对上述问题，本文提出了一个基于半监督学习的异常检测方法 Traj2Vec。该方法包括无监督学习和有监督异常检测两个阶段。其中无监督学习阶段采用 RNN 对轨迹序列进行处理并学习轨迹的表示向量。具体来说，给定待处理的原始轨迹，Traj2Vec 首先将每一条原始轨迹转换成固定长度的向量，使得该向量可以有效地编码目标的移动行为。在有监督学习阶段，基于上述表示向量和少量数据标签，Traj2Vec 利用分歧学习机制训练检测轨迹异常的分类器解决数据标签稀少的问题。该方法包括轨迹预处理、移动行为抽取、序列自编码和异常检测四个步骤。

首先，轨迹预处理将目标的记录序列按时间间隔切分，将时间间隔过长的连续记录划分到不同轨迹中。其次，将轨迹输入到移动行为特征抽取算法中，该算法利用时间滑动窗口，依序抽取轨迹的移动行为特征，将位置记录序列转换为行为特征序列。再次，Traj2Vec 基于序列自编码器将轨迹移动行为特征序列进行编码，并得到轨迹移动行为的表示向量，该过程基于无监督学习来优化特征序列的重构误差。最后，以序列自编码层的轨迹表示向量为基础，利用少量的带标签的数据训练轨迹的异常检测分类器。实验基于仿真数据和真实数据中对 Traj2Vec 进行评估。实验结果表明，与已有轨迹异常检测方法相比，Traj2Vec 在异常检测准确率略有提高（3%）的前提下，将异常轨迹的召回率提高了 10%。

5.2　基于移动行为特征的半监督轨迹异常检测方法

本节详细介绍了基于移动行为特征的半监督轨迹异常检测方法 Traj2Vec。首先定义了轨迹异常检测任务并概述了 Traj2Vec 的主要流程。然后针对 Traj2Vec 中的主要模块，详细阐述了其中模型结构和计算过程。

5.2.1　问题定义与方法概述

本章首先定义了仅有少量标签数据下的轨迹异常检测问题，然后简要介绍了本文提出的基于移动行为特征的半监督异常检测方法 Traj2Vec。

5.2.1.1　问题定义

给定待检测的目标集合$O = \{o_1, o_2, \dots, o_n\}$，每个目标$o_i$的记录序列表示为$S_o = (x_1, x_2, \dots, x_M)$。每条记录$x$则由一个元组$(t_x, l_x, a_x, o_x)$表示，$t_x$表示时间戳，$l_x$一个表示经纬度坐标的二维向量，$a_x$表示由其他传感器采集的属性信息，$o_x$表示目标的 ID。

首先，需要将序列S_o分割成多个轨迹序列$T_o = (T_1, T_2, \dots, T_n)$。给定$S_o = (x_1, x_2, \dots, x_M)$与时间间隔阈值$\Delta t > 0$，如果子序列$S_o^T = (x_i, x_{i+1}, \dots, x_{i+k})$满足条件：（1）$\forall 1 < j \le k, t_{x_j} - t_{x_{j-1}} \le \Delta t$；（2）没有其他满足条件（1）的子序列包含$S_o^T$，则称$S_o^T$为一条轨迹。

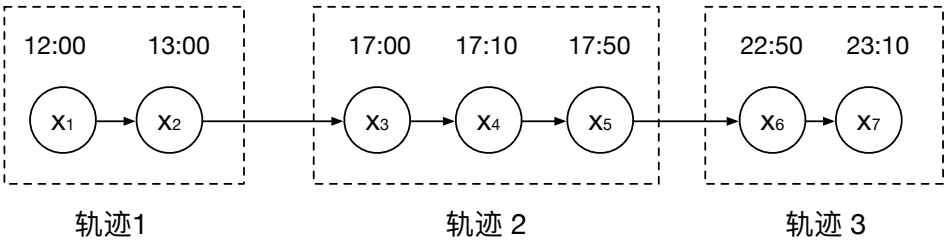

图 5. 1 序列分割及轨迹生成

Figure 5. 1 Partition the sparse sequence into trajectories

图 5.1 描述了轨迹生成过程。给定$S_o = (x_1, x_2, \dots, x_7)$，$\Delta t = 3$小时，整个序列分割成三条轨迹：$T_o = (T_1, T_2, T_3)$。

合并所有目标的轨迹可以获得一个轨迹集合$\mathcal{T} = \{T_1, T_2, \dots, T_N\}$。对于整个轨迹集合$\mathcal{T}$，仅有少量的带标签的轨迹数据$\mathcal{T}^L = \{T'_1, T'_2, \dots, T'_M\}$。轨迹异常检测的目标是检测$\mathcal{T}$中存在的异常轨迹。具体来说，根据目标移动模式生成一个异常轨

迹的集合$\mathcal{C} = \{A_1, A_2, \ldots, A_k\}$，使得$\mathcal{C}$包含所有$\mathcal{T}$的异常轨迹。

5.2.1.2 方法概述

完整的轨迹表示生成流程如图 5.2 所示：

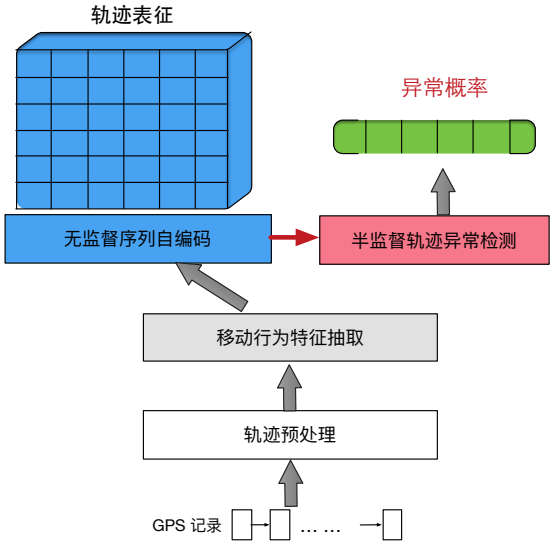

图 5. 2 Traj2Vec 示意图

Figure 5. 2 The framework for semi-supervised trajectory anamoly detection

- 轨迹预处理层：该层以原始 GPS 轨迹数据作为输入。由于原始轨迹序列包含噪声，且连续轨迹记录间可能存在极大的时间间隔。因此，该层将移除低质量的轨迹记录，并将序列按照时间连续性规则切割成多个轨迹。
- 移动行为特征提取层：在该层中，输入的轨迹将会被移动行为特征提取算法处理。通过滑动窗口，轨迹序列将被转换移动行为特征序列。
- 序列到序列的自编码层：利用基于 RNN 的序列到序列的自编码，将移动特征序列编码成固定长度的向量。该向量能够有效地保存轨迹的行为信息。
- 异常检测层：最后，以序列自编码层的轨迹表示向量为基础，利用少量的带标签的数据，使用分歧学习的方法训练轨迹的异常分类器。

可以看出 Traj2Vec 中有三个核心模块，即轨迹移动行为特征抽取，无监督序列自编码及半监督轨迹异常检测。下面将逐个介绍以上三个步骤。

5.2.2 轨迹移动行为特征抽取

为了提取轨迹的行为特征，本文利用滑动时间窗口顺序地遍历一条轨迹的所有记录。如图 5.3 所示，在一个时间窗口内，Traj2Vec 抽取一组具有时空不变性

的特征用于描述目标的移动行为。详细抽取过程如下：

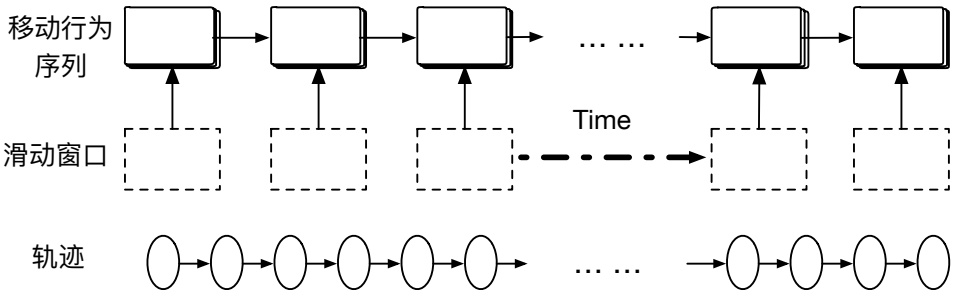

图 5.3 移动行为提取

Figure 5. 3 Moving behavior extraction

首先，定义滑动窗口及滑动偏移量。令L_p与$offset_p$分别表示滑动窗口的宽度与偏移量。为了更好地捕捉轨迹的移动特征，本文将$offset_p$的值定义为$1/2\ L_p$即$offset_p = 0.5 \times L_p$。通过这种方式，轨迹中的每条记录都会将被分配到两个窗口中，这样可以更充分地捕获轨迹的移动行为特征。由于轨迹记录的密度不均匀，特征提取时会出现一些空窗口，如图 5.4 中的W_6，这些空窗口的移动行为特征计算方法将在下文中详细阐述。

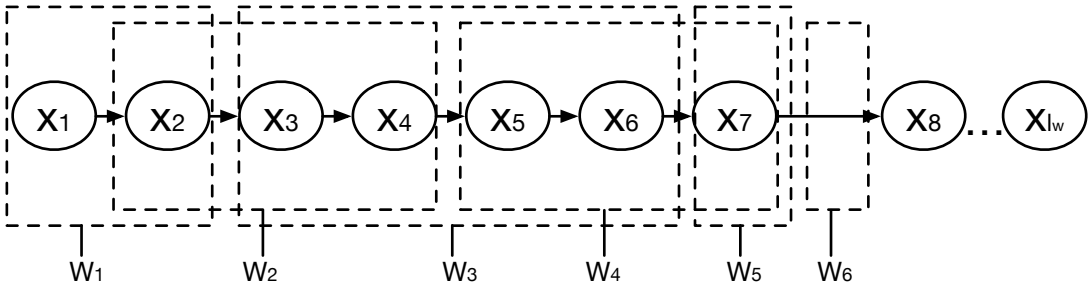

图 5.4 滑动窗口定义

Figure 5. 4 Sliding time windows generation

接下来，针对每个窗口中的轨迹记录，Traj2Vec 通过提取连续两个轨迹记录间属性的变化来描述目标移动行为的变化。假设一个包含R条轨迹记录的窗口$W = (x_1, x_2, \dots, x_R)$。移动行为的特征将包括：时间间隔$\Delta t_i$，空间变化$\Delta l_i$，速度变化$\Delta s_i$，转角变化$\Delta r_i$。

$$\Delta t_i = t_{x_i} - t_{x_{i-1}};$$

$$\Delta l_i = l_{x_i} - l_{x_{i-1}};$$

$$\Delta s_i = s_{x_i} - s_{x_{i-1}};$$

$$\Delta r_i = r_{x_i} - r_{x_{i-1}} \tag{5.1}$$

其中，i从 2 到R取值。通过这种方式，一个包含R条轨迹记录的窗口将包含$R-1$个移动行为特征$(\Delta l, \Delta s, \Delta r)$。

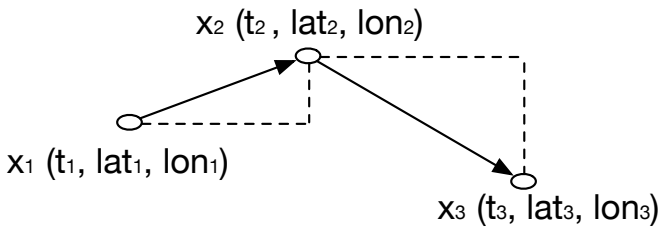

图 5.5 行为特征计算示意图

Figure 5. 5 Attributes completely

轨迹记录的速度和转角率同样可以根据坐标信息计算得到。如图 5.5 所示，给定一个包含T个轨迹记录的轨迹$T = (x_1, x_2, \dots, x_T)$。其中，$x$仅包含时间戳与坐标信息$(t, lon, lat)$。对第一个轨迹记录$x_1$，假设$s_{x_1} = 0$，$r_{x_1} = 0$。可以通过以下公式计算每个轨迹记录的速度和转角率：

$$s_{x_i} = \frac{\sqrt{\left(lat_{x_i} - lat_{x_{i-1}}\right)^2 + \left(lon_{x_i} - lon_{x_{i-1}}\right)^2}}{t_{x_i} - t_{x_{i-1}}} \ ;$$

$$r_{x_i} = arctan \frac{lon_{x_i} - lon_{x_{i-1}}}{lat_{x_i} - lat_{x_{i-1}}} \tag{5.2}$$

其中，i从 2 到T取值。经过该步骤，每个轨迹记录将获得速度与转角率特征。当$R \geq 1$时，针对每个轨迹记录，计算$\Delta t_i, \Delta l_i, \Delta s_i$与$\Delta r_i$，其中$i$从 1 到$R$取值。

然后，进一步计算这些特征的变化率$f_i = \left(f_{\Delta l_i}, f_{\Delta s_i}, f_{\Delta r_i}\right)$：

$$f_{\Delta l_i} = \frac{\Delta l_i}{\Delta t_i} \ ; \ f_{\Delta s_i} = \Delta s_i \ ; \ f_{\Delta r_i} = \Delta r_i \tag{5.3}$$

对于两个连续的轨迹记录，$f_{\Delta l_i}$代表平均速度，$f_{\Delta s_i}$表示速度变化量，$f_{\Delta r_i}$表示转角变化量。为每个轨迹对计算以上特征后，将得到一个新的特征集合$f = \{f_1, f_2, \dots, f_R\}$。

最后，Traj2Vec 利用f的统计值生成指定滑动窗口中的特征。这里，对选择 6 个统计特征$\{$均值, 最大值, 75%分位数, 50%分位数, 25%分位数, 最小值$\}$。总的来说，每个窗口的移动行为特征b包含$3 \times 6 = 18$个维度，即：

$$\{f_{\Delta l}, f_{\Delta s}, f_{\Delta r}\} \times \{\text{均值}, \text{最大值}, 75\%\text{分位数}, 50\%\text{分位数}, 25\%\text{分位数}, \text{最小值}\}$$

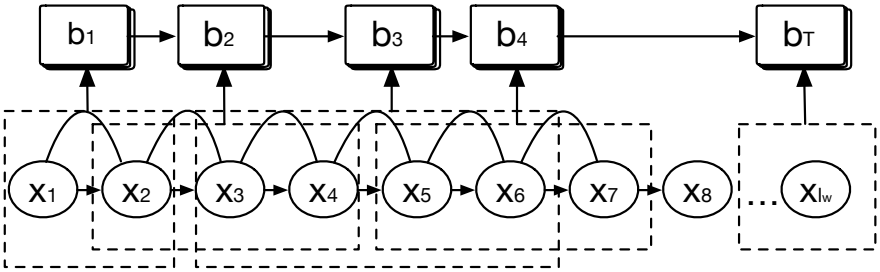

图 5.6 移动行为特征序列生成

Figure 5. 6 The generation of moving behavior sequence

算法 5.1：行为特征获取算法

输入：

轨迹T的 GPS 记录

输出：

轨迹T的行为序列B_T

Initialize $B_T = []$
$windows = sliding\_windows(T)$
for each windows W in *windows* do
  if $len(W.records) \geq 1$ **then**
    Initialize $F_W = []$
    for each record r_i in *W.records do*
      $r_{i-1} = find\_pre(r_i)$
      $F_i = compute\_features(r_i, r_{i-1})$
      $F_W.add(F_i)$
    end for
    $B_W = generate\_behavior(F_W)$
    $B_T.add(B_W)$
  end if
end for
return B_T

如图 5.6 所示，对于$\mathcal{T}$中的每条轨迹，经过上述处理后，可以生成相应的移动行为特征序列。该轨迹移动行为特征序列的集合由$\{B_{T_1}, B_{T_2}, \ldots, B_{T_N}\}$表示。轨迹移动行为特征提取算法流程见算法 5.1。得到轨迹移动行为特征序列后，将其输

入序列自编码器（auto-encoder），学习表示向量。

5.2.3 移动行为序列自编码

该步骤基于RNN的序列自编码生成$\mathcal{B}$中每条移动行为特征序列的表示向量。在实际应用中，普通的RNN在处理长序列学习上存在着不足，且容易导致梯度消失或爆炸。为了改善上述缺点，本文使用了LSTM和GRU两种循环神经网络单元作为$f(.,.)$编码轨迹。

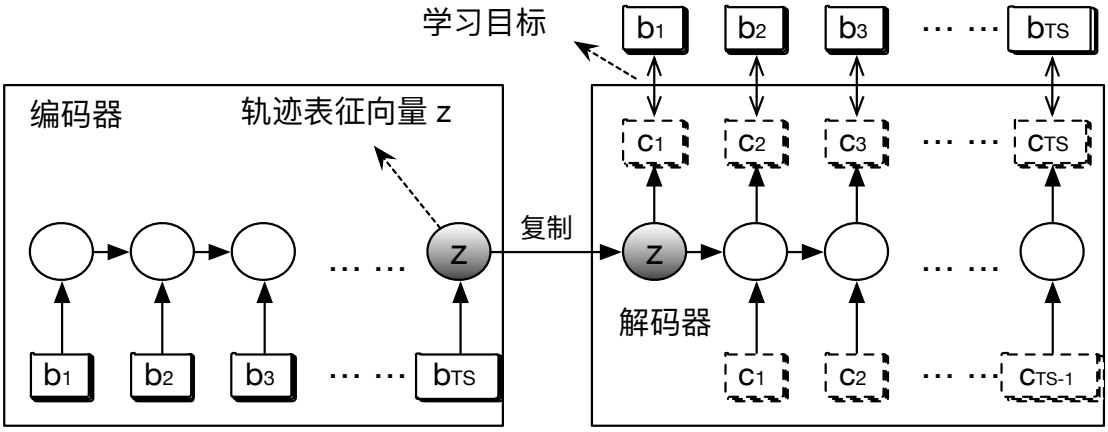

图 5.7 序列到序列的自动编码机结构

Figure 5. 7 Architecture of sequence-to-sequence auto-encoder

本文使用了序列到序列自动编码结构，学习轨迹的表示向量。该结构对移动行为序列进行重构，为轨迹生成了一个固定长度的表示向量。如图5.7所示，序列到序列自动编码模型由两个RNN组成：左侧为编码RNN与右侧为解码RNN。该模型的输入是行为序列$B_{T_i} = (b_1, b_2, \dots, b_{TS})$。编码 RNN 依次读取输入序列，并更新隐含层向量h_t:

$$h_t = f(h_{t-1}, b_t) \tag{5.4}$$

当最后一个移动行为特征b_{TS}被处理，隐含层的状态h_{TS}将作为整个序列的表示。然后，解码 RNN 将以h_{TS}作为初始的隐含层状态，生成输出c_1，以及后续的$c_2, \dots, c_{TS}$。解码RNN的更新公式如下：

$$h_t^d = f\left(h_{t-1}^d, c_{t-1}; h_{TS}\right) \tag{5.5}$$

整个自编码器的优化目标是重构输入序列$B_{T_i} = (b_1, b_2, \dots, b_{TS})$。换句话说，优化目标是最小化编码前移动行为序列$(b_1, b_2, \dots, b_{TS})$与解码后移动行为序列

$(c_1, c_2, \dots, c_{TS})$之间的差别。因此，Traj2Vec 利用均方误差作为损失函数：

$$L_{MSE} = \sum_{t=1}^{TS} \|b_t - c_t\|^2 \tag{5.6}$$

该训练过程将不需要任何标注的数据。由于输入序列可以通过解码器利用z进行解码，所以向量z能够有效的表示输入的行为序列$B_{T_i} = (b_1, b_2, \dots, b_{TS})$。通过上述处理，可以获得轨迹的移动行为向量集合$Z = \{z_{T_1}, z_{T_2}, \dots, z_{T_N}\}$。

5.2.4　有监督轨迹异常分类方法

利用上述无监督学习中的轨迹编码器，本模块利用多层感知器 MLP（Multilayer Perceptron）层+softmax 构建异常轨迹分类器。然后利用轨迹数据集中的有标签数据$\mathcal{T}^L = \{T'_1, T'_2, \dots, T'_M\}$进一步优化（fine-tuning）分类器参数。由于轨迹数据库中异常轨迹数量稀少，$\mathcal{T}^L$中的样本不足以直接训练分类器，因此本文提出一种分歧学习方法，该方法可以利用无标签轨迹中的信息辅助训练异常分类器。异常检测的分歧学习方法分为三步：

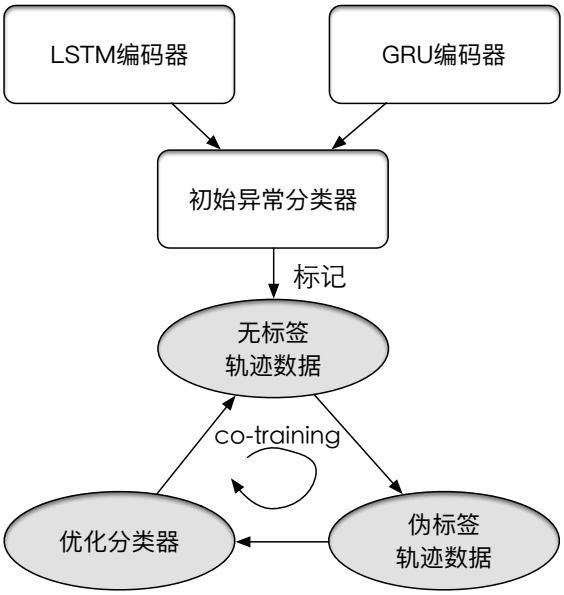

图 5.8 基于分歧学习的轨迹异常检测

Figure 5. 8 Disagreement-based learning in trajectory anomaly detection

- 第一步，利用无监督学习中预训练好的 RNN 轨迹编码器，构建初始异常分类器。首先将 LSTM 和 GRU 作为 RNN 循环单元的两个编码器构成两个初始分类器，C_0^{LSTM}和C_0^{GRU}。然后利用带标签的轨迹数据$\mathcal{T}^L$优化上述初始分类器。

- 第二步，利用优化后的初始分类器，Traj2Vec 首先标记轨迹数据库中的无标签的轨迹数据$\mathcal{T}^U = \mathcal{T} - \mathcal{T}^L$，然后在两个分类器的标记结果中分别选择分类置信度高的轨迹构成伪标签轨迹数据集$\mathcal{T}_{LSTM}^{PL}$和$\mathcal{T}_{GRU}^{PL}$。最后将伪标签轨迹涉及到的轨迹从相关轨迹从$\mathcal{T}^U$中剔除，即$\mathcal{T}^U = \mathcal{T}^U - \{\mathcal{T}_{LSTM}^{PL} \cup \mathcal{T}_{GRU}^{PL}\}$。
- 第三步，使用协同训练的方式，迭代地优化两个分类器。具体来说，首先利用$\mathcal{T}_{LSTM}^{PL} \cup \mathcal{T}^L$优化$\mathcal{C}_0^{GRU}$并同时利用$\mathcal{T}_{GRU}^{PL} \cup \mathcal{T}^L$优化$\mathcal{C}_0^{LSTM}$并将优化之后的分类器记作$\mathcal{C}_1^{LSTM}$和$\mathcal{C}_1^{GRU}$。然后，使用$\mathcal{C}_1^{LSTM}$和$\mathcal{C}_1^{GRU}$再次标记无标签轨迹数据$\mathcal{T}^U$，重复上述过程直到两个分类器检测参数都不再变化。

以上步骤中的核心部分在于第二步和第三步的协同训练过程。Traj2Vec 中协同训练的详细过程如图 5.9 所示。

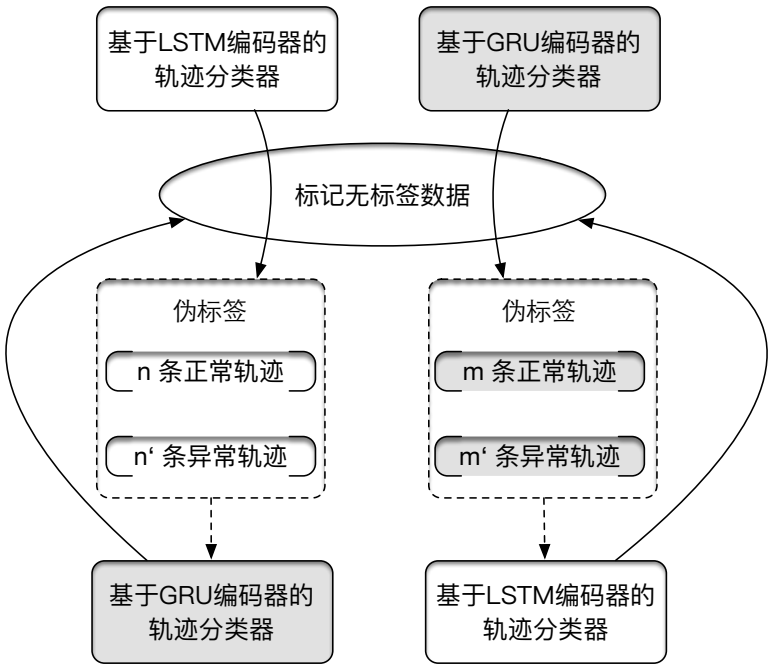

图 5.9 协同训练示意图

Figure 5. 9 Illsutration of co-training

当输入轨迹为T时，异常分类器的输出的向量为q^T。当异常检测任务是 0，1 分类时该向量为二维向量，其中的值可以看作轨迹为异常轨迹的概率分布。在生成伪标签轨迹时，Traj2Vec 首先标记所有无标签的轨迹，计算其异常概率。然后取异常概率最低的 n 条轨迹作为伪正常轨迹、n’条异常概率最高的轨迹作为伪异常轨迹。利用伪标签轨迹，Traj2Vec 交替地训练基于 LSTM 编码器的轨迹分类器和基于 GRU 编码器的轨迹分类器。重复上述过程直至两个分类器参数都不再变

化，然后使用测试数据中异常检测准确率高的分类器检测异常轨迹。在上述过程中， Traj2Vec 采用交叉熵损失（crossentropy-loss）优化分类器:

$$L_{\mathrm{CE}} = \sum_T \sum_i p(i) \cdot \log\left(\frac{1}{q^T(i;\Theta)}\right) \tag{5.6}$$

其中$p(i)$是异常的概率分布，i 在{0,1}中取值，表示轨迹为异常或正常。另外，$q(i)$是经过 softmax 后输出的轨迹为异常的概率分布，Θ为分类器的参数。参数更新过程采用随机梯度下降算法优化分类器参数。

$$\Theta = \Theta + \lambda \cdot \frac{\partial L_{\mathrm{CE}}}{\partial \Theta} \tag{5.7}$$

其中λ为学习率。

5.3　性能评估

基于仿真轨迹数据和真实轨迹数据，本实验从无监督学习和有监督异常检测两个方面出发，综合评估 Traj2Vec 的性能。对于无监督学习部分，本文通过轨迹聚类评价轨迹自编码器产生的表示向量的性能。对于有监督学习部分，本文通过异常检测任务评估异常检测的准确率和对异常的召回率。

5.3.1　实验设置

本节首先介绍了用于评测 Traj2Vec 方法的数据集，然后介绍了对比方法及评价指标，最后介绍了实验中用的到的参数设置。

5.3.1.1　实验数据

轨迹聚类中，本文基于仿真轨迹数据集和真实船舶 AIS 轨迹数据集进行实验。仿真数据方面，我们仿真了 9000 条轨迹，其中包含 3 种基本移动方式——直线、绕圈、拐弯，6 种组合模式——直线+绕圈、直线+拐弯、绕圈+拐弯、绕圈+直线、拐弯+直线、拐弯+绕圈。每种模式有 1000 条轨迹。其中，每条轨迹的采样频率与时间长度从 2500 秒到 5000 秒内随机选取并在轨迹生成过程中加入高斯噪声。部分仿真数据如图 5.10 所示。真实数据方面，本实验基于真实的船舶 AIS 数据测试聚类效果。AIS 数据集包含中国近海及内河的 200 只船舶的轨迹，其中包括客船、货船、油船和渔船各 50 只。由于真实船舶轨迹没有异常标签，本文仅将其应用于聚类任务的性能评估。

对于轨迹异常检测任务来说，由于真实的轨迹异常数据很难获得，本文仅利用仿真数据集上进行了实验，验证异常检测算法的效果。与聚类相似，异常检测实验也仿真了 9000 条轨迹作为正常移动模式的轨迹数据。异常轨迹方面，本实验在与正常轨迹相同空间区域内利用 random walk 仿真了 1000 条轨迹作为异常轨迹。实验过程中，我们从 10000 条仿真轨迹中随机抽取其中 500 条作为有标签的训练数据，6500 条作为无标签的训练数据，剩余 3000 条作为测试数据。

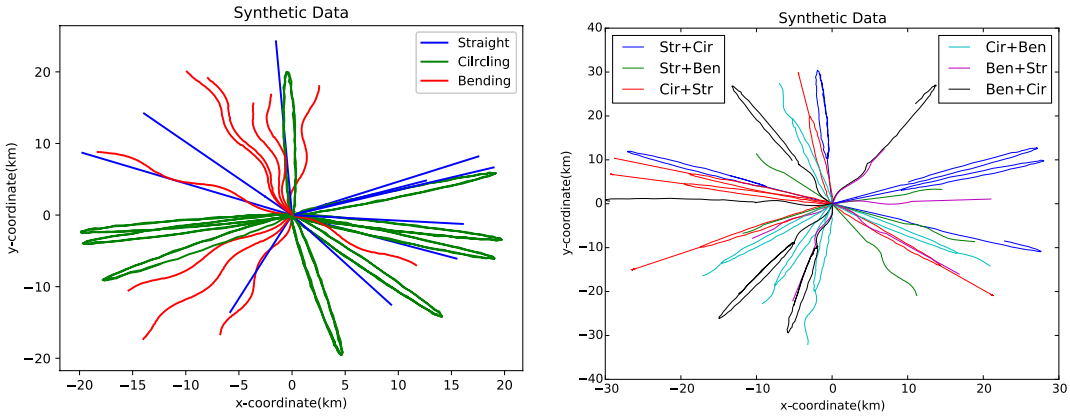

图 5. 10 合成数据中的部分基本模式和组合模式

Figure 5. 10 Parts of the synthetic trajectories with basic& combine movement patterns

5.3.1.2 对比方法及评价指标

轨迹聚类任务中，Traj2Vec 比较了基于 LCSS、DTW、EDR、Hausdorff 的轨迹聚类方法。本实验中所有方法都选择 K-Medoids 进行聚类。由于仿真数据集中行为模式的个数已知，所以对于仿真数据的聚类数量被设为 9。对于真实数据来说，我们通过调整聚类的数量并分析相应的结果选择最佳的 K 值。

评价指标方面，本文采用精确率（precision）、召回率（recall）和综合准确率（accuracy）来评价聚类结果。对于每种聚类方法，首先需要匹配聚类结果与九种移动模式之间的对应关系。本实验选取综合准确率最高的匹配方式作为最佳匹配，后续基于最佳匹配评估方法性能。针对每类移动模式，其精确率与召回率的定义如下：

$$p = \frac{TP}{TP+FP};\ r = \frac{TP}{TP+FN} \tag{5.8}$$

TP 表示匹配该移动模式的簇中的正确的轨迹数量；FP 表示匹配该移动模式的簇中的不正确的轨迹数量；FN表示应该在该簇中但未在的轨迹的数量。最后，计算

每个方法的综合准确率，即$Acc_{cluster}$ = *Sum of All TPs / Number of Trajectories*。

轨迹异常检测任务中，本文将 Traj2Vec 与基于轨迹统计特征利的分类算法进行了对比。对比方法包括支持向量机 SVM、多层感知机 MLP、决策树 DT 和提升方法 Ada。评价指标方面，本实验采用异常检测结果的精确率、召回率和异常检测准确率来评价实验结果。假设测试数据集中有 N 条轨迹，其中有 k 条为异常轨迹。异常检测方法对 N 条轨迹进行异常检测后，将检测结果中正常轨迹检测为正常轨迹的数目为N_{TP}，将异常轨迹检测为异常轨迹的数目N_{TN}；将正常轨迹检测为异常轨迹的数目为N_{FP}；将异常轨迹检测为异常轨迹的数目为N_{FN}。那么正常轨迹的精确率和召回率定义为：

$$p_{normal} = \frac{N_{TP}}{N_{TP}+N_{FP}} \; ; r_{normal} = \frac{N_{TP}}{N-k} \tag{5.9}$$

异常轨迹的精确率和召回率定义为：

$$p_{anomaly} = \frac{N_{TN}}{N_{TN}+N_{FN}} \; ; r_{anomaly} = \frac{N_{TN}}{k} \tag{5.10}$$

异常检测准确率的定义为：

$$Acc_{anomaly} = \frac{N_{TP}+N_{TN}}{N} \tag{5.11}$$

5.3.1.3　参数设置

实验采用 TensorFlow 的框架实现 Traj2Vec。所有的实验都是在具有 Intel Xeon CPU 2.10GHz 的服务器中进行。Traj2Vec 有三个主要的参数：（1）控制参数更新的学习率α；（2）控制轨迹表示向量大小的隐含状态尺寸 m；（3）训练次数 n；经过测试，参数值设置如下：在 LSTM 中，$\alpha = 0.00001, m = 250, n = 400$；在 GRU 中，$\alpha = 0.00001, m = 100, n = 800$；。在轨迹移动行为抽取模块中，滑动窗口被设为 600 秒，窗口的偏移量设为 300 秒。此外，EDR 与 LCSS 需要一个距离阈值来判断两条记录是否匹配。经过调参，阈值被设定为 100 米。

5.3.2　实验结果分析

本节分别从仿真数据聚类结果、真实数据聚类结果、仿真数据异常检测和参数敏感度分析四个方面分析了 Traj2Vec 中无监督学习部分和有监督异常检测部分的性能。

5.3.2.1 仿真数据聚类结果分析

本实验在仿真数据集上，利用不同的相似性度量，计算轨迹对之间的相似度，并基于 K-Medoids 生成聚类结果，不同方法得到的聚类结果如表 5.1 所示。

表 5. 1 合成数据集上的聚类结果

Table 5. 1 Clustering performance on synthetic data

	EDR	LCSS	DTW	Hausdorff	GRU-s2s	LSTM-s2s
Straight	0.465/0.563	0.460/0.411	0.411/0.613	0.423/0.263	0.643/0.723	**0.760/0.703**
Circling	0.550/0.482	0.610/0.643	0.540/0.462	0.415/0.531	**0.768/0.756**	0.766/0.823
Bending	0.668/0.678	0.621/0.392	0.472/0.322	0.465/0.379	**0.733/0.546**	0.652/0.752
S+C	0.359/0.468	0.573/0.523	0.503/0.474	0.507/0.414	0.571/0.684	**0.596/0.410**
S+B	0.453/0.427	0.462/0.574	0.507/0.746	0.435/0.510	0.646/0.823	**0.783/0.763**
C+B	0.600/0.581	0.469/0.313	0.766/0.480	0.389/0.283	0.563/0.522	**0.738/0.685**
C+S	0.470/0.434	0.388/0.661	0.595/0.377	0.348/0.429	**0.664/0.312**	0.621/0.891
B+S	0.327/0.374	0.409/0.582	0.769/0.379	0.375/0.534	**0.609/0.927**	0.500/0.316
B+C	0.674/0.419	0.528/0.251	0.525/0.879	0.442/0.387	**0.715/0.539**	0.688/0.819
$Acc_{cluster}$	49.18%	48.33%	53.69%	41.64%	64.80%	**68.47%**

本表中每个单元中的两个数值分别代表了精确率/召回率。结果显示，对比基于轨迹相似度的聚类方法（如 EDR、LCSS、Hausdorff 和 DTW），本文提出的基于轨迹自编码的聚类方法（GRU-s2s 和 LSTM-s2s）在所有移动模式上的评价指标的提升均超过 10%。这表示 Traj2Vec 的无监督学习部分学习到的轨迹表示向量可以捕捉轨迹的移动行为模式。

5.3.2.2 真实数据聚类结果分析

基于 AIS 船舶数据，本文分别进行两个实验。第一个实验是轨迹聚类，该实验首先利用 Traj2Vec 无监督学习部分生成的轨迹表示向量对真实轨迹进行聚类，然后通过观察部分聚类结果直观地分析 Traj2Vec 聚合移动行为相似轨迹的能力。第二个任务是船舶类型分析，该实验通过测试相同类型船舶的轨迹是否被聚类到同一个簇中，从而评价 Traj2Vec 抽取真实轨迹移动行为模式的效果。

轨迹聚类任务中轨迹编码模型选用 LSTM 作为循环神经网络单元。由于真实轨迹数据中聚类簇的数目未知，本实验首先通过 ELBOW 方法选择聚类的数

目，即 K 的值。我们将K从 3 增加到 100 生成聚类结果。对于每个K值，计算所有轨迹表示向量到它最近的簇中心的距离之和，记为E_K。结果如图 5.11 所示，连接所有由 K 和E_K组成的样本点得到了一个“肘”状曲线，该曲线中的拐点即为 K 的最佳值。在我们的数据集中，K 的值被设定为 33。

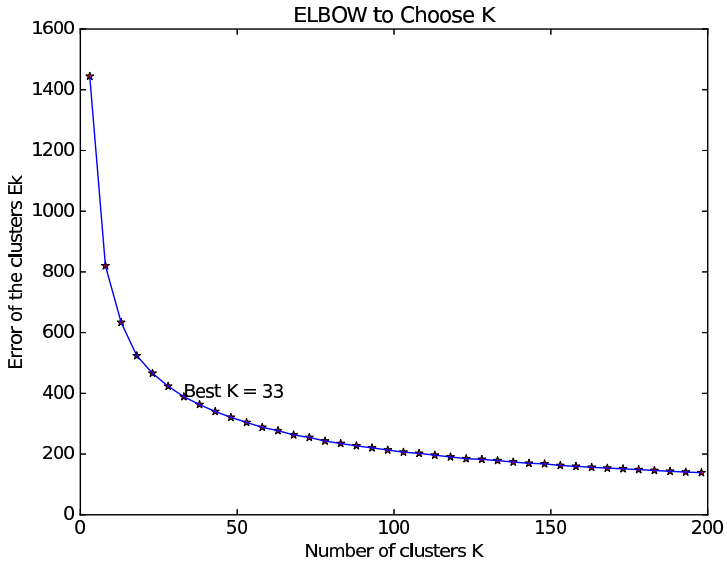

图 5. 11 利用 ELBOW 方法选择 K 值

Figure 5. 11 ELBOW method to choose K

由于船舶 AIS 数据中没有可用于比较的轨迹移动模式分布的真实情况，该实验无法采用与仿真数据相同的方法进行评估。我们仅能通过展示部分示例直观地评价 Traj2Vec 的效果。部分聚类结果在图 5.12、5.13 中展示，图中的蓝线代表了轨迹，黄点为轨迹起始点，红点为轨迹终止点。可以看出图 5.12 中的大多轨迹分布在旅游景点周围。这些轨迹大多都是短距离的往返轨迹。图 5.13 中的大多数轨迹是由内陆河中的货船产生的，这些轨迹大多是长轨迹且具有记录稀疏的特点。除上述两个簇外，其他簇中的轨迹移动行为模式也相似，由于篇幅限制本文不作展示。直观来看，Traj2Vec 无监督学习部分生成的表示向量可以将移动行为相似的轨迹聚类到同一簇中。

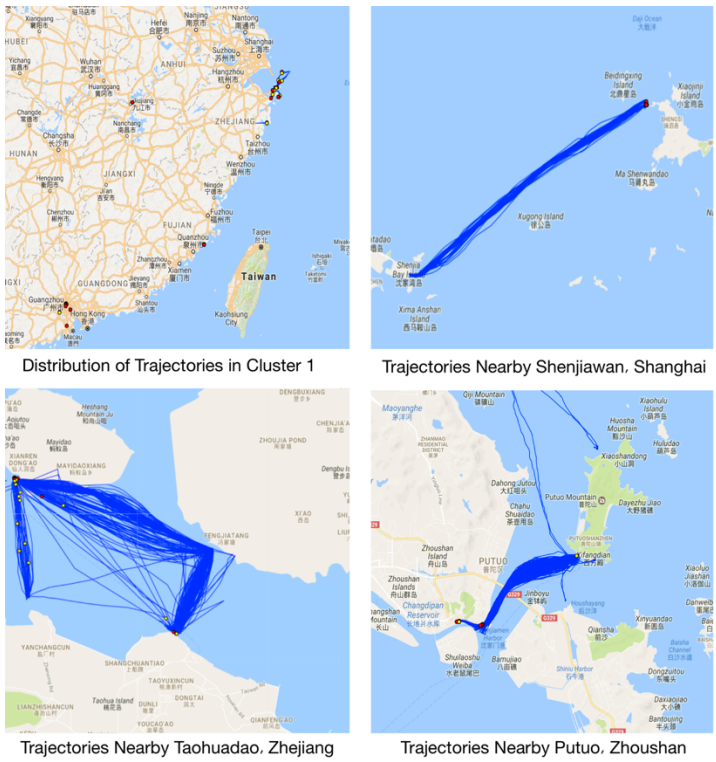

图 5.12 聚类 1 中的轨迹

Figure 5. 12 Trajectories in Cluster 1

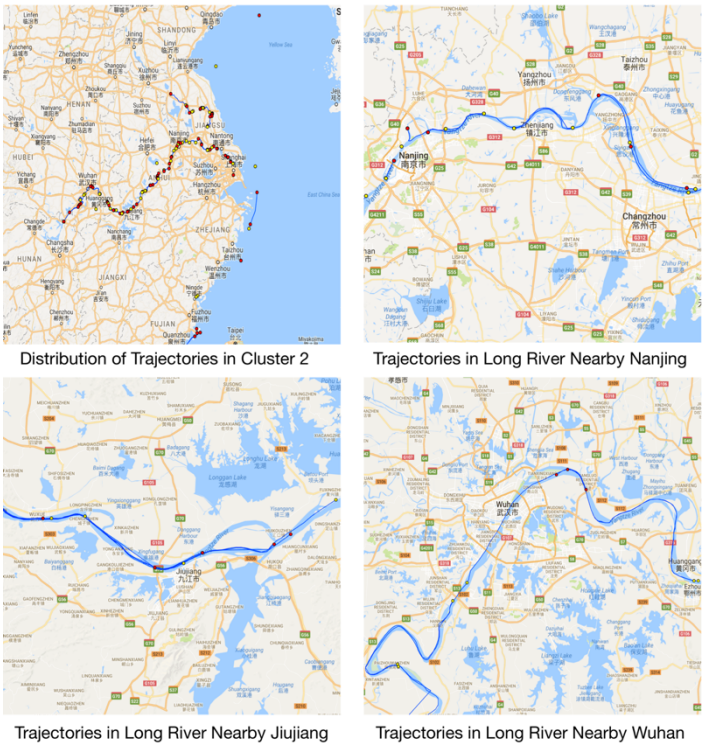

图 5.13 聚类 2 中的轨迹

Figure 5. 13 Trajectories in Cluster 2

船舶类型分析实验中，通过将轨迹表示向量相加可以得到船舶表示向量。本实验基于船舶表示向量的将船舶聚为 4 个簇，并利用聚类结果分析船舶类型。理想情况下，在不同聚类簇中的船舶向量对应的船舶类型也不同。该实验的结果如表 5.2 所示。

表 5. 2 船舶类型聚类结果

Table 5. 2 Vessel Type Clustering Results

	客船	渔船	货船	运油船
Total Number	50	50	50	50
Precision	46/53=0.87	38/44=0.86	23/37=0.62	50/66=0.76
Recall	46/50=0.92	38/50=0.76	23/50=0.46	50/50=1.0
$Acc_{cluster}$**: (46+38+23+50)/200 = 0.785**				

可以看出，尽管整个过程是无监督的，Traj2Vec 在该实验中仍取得了 78.5%的综合准确率。其中运油船和客船的准确率/召回率分别为 0.76/1.0 和 0.87/0.92。货船的结果相对较差，仅为 0.62/0.46。这是由于货船按运送货物类型不同移动行为的差别较大。如果可以对各类货船的子类别进行细分，聚类结果可能会有所提高。本实验再次表明，Traj2Vec 生成的轨迹表示向量可以捕捉真实轨迹中目标的移动行为特征。

5.3.2.3 仿真数据异常检测结果分析

表 5. 3 有监督轨迹异常检测结果

Table 5. 3 Results of semi-supervised trajectory anomaly detection

方法	p_{normal}	r_{normal}	$p_{anomaly}$	$r_{anomaly}$	$Acc_{anomaly}$
SVM	0.920	0.937	0.321	0.269	0.870
MLP	0.929	0.914	0.324	0.369	0.860
DT	0.944	0.903	0.374	0.519	0.865
Ada	0.957	0.959	0.622	0.605	0.924
T2V-RNN	0.962	0.970	0.709	0.657	0.939
Traj2Vec	**0.968**	**0.981**	**0.807**	**0.709**	**0.954**

轨迹异常检测算法的实验结果基于仿真数据集，结果如表 5.3 所示。本实验不仅对比了基于传统机器学习的轨迹异常检测方法如 SVM、MLP、DT 和 Ada，

还对比了 Traj2Vec 的变种方法 T2V-RNN。该方法直接基于无监督学习中的编码器利用有标签数据优化，没有使用基于分歧学习的方法优化。对比传统的机器学习方法，Traj2Vec 在异常检测准确率上稍有提升（从 0.924 到 0.954）的前提下将异常轨迹召回率提升超过 10%（从 0.605 到 0.709）。其主要原因是 Traj2Vec 可以通过协同训练从无标签轨迹数据中提取对异常检测有用的信息。实际应用中，异常轨迹往往需要保证漏报的情况尽可能少，因此异常轨迹召回率的提升对实际应用更有意义。另外，Traj2Vec 对比 T2V-RNN 在各个评价指标上均有提升。该现象说明基于分歧学习的方法有益于轨迹异常检测。

5.3.2.4 参数敏感度分析

本实验测试了不同学习率α和隐含状态大小m对实验结果的影响。训练次数设置为$n = 100$，测试参数组合：

$$\alpha = [0.00001, 0.00005, 0.0001, 0.0005]$$

$$m = [50, 100, 150, 200, 250, 300]$$

在对一个参数带来的影响进行研究时，其他的参数设为默认值，比较训练过程中的均方误差 MSE。

结果如图 5.14、5.15 所示。总的来说，随着 *m* 和 *n* 的增加，训练误差首先会下降，随后保持稳定。为了在精确度和准确率中达到平衡， LSTM 中的设置如下：$m = 250, n = 400, \alpha = 0.00001$，在 GRU 中做了如下设置：$m = 100, n = 100, \alpha = 0.00001$。可以看出，学习率$\alpha$对模型的最终结果非常重要。与深度学习方法在其他任务（如图像识别、自然语言处理）上的应用不同，Traj2Vec 更适用小的学习率。该现象可能是由于 Traj2Vec 损失曲线相对平缓，低的学习率可以让模型收敛到更优的结果。与较低的学习率相对应，模型优化时的迭代次数需要比其他任务更多。因此，本实验中的迭代次数均设置为 100 次。

随着隐含状态大小 m 的增加，模型的精度先增加后略微下降。这是由于当 m 较小（50）时，模型的表现能力不足，Traj2Vec 学习到的轨迹表示向量不能很好的重构移动行为序列。随着 m 的增加，训练模型所需的数据也随之增加。当 m 较大（300）时，用于训练的轨迹数据不足以充分优化 Traj2Vec 中的所有参数，导致模型精度略有下降。

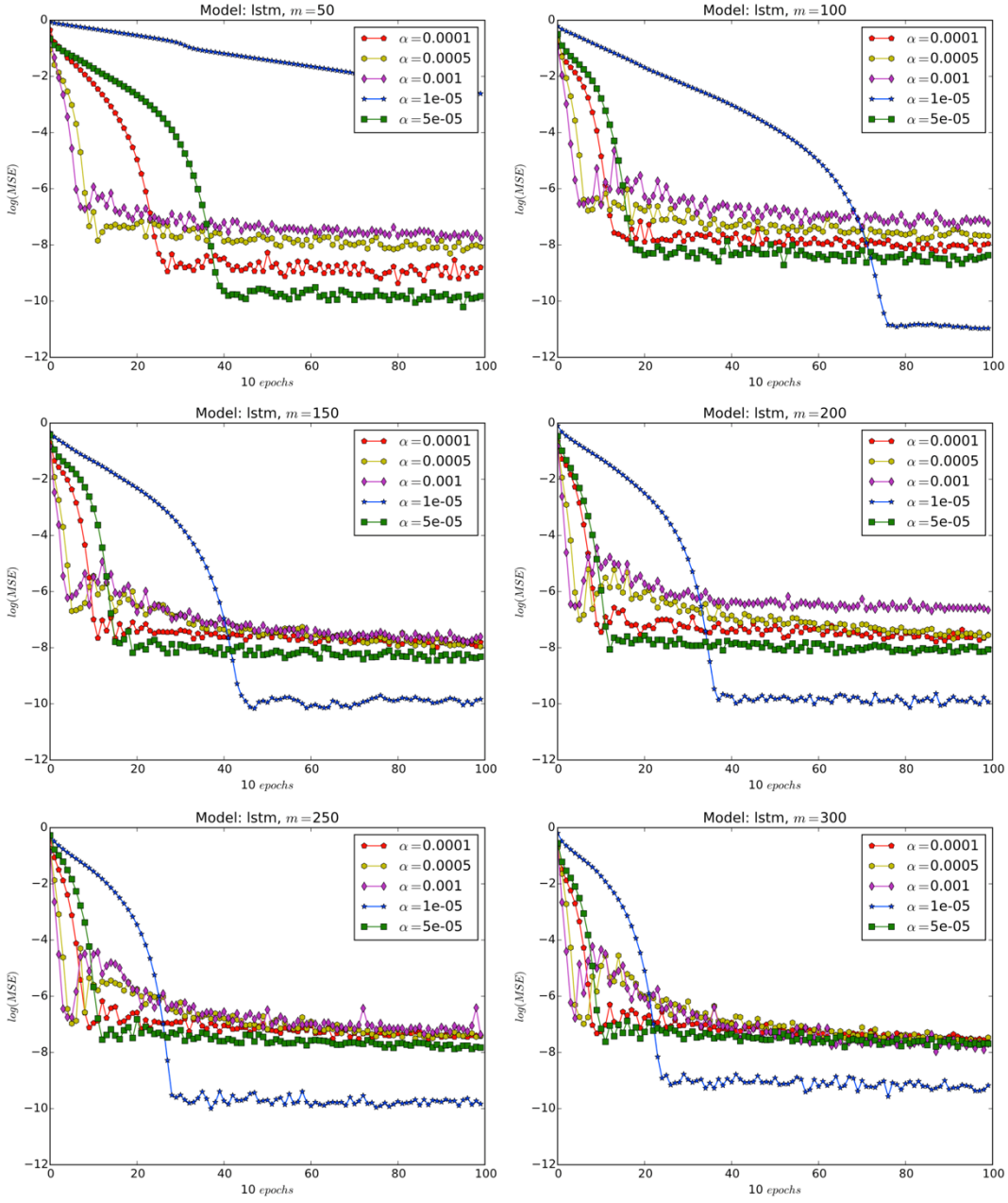

图 5. 14 LSTM 中*log* (*MSE*)在不同参数设置下的变化

Figure 5. 14 LSTM *log(MSE)* changes with different parameters

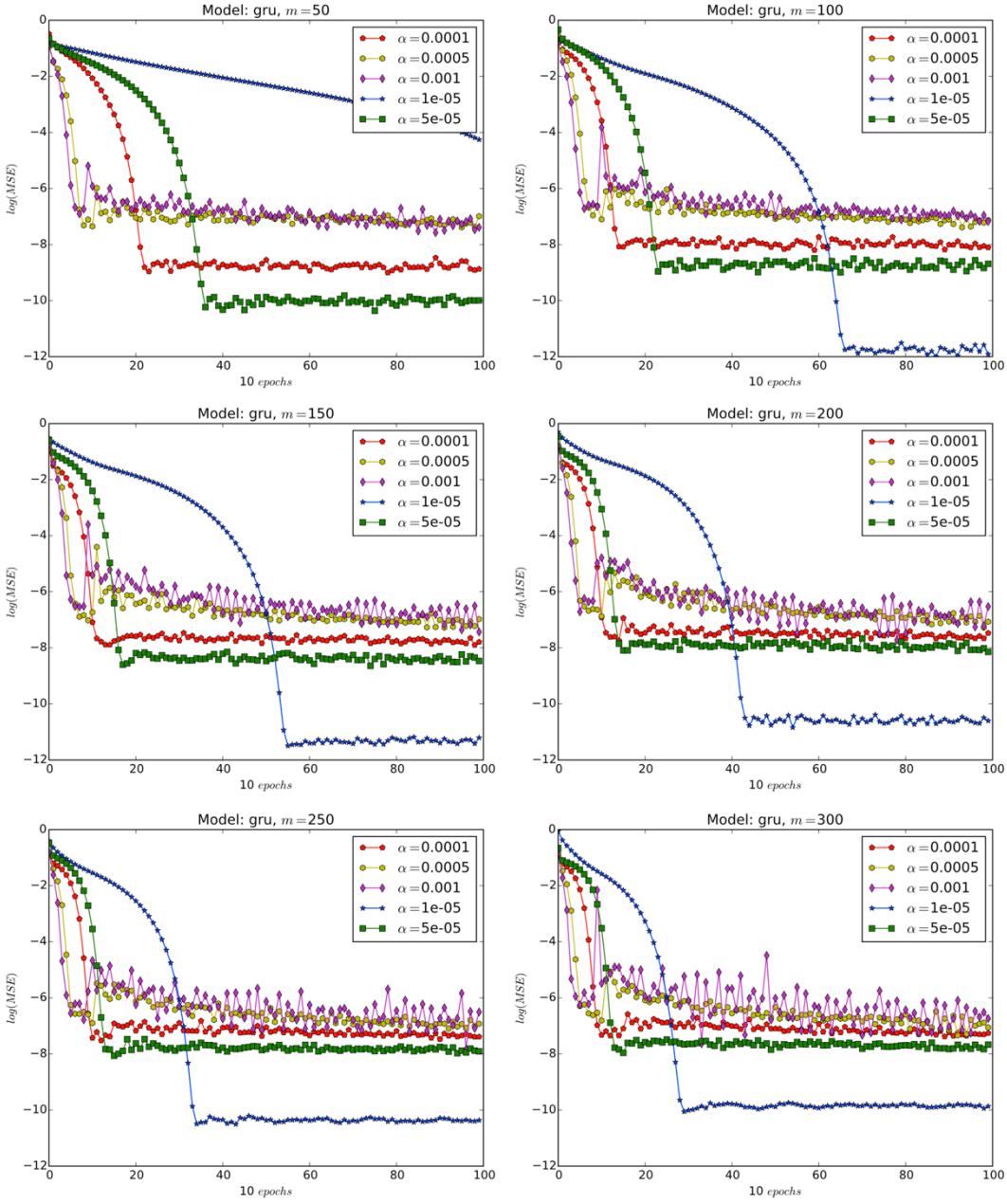

图 5. 15 GRU 中*log* (*MSE*)在不同参数设置下的变化

Figure 5. 15 GRU *log(MSE)* changes with different parameters

5.4 小结

本文提出了基于移动行为的半监督轨迹异常检测方法 Traj2Vec。该方法的核心步骤包括移动行为抽取、序列自编码和轨迹异常检测。其中基于滑动窗口的移动特征抽取算法能够提取轨迹的移动行为特征，解决了轨迹采样率不均的问题。此外，Traj2Vec 利用自动编码器学习轨迹移动行为序列的深度表示向量，并以此

为基础检测异常轨迹。本文在合成数据集和真实数据集中验证了 Traj2Vec 的有效性。实验结果表明，对比已有轨迹异常检测方法 Traj2Vec 可以获得更高的综合准确率和异常轨迹召回率。

第 6 章 结论与展望

6.1　论文主要贡献和创新

本文深入分析和探讨了轨迹数据挖掘背景、内容和研究现状。首先简要介绍了轨迹数据的定义、特点及分类；然后，介绍了轨迹数据挖掘的研究框架，并概述了主要的研究工作。最后，深入调研了已有研究在轨迹静默时间补全、轨迹相似度计算和轨迹异常检测上已经取得的成果，分析了这些成果的不足和挑战。

本文围绕稀疏轨迹补全、轨迹相似度计算和轨迹异常检测三个方面的问题。利用深度学习方法在隐含特征提取和复杂函数近似方面的优势，提出了基于上下文信息的稀疏轨迹补全方法、基于深度度量学习的轨迹相似度计算方法和基于移动行为特征的半监督轨迹异常检测方法，解决了轨迹数据挖掘面临的数据不确定、相似度计算复杂度高、异常标签少等问题。本文创新性工作主要体现在以下三个方面：

（1）针对稀疏轨迹补全的问题，提出了上下文信息感知的稀疏轨迹补全方法 TrajCom

TrajCom 从轨迹的上下文信息入手，自动地从历史轨迹中筛选出有用信息，补全目标轨迹缺失位置。TrajCom 首先利用目标移动的时空约束、目标偏好等信息，基于 NCF 筛选相关轨迹用于解决数据稀疏问题。然后，基于 RNN 设计轨迹编码模型并提出时间间隔感知的循环神经网络单元 tGRU，用于捕捉动态变化的时间间隔信息。最后利用注意力机制估计缺失位置。我们在四个真实数据集评估了 TrajCom 的性能，实验结果表明，与已有轨迹补全方法相比，TrajCom 在补全准确率上的提升幅度达到 25%。

（2）针对轨迹相似度计算复杂度高的问题，提出了基于深度度量学习的轨迹相似度计算方法 NeuTraj

NeuTraj 是一种基于深度度量学习加速轨迹相似度计算的方法，该方法属于近似算法的一种，可以将相似度计算复杂度降低到线性。给定一个轨迹数据库，NeuTraj 从数据库中采样种子轨迹集合后计算种子对的相似度，并将其作为监督

信息来训练基于深度神经网络的轨迹编码器。为了解决传统 RNN 不能建模序列间关系的问题，本文提出了空间注意力记忆机制用于增强现有循环神经网络单元。除此之外，本文提出了加权排序损失，使 NeuTraj 将优化的重点聚焦在更有区分力的轨迹对上。实验部分，本文使用两个真实的轨迹数据集评估了 NeuTraj 在四类相似性度量方法上的准确率和效率。实验结果表明 NeuTraj 在准确率和计算效率上均优于现有加速方法。对比已有近似算法，NeuTraj 的速度提升在 3 倍以上。

（3）针对轨迹异常检测中有标签数据稀少的问题，提出了基于移动行为特征的半监督轨迹异常检测方法 Traj2Vec

Traj2Vec 包括无监督学习和有监督学习两个阶段。在无监督学习阶段，Traj2Vec 基于轨迹自身移动行为特点，利用滑动窗口抽取轨迹移动行为并利用 RNN 序列自编码模型学习轨迹的表示向量，解决了轨迹采样率不均的问题。在有监督学习阶段，Traj2Vec 根据少量的数据标签基于分歧学习技术学习异常轨迹分类器。该阶段中 Traj2Vec 首先利用数据标签训练初始异常检测模型，然后利用无标签轨迹基于分歧学习的方法迭代地优化模型参数。实验结果表明，与已有轨迹异常检测方法相比，Traj2Vec 在异常检测准确率略有提高（3%）的前提下，将异常轨迹的召回率提高了 10%。

6.2 下一步研究工作展望

在现有工作基础上，作者计划在未来展开以下几方面的研究：

（1）时间约束下轨迹相似度计算方法研究

本文提出的 NeuTraj 仅考虑二维轨迹序列，计算轨迹在空间中的相似度。实际应用中，轨迹相似度计算不仅需要关注轨迹的空间属性，还需要考虑时间因素。在此场景下，直接利用已有的序列相似性度量方法计算经度、维度和时间三维属性上的相似度，存在计算开销高于二维轨迹的问题。因此，下一步工作中，将继续研究时间约束下的轨迹相似度定义，并提出新的模型结构，解决时间约束下轨迹相似度计算的加速问题。

（2）面向语义可解释的无监督轨迹异常发现及检测研究

目前，无监督下轨迹异常的发现及检测大多基于轨迹聚类将不能划分到聚类中的轨迹作为异常轨迹。该类方法无法从语义层面解释轨迹为什么是异常，而实际应用中，异常轨迹语义的可解释性对轨迹分析十分重要。本文所提出的 Traj2Vec 方法，可以在有少量数据标签的情况下检测异常轨迹，但不能解决无监督下的轨迹异常检测的可解释性。在下一步工作中我们将研究面向轨迹语义可解释的无监督轨迹异常检测，重点解决无监督地发现轨迹中可能为异常的轨迹，解释异常轨迹的语义。

（3）基于哈希编码的轨迹表示学习方法研究

基于深度学习模型学习轨迹的表示向量是一个基础且通用的技术。目前，基于 RNN、CNN 和 Autoencoder 的表示方法都是学习一个固定长度的实数向量作为轨迹的表示向量。在轨迹数据规模特别大的情况下，实数向量之间的计算开销也可能成为轨迹分析的瓶颈。而二进制哈希编码之间的计算，可以进一步利用硬件加速，从而适用于超大规模数据集。然而，相同长度的二进制哈希编码可蕴含的信息量少于实数向量，因此直接将实数向量编码转化为哈希编码会造成信息损失。在下一步工作中，需要重点研究如何学习哈希编码作为轨迹表示向量使信息损失最小。

参考文献

姚迪, 张超, 黄建辉,等. 时空数据语义理解：技术与应用[J].软件学报, 2018, 29(7):196-223.

朱燕, 李宏伟, 樊超,等. 基于聚类的出租车异常轨迹检测[J].计算机工程, 2017, 43(2):16-20.

毛嘉莉, 金澈清, 章志刚,等. 轨迹大数据异常检测:研究进展及系统框架[J].软件学报, 2017, 28(1):17-34.

高强, 张凤荔, 王瑞锦,等. 轨迹大数据:数据处理关键技术研究综述[J].软件学报, 2017, 28(4):959-992.

ABUL O, BONCHI F, NANNI M. Never walk alone: Uncertainty for anonymity in moving objects databases[C/OL]//Proceedings of the 24th International Conference on Data Engineering, ICDE 2008, April 7-12, 2008, Cancún, Mexico. 2008: 376-385. https://doi.org/10.1109/ICDE.2008. 4497446.

AGARWAL P K, FOX K, PAN J, et al. Approximating dynamic time warping and edit distance for a pair of point sequences[C/OL]//32nd International Symposium on Computational Geometry, SoCG 2016, June 14-18, 2016, Boston, MA, USA. 2016: 6:1-6:16. https://doi.org/10.4230/ LIPIcs.SoCG.2016.6.

ALT H, GODAU M. Computing the fréchet distance between two polygonal curves[J/OL]. Int. J. Comput. Geometry Appl., 1995, 5:75-91. https://doi.org/10.1142/S0218195995000064.

ARNAUD V, RODOLPHE D, ALDO N. A semi-supervised learning framework based on spatio- temporal semantic events for maritime anomaly detection and behavior analysis [C]//CoastGIS 2013-The 11th International Symposium for GIS and Computer Cartography for Coastal Zone Management. 2013: 4-pages.

ATEV S, MILLER G, PAPANIKOLOPOULOS N P. Clustering of vehicle trajectories[J/OL]. IEEE Trans. Intelligent Transportation Systems, 2010, 11(3):647-657. https://doi.org/10.1109/TITS. 2010.2048101.

B M R. A statistical analysis of mathematical measures for linear simplification[J]. The American Cartographer, 1986, 13(2):103-116.

BACKURS A, SIDIROPOULOS A. Constant-distortion embeddings of hausdorff metrics into constant-dimensional l_p spaces[C/OL]//Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2016, September

7-9, 2016, Paris, France. 2016: 1:1-1:15. https://doi.org/10.4230/LIPIcs.APPROX-RANDOM.2016.1.

BAO L, DU M. A distance-based trajectory outlier detection method on maritime traffic data[C]// 2018 4th International Conference on Control, Automation and Robotics (ICCAR). IEEE, 2018: 340-343.

BROMLEY J, GUYON I, LECUN Y, et al. Signature verification using a siamese time delay neural network[C/OL]//Advances in Neural Information Processing Systems 6, [7th NIPS Conference, Denver, Colorado, USA, 1993]. 1993: 737-744. http://papers.nips.cc/paper/769-signature-verification-using-a-siamese-time-delay-neural-network.

BU Y, CHEN L, FU A W, et al. Efficient anomaly monitoring over moving object trajectory streams[C/OL]//Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Paris, France, June 28 - July 1, 2009. 2009: 159-168. https://doi.org/10.1145/1557019.1557043.

CAO Z, QIN T, LIU T, et al. Learning to rank: from pairwise approach to listwise approach[C/OL]// Machine Learning, Proceedings of the Twenty-Fourth International Conference (ICML 2007), Corvallis, Oregon, USA, June 20-24, 2007. 2007: 129-136. https://doi.org/10.1145/1273496. 1273513.

CHANDLER J, OBERMAIER H, JOY K I. Interpolation-based pathline tracing in particle-based flow visualization[J/OL]. IEEE Trans. Vis. Comput. Graph., 2015, 21(1):68-80. https://doi.org/ 10.1109/TVCG.2014.2325043.

CHEN L, NG R T. On the marriage of lp-norms and edit distance[C/OL]//(e)Proceedings of the Thirtieth International Conference on Very Large Data Bases, Toronto, Canada, August 31 - September 3 2004. 2004: 792-803. http://www.vldb.org/conf/2004/RS21P2.PDF.

CHEN L, ÖZSU M T, ORIA V. Robust and fast similarity search for moving object trajectories [C/OL]//Proceedings of the ACM SIGMOD International Conference on Management of Data, Baltimore, Maryland, USA, June 14-16, 2005. 2005: 491-502. https://doi.org/10.1145/1066157. 1066213.

CHEN X, ZENG Y, CONG G, et al. On information coverage for location category based point-of-interest recommendation[C/OL]//Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA. 2015: 37-43. http://www.aaai.org/ ocs/index.php/AAAI/AAAI15/paper/view/9703.

CHEN Y, JIANG K, ZHENG Y, et al. Trajectory simplification method for location-based social networking services[C/OL]//Proceedings of the 2009 International Workshop on Location Based Social Networks, LBSN 2009, November 3, 2009, Seattle, Washington, USA, Proceedings. 2009: 33-40. https://doi.org/10.1145/1629890.1629898.

CHENG T, LI Z. A multiscale approach for spatio-temporal outlier detection[J/OL]. Trans. GIS, 2006, 10(2):253-263. https://doi.org/10.1111/j.1467-9671.2006.00256.x.

CHOW C, MOKBEL M F. Privacy of spatial trajectories[M/OL]//Computing with Spatial Trajectories. 2011: 109-141. https://doi.org/10.1007/978-1-4614-1629-6_4.

CIVILIS A, JENSEN C S, PAKALNIS S. Techniques for efficient road-network-based trac king of moving objects[J/OL]. IEEE Trans. Knowl. Data Eng., 2005, 17(5):698-712. h ttps://doi.org/10. 1109/TKDE.2005.80.

DRIEMEL A, SILVESTRI F. Locality-sensitive hashing of curves[C/OL]//33rd International Symposium on Computational Geometry, SoCG 2017, July 4-7, 2017, Brisbane, Aust ralia. 2017: 37:1-37:16. https://doi.org/10.4230/LIPIcs.SoCG.2017.37.

FARACH-COLTON M, INDYK P. Approximate nearest neighbor algorithms for hausdorff metrics via embeddings[C/OL]//40th Annual Symposium on Foundations of Computer Science, FOCS '99, 17-18 October, 1999, New York, NY, USA. 1999: 171-180. https: //doi.org/10.1109/SFFCS. 1999.814589.

FENG J, LI Y, ZHANG C, et al. Deepmove: Predicting human mobility with attentional r ecurrent networks[C/OL]//Proceedings of the 2018 World Wide Web Conference on W orld Wide Web, WWW 2018, Lyon, France, April 23-27, 2018. 2018: 1459-1468. htt ps://doi.org/10.1145/ 3178876.3186058.

FENG S, LI X, ZENG Y, et al. Personalized ranking metric embedding for next new POI recommendation[C/OL]//Proceedings of the Twenty-Fourth International Joint Conferen ce on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015. 2015: 2069-2075. http://ijcai.org/Abstract/15/293.

GE Y, XIONG H, LIU C, et al. A taxi driving fraud detection system[C/OL]//11th IEEE International Conference on Data Mining, ICDM 2011, Vancouver, BC, Canada, Dece mber 11-14, 2011. 2011: 181-190. https://doi.org/10.1109/ICDM.2011.18.

GIDÓFALVI G, HUANG X, PEDERSEN T B. Privacy-preserving data mining on moving object trajectories[C/OL]//8th International Conference on Mobile Data Management (MDM 2007), Mannheim, Germany, May 7-11, 2007. 2007: 60-68. https://doi.org/10.1

109/MDM.2007.18.

GONG X, XIONG Y, HUANG W, et al. Fast similarity search of multi-dimensional time series via segment rotation[C/OL]//Database Systems for Advanced Applications - 20th International Conference, DASFAA 2015, Hanoi, Vietnam, April 20-23, 2015, Proceedings, Part I. 2015: 108-124. https://doi.org/10.1007/978-3-319-18120-2_7.

GOWANLOCK M G, CASANOVA H. Distance threshold similarity searches: Efficient trajectory indexing on the GPU[J/OL]. IEEE Trans. Parallel Distrib. Syst., 2016, 27(9):2533-2545. https: //doi.org/10.1109/TPDS.2015.2500896.

GUO Y, XU Q, YANG Y, et al. Anomaly detection based on trajectory analysis using kernel density estimation and information bottleneck techniques[M]//Tech. Rep., Technical Report 108. University of Girona, 2014.

HAN Y, YAO J, LIN X, et al. GALLOP: global feature fused location prediction for different check- in scenarios[J/OL]. IEEE Trans. Knowl. Data Eng., 2017, 29(9):1874-1887. https://doi.org/10. 1109/TKDE.2017.2705083.

HOH B, GRUTESER M, XIONG H, et al. Achieving guaranteed anonymity in GPS traces via uncertainty-aware path cloaking[J/OL]. IEEE Trans. Mob. Comput., 2010, 9(8):1089-1107. https: //doi.org/10.1109/TMC.2010.62.

HSIEH H, LI C. Inferring online social ties from offline geographical activities[J/OL]. ACM TIST, 2019, 10(2):17:1-17:21. https://dl.acm.org/citation.cfm?id=3293319.

JENSEN C S. Review - r-trees: A dynamic index structure for spatial searching[J/OL]. ACM SIGMOD Digital Review, 1999, 1. db/journals/dr/Jensen99.html.

KEOGH E J, CHU S, HART D M, et al. An online algorithm for segmenting time series [C/OL]// Proceedings of the 2001 IEEE International Conference on Data Mining, 29 November 2 December 2001, San Jose, California, USA. 2001: 289-296. https://doi.org/10.1109/ICDM.2001. 989531.

KRUMM J, HORVITZ E. LOCADIO: inferring motion and location from wi-fi signal strengths [C/OL]//1st Annual International Conference on Mobile and Ubiquitous Systems (MobiQuitous 2004), Networking and Services, 22-25 August 2004, Cambridge, MA, USA. 2004: 4-13. https: //doi.org/10.1109/MOBIQ.2004.1331705.

KURBIEL T, KHALEGHIAN S. Training of deep neural networks based on distance measures using rmsprop[J/OL]. arXiv, 2017, abs/1708.01911. http://arxiv.org/abs/1708.01911.

KWAK H, LEE C, PARK H, et al. What is twitter, a social network or a news media? [C/OL]// Proceedings of the 19th International Conference on World Wide Web, WW W 2010, Raleigh, North Carolina, USA, April 26-30, 2010. 2010: 591-600. https://doi. org/10.1145/1772690. 1772751.

LAXHAMMAR R, FALKMAN G. Online detection of anomalous sub-trajectories: A slidin g win- dow approach based on conformal anomaly detection and local outlier factor[C /OL]//Artificial Intelligence Applications and Innovations - AIAI 2012 International W orkshops: AIAB, AIeIA, CISE, COPA, IIVC, ISQL, MHDW, and WADTMB, Halkidi ki, Greece, September 27-30, 2012, Proceedings, Part II. 2012: 192-202. https://doi.org /10.1007/978-3-642-33412-2_20.

LAXHAMMAR R, FALKMAN G. Online learning and sequential anomaly detection in tra jectories [J/OL]. IEEE Trans. Pattern Anal. Mach. Intell., 2014, 36(6):1158-1173. https: //doi.org/10.1109/ TPAMI.2013.172.

LEE J, HAN J, WHANG K. Trajectory clustering: a partition-and-group framework[C/OL]/ / Proceedings of the ACM SIGMOD International Conference on Management of Dat a, Beijing, China, June 12-14, 2007. 2007: 593-604. https://doi.org/10.1145/1247480.12 47546.

LEE J, HAN J, LI X. Trajectory outlier detection: A partition-and-detect framework[C/OL] // Proceedings of the 24th International Conference on Data Engineering, ICDE 2008, April 7-12, 2008, Cancún, Mexico. 2008: 140-149. https://doi.org/10.1109/ICDE.2008.4 497422.

LEI P. A framework for anomaly detection in maritime trajectory behavior[J/OL]. Knowl. Inf. Syst., 2016, 47(1):189-214. https://doi.org/10.1007/s10115-015-0845-4.

LI M, AHMED A, SMOLA A J. Inferring movement trajectories from GPS snippets[C/OL] // Proceedings of the Eighth ACM International Conference on Web Search and Data Mining, WSDM 2015, Shanghai, China, February 2-6, 2015. 2015a: 325-334. https://d oi.org/10.1145/2684822.2685313.

LI X, HAN J, KIM S. Motion-alert: Automatic anomaly detection in massive moving obje cts [C/OL]//Intelligence and Security Informatics, IEEE International Conference on Int elligence and Security Informatics, ISI 2006, San Diego, CA, USA, May 23-24, 2006, Proceedings. 2006: 166-177. https://doi.org/10.1007/11760146_15.

LI X, HAN J, LEE J, et al. Traffic density-based discovery of hot routes in road network

s [C/OL]//Advances in Spatial and Temporal Databases, 10th International Symposium, SSTD 2007, Boston, MA, USA, July 16-18, 2007, Proceedings. 2007: 441-459. https://doi.org/10. 1007/978-3-540-73540-3_25.

LI X, ZHAO K, CONG G, et al. Deep representation learning for trajectory similarity computation [C/OL]//34th IEEE International Conference on Data Engineering, ICDE 2018, Paris, France, April 16-19, 2018. 2018: 617-628. https://doi.org/10.1109/ICDE.2018.00062.

LI X, CONG G, LI X, et al. Rank-geofm: A ranking based geographical factorization method for point of interest recommendation[C/OL]//Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, Santiago, Chile, August 9- 13, 2015. 2015b: 433-442. https://doi.org/10.1145/2766462.2767722.

LI Y, LI Y, GUNOPULOS D, et al. Knowledge-based trajectory completion from sparse GPS samples[C/OL]//Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, GIS 2016, Burlingame, California, USA, October 31 - November 3, 2016. 2016: 33:1-33:10. https://doi.org/10.1145/2996913.2996924.

LI Z, DING B, HAN J, et al. Swarm: Mining relaxed temporal moving object clusters[J/OL]. PVLDB, 2010, 3(1):723-734. http://www.vldb.org/pvldb/vldb2010/pvldb_vol3/R65.pdf. DOI: 10.14778/1920841.1920934.

LI Z, DING B, HAN J, et al. Mining periodic behaviors for moving objects[C/OL]//Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, July 25-28, 2010. 2010b: 1099-1108. https://doi.org/10.1145/1835804. 1835942.

LI Z, WANG J, HAN J. Mining event periodicity from incomplete observations[C/OL]//The 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD ’12, Beijing, China, August 12-16, 2012. 2012: 444-452. https://doi.org/10.1145/2339530.2339604.

LIAO D, LIU W, ZHONG Y, et al. Predicting activity and location with multi-task context aware recurrent neural network[C/OL]//Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden. 2018: 3435- 3441. https://doi.org/10.24963/ijcai.2018/477.

LIAO L, PATTERSON D J, FOX D, et al. Learning and inferring transportation routines [J/OL]. Artif. Intell., 2007, 171(5-6):311-331. https://doi.org/10.1016/j.artint.2007.01.006.

LIU H, XU J, ZHENG K, et al. Semantic-aware query processing for activity trajectories [C/OL]// Proceedings of the Tenth ACM International Conference on Web Search and Data Mining, WSDM 2017, Cambridge, United Kingdom, February 6-10, 2017. 2017: 283-292. http: //dl.acm.org/citation.cfm?id=3018678.

LIU Q, WU S, WANG L, et al. Predicting the next location: A recurrent model with spat ial and temporal contexts[C/OL]//Proceedings of the Thirtieth AAAI Conference on Ar tificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA. 2016: 194-200. htt p://www.aaai.org/ocs/index. php/AAAI/AAAI16/paper/view/11900.

LIU S, NI L M, KRISHNAN R. Fraud detection from taxis' driving behaviors[J/OL]. IEE E Trans. Vehicular Technology, 2014, 63(1):464-472. https://doi.org/10.1109/TVT.2013. 2272792.

LONG J A. Kinematic interpolation of movement data[J/OL]. International Journal of Geo graphical Information Science, 2016, 30(5):854-868. https://doi.org/10.1080/13658816.20 15.1081909.

MANMATHA R, WU C, SMOLA A J, et al. Sampling matters in deep embedding learni ng[C/OL]// IEEE International Conference on Computer Vision, ICCV 2017, Venice, I taly, October 22-29, 2017. 2017: 2859-2867. https://doi.org/10.1109/ICCV.2017.309.

MATHEW W, RAPOSO R, MARTINS B. Predicting future locations with hidden markov models [C/OL]//The 2012 ACM Conference on Ubiquitous Computing, Ubicomp '12, Pittsburgh, PA, USA, September 5-8, 2012. 2012: 911-918. https://doi.org/10.1145/2370 216.2370421.

MCFEE B, LANCKRIET G R G. Metric learning to rank[C/OL]//Proceedings of the 27th International Conference on Machine Learning (ICML-10), June 21-24, 2010, Haifa, Is rael. 2010: 775-782. https://icml.cc/Conferences/2010/papers/504.pdf.

MERATNIA N, DE BY R A. Spatiotemporal compression techniques for moving point ob jects [C/OL]//Advances in Database Technology - EDBT 2004, 9th International Confe rence on Extending Database Technology, Heraklion, Crete, Greece, March 14-18, 200 4, Proceedings. 2004: 765-782. https://doi.org/10.1007/978-3-540-24741-8_44.

MIKOLOV T, KARAFIÁT M, BURGET L, et al. Recurrent neural network based languag e model[C/OL]//INTERSPEECH 2010, 11th Annual Conference of the International Sp

eech Com- munication Association, Makuhari, Chiba, Japan, September 26-30, 2010. 2 010: 1045-1048. http://www.isca-speech.org/archive/interspeech_2010/i10_1045.html.

MONREALE A, PINELLI F, TRASARTI R, et al. Wherenext: a location predictor on traj ectory pattern mining[C/OL]//Proceedings of the 15th ACM SIGKDD International Co nference on Knowledge Discovery and Data Mining, Paris, France, June 28 - July 1, 2009. 2009: 637-646. https://doi.org/10.1145/1557019.1557091.

MOREIRA-MATIAS L, GAMA J, FERREIRA M, et al. Time-evolving O-D matrix estima tion using high-speed GPS data streams[J/OL]. Expert Syst. Appl., 2016, 44:275-288. https://doi. org/10.1016/j.eswa.2015.08.048.

NERGIZ M E, ATZORI M, SAYGIN Y, et al. Towards trajectory anonymization: a gener alization- based approach[J/OL]. Trans. Data Privacy, 2009, 2(1):47-75. http://www.tdp. cat/issues/abs. a020a09.php.

PALLOTTA G, VESPE M, BRYAN K. Vessel pattern knowledge discovery from AIS dat a: framework for anomaly detection and route prediction[J/OL]. Entropy, 2013, 15(6):2 218-2245. https://doi.org/10.3390/e15062218.

PATTERSON D J, LIAO L, FOX D, et al. Inferring high-level behavior from low-level s ensors[C/OL]//UbiComp 2003: Ubiquitous Computing, 5th International Conference, Se attle, WA, USA, October 12-15, 2003, Proceedings. 2003: 73-89. https://doi.org/10.100 7/ 978-3-540-39653-6_6.

PFOSER D, JENSEN C S, THEODORIDIS Y. Novel approaches to the indexing of movi ng object trajectories[C/OL]//VLDB 2000, Proceedings of 26th International Conference on Very Large Data Bases, September 10-14, 2000, Cairo, Egypt. 2000: 395-406. htt p://www.vldb.org/conf/ 2000/P395.pdf.

PINK O, HUMMEL B. A statistical approach to map matching using road network geome try, Topology and vehicular motion constraints[C/OL]//11th International IEEE Confere nce on Intelligent Transportation Systems, ITSC 2008, Beijing, China, 12-15 October 2008. 2008: 862-867. https://doi.org/10.1109/ITSC.2008.4732697.

QUDDUS M A, NOLAND R B, OCHIENG W Y. A high accuracy fuzzy logic based ma p matching algorithm for road transport[J/OL]. J. Intellig. Transport. Systems, 2006, 1 0(3):103-115. https: //doi.org/10.1080/15472450600793560.

RAKTHANMANON T, CAMPANA B J L, MUEEN A, et al. Searching and mining trilli ons of time series subsequences under dynamic time warping[C/OL]//The 18th ACM

SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD ’ 12, Beijing, China, August 12-16, 2012. 2012: 262-270. https://doi.org/10.1145/2339530.2339576.

RENDLE S, FREUDENTHALER C, GANTNER Z, et al. BPR: bayesian personalized ranking from implicit feedback[C]//UAI 2009, Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, Montreal, QC, Canada, June 18-21, 2009. 2009: 452-461.

RICHTER K, SCHMID F, LAUBE P. Semantic trajectory compression: Representing urban movement in a nutshell[J/OL]. J. Spatial Information Science, 2012, 4(1):3-30. https://doi.org/10. 5311/JOSIS.2012.4.62.

S G J. Matching gps observations to locations on a digital map[C]//81th annual meeting of the transportation research board: volume 1. Washington, DC, 2002: 164-173.

SAKAKI T, OKAZAKI M, MATSUO Y. Earthquake shakes twitter users: real-time event detection by social sensors[C/OL]//Proceedings of the 19th International Conference on World Wide Web, WWW 2010, Raleigh, North Carolina, USA, April 26-30, 2010. 2010: 851-860. https://doi.org/ 10.1145/1772690.1772777.

SERRANO M E, GODOY S A, MONTOYA L Q, et al. Interpolation based controller for trajectory tracking in mobile robots[J/OL]. Journal of Intelligent and Robotic Systems, 2017, 86(3-4):569- 581. https://doi.org/10.1007/s10846-016-0422-4.

SHANG S, CHEN L, WEI Z, et al. Trajectory similarity join in spatial networks[J/OL]. PVLDB, 2017, 10(11):1178-1189. http://www.vldb.org/pvldb/vol10/p1178-shang.pdf. DOI: 10.14778/3137628.3137630.

SILLITO R R, FISHER R B. Semi-supervised learning for anomalous trajectory detection [C/OL]// Proceedings of the British Machine Vision Conference 2008, Leeds, UK, September 2008. 2008: 1-10. https://doi.org/10.5244/C.22.103.

SILVA F A, CELES C, BOUKERCHE A, et al. Filling the gaps of vehicular mobility traces[C/OL]// Proceedings of the 18th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems, MSWiM 2015, Cancun, Mexico, November 2-6, 2015. 2015: 47-54. https://doi.org/10.1145/2811587.2811612.

SOHN T, VARSHAVSKY A, LAMARCA A, et al. Mobility detection using everyday GSM traces [C/OL]//UbiComp 2006: Ubiquitous Computing, 8th International Conference, UbiComp 2006, Orange County, CA, USA, September 17-21, 2006. 2006: 212-224.

https://doi.org/10.1007/ 11853565_13.

SONG R, SUN W, ZHENG B, et al. PRESS: A novel framework of trajectory compressi on in road networks[J/OL]. PVLDB, 2014, 7(9):661-672. http://www.vldb.org/pvldb/vol 7/p661-song.pdf. DOI: 10.14778/2732939.2732940.

SPACCAPIETRA S, PARENT C, DAMIANI M L, et al. A conceptual view on trajectorie s[J/OL]. Data Knowledge Engineering, 2008, 65(1):126-146. https://doi.org/10.1016/j.dat ak.2007.10. 008.

SU H, ZHENG K, HUANG J, et al. Calibrating trajectory data for spatio-temporal similari ty analysis [J/OL]. VLDB J., 2015, 24(1):93-116. https://doi.org/10.1007/s00778-014-03 65-y.

TANG L A, ZHENG Y, XIE X, et al. Retrieving k-nearest neighboring trajectories by a s et of point locations[C/OL]//Advances in Spatial and Temporal Databases - 12th Inter national Symposium, SSTD 2011, Minneapolis, MN, USA, August 24-26, 2011, Proce edings. 2011: 223-241. https: //doi.org/10.1007/978-3-642-22922-0_14.

TANG L A, ZHENG Y, YUAN J, et al. On discovery of traveling companions from stre aming trajectories[C/OL]//IEEE 28th International Conference on Data Engineering (IC DE 2012), Washington, DC, USA (Arlington, Virginia), 1-5 April, 2012. 2012: 186-19 7. https://doi.org/10.1109/ ICDE.2012.33.

TERROVITIS M, MAMOULIS N. Privacy preservation in the publication of trajectories[C/ OL]// 9th International Conference on Mobile Data Management (MDM 2008), Beijin g, China, April 27-30, 2008. 2008: 65-72. https://doi.org/10.1109/MDM.2008.29.

VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need[C/OL]//Advanc es in Neural Information Processing Systems 30: Annual Conference on Neural Infor mation Pro- cessing Systems 2017, 4-9 December 2017, Long Beach, CA, USA. 2017: 6000-6010. http: //papers.nips.cc/paper/7181-attention-is-all-you-need.

VLACHOS M, GUNOPULOS D, KOLLIOS G. Discovering similar multidimensional trajec tories [C/OL]//Proceedings of the 18th International Conference on Data Engineering, San Jose, CA, USA, February 26 - March 1, 2002. 2002: 673-684. https://doi.org/10.1 109/ICDE.2002.994784.

WANG S, BAO Z, CULPEPPER J S, et al. Torch: A search engine for trajectory data[C/ OL]//The 41st International ACM SIGIR Conference on Research & Development in I nformation Retrieval, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018. 2018: 535

-544. https://doi.org/10.1145/ 3209978.3209989.

XIANYUAN Z, V U S, C R P S. Dynamics of functional failures and recovery in compl ex road networks[J]. Physical Review E, 2017, 96(5):052301.

XIAO X, ZHENG Y, LUO Q, et al. Finding similar users using category-based location h istory [C/OL]//18th ACM SIGSPATIAL International Symposium on Advances in Geo graphic Information Systems, ACM-GIS 2010, November 3-5, 2010, San Jose, CA, U SA, Proceedings. 2010: 442-445. https://doi.org/10.1145/1869790.1869857.

XIE D, LI F, PHILLIPS J M. Distributed trajectory similarity search[J/OL]. PVLDB, 2017, 10(11): 1478-1489. http://www.vldb.org/pvldb/vol10/p1478-xie.pdf. DOI: 10.14778/3137 628.3137655.

XUE A Y, ZHANG R, ZHENG Y, et al. Destination prediction by sub-trajectory synthesi s and privacy protection against such prediction[C/OL]//29th IEEE International Confer ence on Data Engineering, ICDE 2013, Brisbane, Australia, April 8-12, 2013. 2013: 2 54-265. https://doi.org/ 10.1109/ICDE.2013.6544830.

YAN Z, CHAKRABORTY D, PARENT C, et al. Semantic trajectories: Mobility data com putation and annotation[J/OL]. ACM TIST, 2013, 4(3):49:1-49:38. https://doi.org/10.114 5/2483669. 2483682.

YANG C, BAI L, ZHANG C, et al. Bridging collaborative filtering and semi-supervised l earning: A neural approach for POI recommendation[C/OL]//Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, August 13 - 17, 2017. 2017: 1245-1254. https://doi.org/10.1145/3 097983.3098094.

YAO D, ZHANG C, HUANG J, et al. SERM: A recurrent model for next location predic tion in semantic trajectories[C/OL]//Proceedings of the 2017 ACM on Conference on I nformation and Knowledge Management, CIKM 2017, Singapore, November 06 - 10, 2017. 2017: 2411-2414. https://doi.org/10.1145/3132847.3133056.

YI B, JAGADISH H V, FALOUTSOS C. Efficient retrieval of similar time sequences und er time warping[C/OL]//Proceedings of the Fourteenth International Conference on Dat a Engineering, Orlando, Florida, USA, February 23-27, 1998. 1998: 201-208. https://do i.org/10.1109/ICDE. 1998.655778.

YI X, ZHENG Y, ZHANG J, et al. ST-MVL: filling missing values in geo-sensory time series data[C/OL]//Proceedings of the Twenty-Fifth International Joint Conference on

Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016. 2016: 2704 -2710. http: //www.ijcai.org/Abstract/16/384.

YIN H, WOLFSON O. A weight-based map matching method in moving objects databases [C/OL]// Proceedings of the 16th International Conference on Scientific and Statistical Database Management (SSDBM 2004), 21-23 June 2004, Santorini Island, Greece. 200 4: 437-438. http://doi.ieeecomputersociety.org/10.1109/SSDBM.2004.10.

YU Y, CAO L, RUNDENSTEINER E A, et al. Detecting moving object outliers in massi ve-scale trajectory streams[C/OL]//The 20th ACM SIGKDD International Conference o n Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014. 2014: 422-431. https://doi.org/10.1145/2623330.2623735.

YUAN J, ZHENG Y, ZHANG C, et al. T-drive: driving directions based on taxi trajectori es[C/OL]// 18th ACM SIGSPATIAL International Symposium on Advances in Geogra phic Information Systems, ACM-GIS 2010, November 3-5, 2010, San Jose, CA, USA, Proceedings. 2010: 99-108. https://doi.org/10.1145/1869790.1869807.

YUAN N J, ZHENG Y, ZHANG L, et al. T-finder: A recommender system for finding p assengers and vacant taxis[J/OL]. IEEE Trans. Knowl. Data Eng., 2013, 25(10):2390-2 403. https://doi.org/ 10.1109/TKDE.2012.153.

ZEINALIPOUR-YAZTI D, LIN S, GUNOPULOS D. Distributed spatio-temporal similarity search [C/OL]//Proceedings of the 2006 ACM CIKM International Conference on Info rmation and Knowledge Management, Arlington, Virginia, USA, November 6-11, 2006. 2006: 14-23. https://doi.org/10.1145/1183614.1183621.

ZHANG W, WANG J. Location and time aware social collaborative retrieval for new suc cessive point-of-interest recommendation[C/OL]//Proceedings of the 24th ACM Internati onal Conference on Information and Knowledge Management, CIKM 2015, Melbourne, VIC, Australia, October 19 - 23, 2015. 2015: 1221-1230. https://doi.org/10.1145/2806 416.2806564.

ZHENG K, ZHENG Y, YUAN N J, et al. Online discovery of gathering patterns over tra jectories [J/OL]. IEEE Trans. Knowl. Data Eng., 2014, 26(8):1974-1988. https://doi.org /10.1109/TKDE. 2013.160.

ZHENG Y. Trajectory data mining: An overview[J/OL]. ACM TIST, 2015, 6(3):29:1-29:41. https: //doi.org/10.1145/2743025.

ZHENG Y, LI Q, CHEN Y, et al. Understanding mobility based on GPS data[C/OL]//Ubi

Comp 2008: Ubiquitous Computing, 10th International Conference, UbiComp 2008, Seoul, Korea, September 21-24, 2008, Proceedings. 2008a: 312-321. https://doi.org/10.1145/1409635.1409677.

ZHENG Y, LIU L, WANG L, et al. Learning transportation mode from raw gps data for geographic applications on the web[C/OL]//Proceedings of the 17th International Conference on World Wide Web, WWW 2008, Beijing, China, April 21-25, 2008. 2008b: 247-256. https://doi.org/10.1145/ 1367497.1367532.

ZHENG Y, LIU L, WANG L, et al. Learning transportation mode from raw gps data for geographic applications on the web[C/OL]//Proceedings of the 17th International Conference on World Wide Web, WWW 2008, Beijing, China, April 21-25, 2008. 2008c: 247-256. https://doi.org/10.1145/ 1367497.1367532.

ZHENG Y, XIE X, MA W. Geolife: A collaborative social networking service among user, location and trajectory[J/OL]. IEEE Data Eng. Bull., 2010, 33(2):32-39. http://sites.computer.org/debull/ A10june/geolife.pdf.

ZHENG Y, LIU F, HSIEH H. U-air: when urban air quality inference meets big data[C/OL]// The 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2013, Chicago, IL, USA, August 11-14, 2013. 2013: 1436-1444. https://doi.org/10.1145/ 2487575.2488188.

ZHU J, JIANG W, LIU A, et al. Time-dependent popular routes based trajectory outlier detection [C/OL]//Web Information Systems Engineering - WISE 2015 - 16th International Conference, Miami, FL, USA, November 1-3, 2015, Proceedings, Part I. 2015: 16-30. https://doi.org/10. 1007/978-3-319-26190-4_2.

ZHU Y, ZHENG Y, ZHANG L, et al. Inferring taxi status using GPS trajectories[J/OL]. CoRR, 2012, abs/1205.4378. http://arxiv.org/abs/1205.4378.

致 谢

回顾这六年的直博生涯，往事历历在目，本部分谨作为论文的补充来记录我在计算所的求学经历和心路历程。

首先，我要特别感谢我的导师毕经平老师！是她给了我宝贵的直博机会并为我的成长进步，悉心指导，不辞辛苦。至今还清楚地记得当年推免面试结束后，得到中科院计算所录取通知时的那份激动心情。六年的时光，在忙碌而充实的学习科研中不觉已经过去。毕老师勤奋上进、治学严谨的态度，让我记忆深刻，成为我今后学习的榜样；毕老师积极培养和激发我的科研兴趣，鼓励我选择感兴趣的科研方向；毕老师循循善诱和高屋建瓴的指导，给了我许多启发，帮助我踏实地走好科研道路上的每一步。

其次，感谢计算所的领导和老师！在计算所这个大家庭中，所领导的关怀和教育让我充满信心，大师的风采和指导让我受益匪浅，老师和同学的帮助让我非常感动。感谢周世佳老师、冯刚老师和研究生部其他老师给我在大小事上提供的支持和帮助。感谢联想公司贺志强老师设立的自强国际交流奖学金，使我获得在新加坡南洋理工留学的机会。感谢课题组的于金萍师姐、丁自旋师姐、谭海宁师弟、朱志华师弟和褚晓恺师弟，感谢他们对我博士研究工作的支持，陪我走过这一段难忘的求学经历。

我要感谢新加坡南洋理工大学丛高教授！2017 年 10 月我到南洋理工大学丛高教授课题组交流学习。一年短暂的留学生活使我收获了人生中一段重要的成长经历。在这个风景宜人的校园里，我接触到了许多优秀的同行学者，与来自不同国家的留学生一同成长。期间，丛高教授在科研上给予我精心指导，在生活上给予我很大帮助。沉稳、敏锐的丛老师帮助我从众多的想法中挖掘出有意义研究点，使我对科学研究有了更进一步的认识和思考。

我科研道路上还有一位重要的引路人，他是美国佐治亚理工大学计算机系助理教授张超博士。在我研究工作遇到瓶颈时，张超博士多次与我仔细讨论研究选题，认真帮助我梳理实验思路，耐心帮我修改论文初稿，大大增强了我在科研道

路上继续前行的信心。在此感谢张超博士对我的无私帮助！

我也要感谢父母对我的支持帮助，他们是我人生中的第一位导师。他们始终鼓励我调整好心态，踏实工作。同时，感谢女朋友六年来对我的鼓励、陪伴和对本论文的耐心校对。没有他们的付出和支持就没有我今天的收获。

科研工作有成功的喜悦，也通常会伴随诸多失败的压力和未知的焦虑。但我常常感觉自己是一个幸运的科研人员，我在探索真理的科研道路中，出现了许多像毕老师、丛高老师和张超博士这样一些业务能力出色、治学严谨和乐与分享的智者。他们让我认识到自己的不足，同时，也激发我前进的动力。在今后的工作中我会以一颗虔诚敬畏的心砥砺前行。

作者简历及攻读学位期间发表的学术论文与研究成果

作者简历：

2009 年 09 月——2013 年 07 月，在东北大学软件学院学习，获学士学位。

2013 年 09 月——2019 年 06 月，在中国科学院计算技术研究所攻读博士学位。

2017 年 10 月——2018 年 10 月，在新加坡南洋理工大学交流学习，联合培养。

获奖情况：

[1] 2018 年　博士国家奖学金

[2] 2017 年　中科院计算所自强国际交流奖学金

[3] 2015 年　中国科学院大学三好学生，优秀学生干部，海淀区优秀志愿者

[4] 2014 年　中国科学院大学三好学生，优秀学生干部

已发表（或正式接受）的学术论文：

[1] **Di Yao,** Gao Cong, Chao Zhang and Jingping Bi. Computing Trajectory Similarity in Linear Time: A Generic Seed-Guided Neural Metric Learning Approach, (**ICDE)** 2019.

[2] Xiaokai Chu, Xinxin Fan, **Di Yao**, Zhihua Zhu, Jianhui Huang and Jingping Bi, Cross-Network Embedding for Multi-Network Alignment, ACM the World Wide Web Conference (**ACM The WebConf**) 2019.

[3] **姚迪，**张超，黄建辉， 陈越新，毕经平；时空数据语义理解: 技术与应用; **软件学报** 2018 年 07 期

[4] Zhihua Zhu, **Di Yao**, Jianhui Huang, Hanqiang Li and Jingping Bi. Sub-trajectory-andTrajectory-Neighbor-based Outlier Detection over Trajectory Streams[C], Pacific-Asia Conference on Knowledge Discovery and Data Mining (**PAKDD)** 2018.

[5] **Di Yao,** Chao Zhang, Zhihua Zhu, Qin Hu, Zheng Wang, Jianhui Huang and Jingping Bi. Learning Deep Representation for Trajectory Clustering[J]. **Expert Systems**, 2018, 35(2), e12252

[6] **Di Yao,** Chao Zhang, Jianhui Huang and Jingping Bi, SERM: A Recurrent Model for Next

Location Prediction in Semantic Trajectories ACM on Conference on Information and Knowledge Management **(ACM CIKM)**, Singapore, Singapore, Nov 2017. Short Paper

[7] **Di Yao,** Chao Zhang, Zhihua Zhu, Jianhui Huang and Jingping Bi. Trajectory Clustering via Deep Representation Learning IEEE International Joint Conference on Neural Networks **(IEEE IJCNN)**, pp. 3880-3887 Anchorage, USA, May 2017.

[8] **Di Yao**, Jingping Bi, Jianhui Huang and Jin Zhu. A word distributed representation based framework for large-scale short text classification. IEEE International Joint Conference on Neural Networks (**IEEE IJCNN**), Killarney, Ireland, July, 2015.

[9] Xiaokai Chu, Xinxin Fan, **Di Yao,** Chenlin Zhang, Jianhui Huang and Jingping Bi. Noise-Aware Network Embedding for Multiplex Network. IEEE International Joint Conference on Neural Networks (**IEEE IJCNN**), Budapest, Hungary, July, 2019, in press.

已投稿论文：

[1] **Di Yao**, Kaiqi Zhao, Chao Zhang, Jin Yao Chin, Gao Cong and Jingping Bi. Trajectory Completion via Context-Guided Neural Encoding and Filtering, Submitted to **ICDE 2020**.

[2] **Di Yao**, Gao Cong, Chao Zhang and Jingping Bi. A Linear Time Approach to Computing Time Series Similarity based on Deep Metric Learning, Submitted to **IEEE Transactions on Knowledge and Data Engineering (TKDE).**

参加的研究项目：

[1] 2016.5-2017.10：基于 AIS 的异常检测技术研究二期（预研项目），技术负责人

[2] 2015.4-2017.10：卫星大数据监控及目标识别系统（预研项目），技术负责人

[3] 2015.1-2016.4：基于 AIS 的行为挖掘技术研究（预研项目），技术负责人

[4] 2013.9-2014.12： 云计算数据中心的网络优化核心技术（中石油合作项目），技术骨干、核心开发人员