摘 要

随着科技和移动设备的发展，生活中的轨迹数据不断增多，本文希望基于个体轨迹数据，设计有效的数据挖掘方法，实现对群体未来分布的预测。本文首先提出了一种轨迹聚类算法，以把相似的轨迹聚成一簇。它首先用JS散度来刻画轨迹之间的相似度（距离），再基于这种相似度度量对轨迹进行K-Means聚类。之后，本文提出了一种轨迹预算法，用于预测每条轨迹下一步可能到达位置的概率分布。它在预测某一条轨迹时，同时参考了它本身的历史轨迹数据和它所属簇内的其他轨迹数据，并基于马尔可夫链思想计算出两个转移矩阵，再将这两个矩阵按权相加即得到个体轨迹预测结果。两个转移矩阵的权重利用多元线性回归和梯度下降法学习得到。最终将所有个体轨迹预测结果整合起来即可得到群体分布预测结果。本文最后利用北京市出租车数据对算法进行实验，实验结果表明利用此算法得出的预测结果基本符合实际。

**关键词：**轨迹，数据挖掘，相似度度量，聚类，预测

ABSTRACT

With the development of science and mobile device technology, the trajectory data in our life is coming more and more. This paper aims to design an effective data mining method based on individual trajectory data, and realize the prediction of the distribution of population in the future. In this paper, we propose a trajectory clustering algorithm, which is used to group similar trajectories into cluster. JS divergence is used to describe the similarity(distance) between two trajectories, and based on which K-Means clustering is performed. After that, this paper presents a trajectory prediction method, which is used to predict the probability distribution of the next step of each trajectory. While making prediction of each trajectory, it refers to both its own historical trajectory data and the trajectory data in the same cluster, and based on the idea of Markov chain to calculate two transition matrices. Then the two matrices are summed as the result of individual trajectories prediction. The weights of the two transition matrices are obtained using multiple linear regression and gradient descent method. Finally, all individual trajectory prediction results can be integrated to get the population distribution prediction results. In the end, this paper uses the taxi data of Beijing for experiment, the experimental results show that the prediction results obtained by using this algorithm is basically in line with the actual results.

**Keywords:** trajectory, data mining, similarity, clustering, prediction

目 录

[第1章 引言 1](#_Toc5875)

[1.1 绪论 1](#_Toc31588)

[1.2 研究背景及相关工作回顾 2](#_Toc14833)

[1.2.1 轨迹数据及其研究意义 2](#_Toc11675)

[1.2.2 轨迹数据挖掘相关工作回顾 3](#_Toc11228)

[1.3 本文研究内容与章节安排 4](#_Toc12233)

[第2章 轨迹聚类算法 5](#_Toc18594)

[2.1 聚类基本原理介绍 5](#_Toc28204)

[2.1.1 聚类分析基本流程 5](#_Toc21377)

[2.1.2 相似度度量 7](#_Toc28435)

[2.2 相关工作回顾 8](#_Toc4440)

[2.3 基于JS相似度的轨迹聚类算法 10](#_Toc1165)

[2.3.1 网格划分 11](#_Toc12818)

[2.3.2 基于JS散度的相似度度量算法 12](#_Toc5251)

[2.3.3 K-Means聚类算法 14](#_Toc27361)

[2.4 实验 17](#_Toc29351)

[第3章 轨迹预测算法 21](#_Toc28391)

[3.1 基本思想 21](#_Toc25313)

[3.2 马尔可夫链与转移矩阵 23](#_Toc8820)

[3.2.1 马尔可夫链原理介绍 23](#_Toc20514)

[3.2.2 转移矩阵 24](#_Toc1299)

[3.2.3 转移矩阵的求解 25](#_Toc25281)

[3.3 权重学习 27](#_Toc28904)

[3.3.1 回归分析 27](#_Toc15054)

[3.3.2 梯度下降法 28](#_Toc571)

[3.3.3 轨迹转移矩阵的权重学习 30](#_Toc30213)

[3.4 实验 32](#_Toc20516)

[第4章 实验与讨论 35](#_Toc9794)

[4.1 数据集说明 35](#_Toc10153)

[4.2 算法流程 35](#_Toc17571)

[4.3 结果及讨论 36](#_Toc17623)

[第5章 结语 39](#_Toc20163)

[5.1本文内容 39](#_Toc8604)

[5.2改进方向 40](#_Toc26281)

[参考文献 41](#_Toc28777)

[致 谢 43](#_Toc21351)

# 第1章 引言

伴随着社会的发展和进步，人类的生活中越来越多的数据被产生和记录下来，因此大家越来越重视对数据的分析。如何从大量数据中挖掘出有效信息并合理利用，成为了现代社会中一个热门研究领域。

## 1.1 绪论

为了有效分析和利用生活中产生的大量数据，一系列数据挖掘方法被提出，以帮助人们发现数据中的有用信息，从而更好得服务于人们的生活[1]。

在前人不断的探索和尝试中，一整套“从数据中学习知识”的流程逐渐成熟，并越来越广泛的应用于数据挖掘研究者的工作中，这一流程主要包括以下几个部分[2]：

1. **数据采集：**采集所需要的数据。

2. **数据清理：**对原始数据进行处理，消除噪声和不一致的数据等。

3. **数据选择：**从数据库中提取与分析任务相关的数据。

4. **数据变换：**把数据变换或统一成适合挖掘的形式，如通过汇总或特征选择。

5. **数据挖掘：**在这个过程中人们使用智能方法来发掘数据中的有用的信息，并提取数据模式，例如聚类、分类、关联规则分析等方法。

6. **结果解释与评估：**最后需要对得到的结果进行解释与评估，如果结果不够满意有可能需要重新进行挖掘。

其中，数据挖掘是最为关键的一部分，因此数据挖掘也常常用来泛指从数据中学习知识这一过程，本文中所涉及的很多算法就属于数据挖掘的范畴。**数据挖掘**所能实现的功能以及它可以发现的数据模式如下[3]：

1. **挖掘频繁模式、关联和相关：**频繁模式是在数据中频繁出现的模式，挖掘频繁模式可以发现数据中有趣的关联和相关，频繁项集挖掘是频繁模式挖掘的最简单形式，常见的频繁项集挖掘方法有Apriori算法等。

2. **分类和预测：**分类问题用于找出描述和区分数据类的模型或函数，以便使用模型来预测类标号未知的对象，如果函数是连续型的就被称为预测问题。常见的分类方法有决策树分类、贝叶斯分类和人工神经网络分类等。

3. **聚类分析：**不同于分类和预测问题，聚类分析的数据对象不考虑已知的类标号，它根据数据的特性将较相似的数据分在同一类，并且尽量使不同类之间相似度较小。常见的聚类方法有K-Means算法、DBSCAN算法和谱聚类方法等。

4. **离群点分析：**离群点分析用于发现数据中与数据的一般行为或模型不一致的离群点数据，虽然大部分数据挖掘方法会将离群点视为噪声或异常丢弃，但是在一些应用中这些罕见的事件可能比正常出现的事件更有研究价值。常见的离群点分析方法有OLAP方法等。

其中聚类和预测在很多领域中都有重要应用，本文也将围绕这两个研究方向提出相关算法，并将其应用于轨迹数据上，以达到研究目的。

## 1.2 研究背景及相关工作回顾

随着卫星、无线网络以及定位设备的发展，大量移动轨迹数据呈急速增长的趋势，如交通轨迹数据、动物迁徙数据、气候气流数据、人员移动数据等。

### 1.2.1 轨迹数据及其研究意义

轨迹数据就是时空环境下，通过对一个或多个移动对象运动过程的采样所获得的数据信息，包括采样点位置、采样时间等，这些采样点数据信息根据采样先后顺序构成了轨迹数据。例如具有定位功能的智能手机，轨迹数据反映了手机持有者某一时间段的行动状况，移动互联网络可以通过无线信号定位手机所在位置，进而采样记录，通过连接采样点形成手机持有者的运动轨迹数据。

一条轨迹数据一般可以表示为一系列点的序列（本文中的轨迹都是指二维空间中的轨迹），例如轨迹可以表示为：



其中，每个点包括一组地理坐标和一个时间标签。

通过轨迹数据挖掘发现隐含的知识，研究其行为模式并做出预测，可以帮助政府和用户做出更好的决策，甚至可以成为解决城市交通、城市环境、突发事件应急等重大社会问题的有效手段[4]。例如，地图管理人员可以通过分析轨迹及时发现新增的道路，从而以最快最有效的方式更新地图，使地图与现实中的路网相符；在交通方面，通过分析车辆的轨迹可以得知道路的拥堵情况，从而可以根据推测的拥堵情况来向车辆提示路况堵塞情况和最佳导航，方便城市交通的协调；另外，通过分析用户历史轨迹数据，还可以挖掘出人们之间的社交关系，从而为人们提供旅游、好友等推荐服务。

因此，近年来轨迹数据挖掘越来越受到各界的关注，包括计算机科学、社会学和地理学等在内的各个领域都将其列为重要研究课题。而移动对象的位置预测技术可以帮助提供更好的基于位置的服务， 有助于分析和理解轨迹数据，具有深远的意义和很大的发展空间。所以说，本文的研究具有重要的理论和现实意义。

### 1.2.2 轨迹数据挖掘相关工作回顾

由于轨迹数据巨大的研究价值，国内外现已对轨迹数据做了一些相关研究，主要包括轨迹压缩、轨迹检索、轨迹共同移动模式和轨迹聚类等[5]。

轨迹压缩旨在提取轨迹中较为重要的点保留下来而抛弃其他不太重要的点，从而降低轨迹数据的存储量和计算量。轨迹压缩中比较常见的为Douglas-Peucker算法和Bellman算法，Douglas-Peucker算法[6]基于递归思想不断选出最大误差点作为分割点，并重复此过程，直到误差值小于给定值为止；Bellman算法[7]则是利用动态规划的思想，寻找使得误差值全局最小的最优解。

轨迹检索旨在用尽量少的时间检索并获取所需要的轨迹。由于轨迹数据的主要信息为空间和时间信息，故根据空间和时间的索引顺序可以把轨迹检索算法分为三类。一是空间和时间同时检索的方法，它把时间看作地图的第三维，索引结构可以使用3D-Rtree等；二是先检索时间再检索空间的方法，先把时间划分为若干时间间隔，再对每个时间间隔建立空间索引，索引结构可以使用Rt-Tree, HR-Tree或H+R-Tree等；三是先检索空间再检索时间的方法，先把空间分为多个网格，再对网格使用时间索引，其中网格索引可以使用CSE-Tree结构。

轨迹共同移动模式旨在寻找一系列轨迹对象中有共同移动行为的轨迹。现有的轨迹共同移动模式挖掘方法一般都要求一组对象以某种模式相聚，并维持一定时间，根据相聚模式的不同和维持的时间是否需要连续，研究者定义了一系列共同移动模式，如群(Flock)，护送者(Convoy)，蜂群(Swarm)，旅行伴侣(Traveling companion)和集会(Gathering)等。

轨迹聚类旨在找到相似的轨迹或轨迹段，并把它们聚为一类。现有的轨迹聚类方法有特征向量法、基于期望最大的轨迹聚类方法、轨迹线段聚类方法和增量式聚类算法等。在本文第2章的2.2小节中将会详细介绍这些轨迹聚类算法。

## 1.3 本文研究内容与章节安排

本文的主要研究内容是从一系列个体轨迹数据中提取有效信息，设计数据挖掘方法，实现对它们未来群体分布的预测。这与传统的轨迹预测方法有所不同，传统的轨迹预测一般是利用个体轨迹历史数据预测个体的未来轨迹，或利用群体分布历史数据预测群体的未来分布。而本文的研究思路是希望将两者结合起来，先基于个体轨迹数据预测出每个个体下一步可能到达位置的概率分布，再将个体预测结果整合起来，从而得到群体分布的预测结果。

基于以上研究内容和研究思路，可以看出本文的研究核心在于如何可靠得对个体轨迹进行预测，于是本文提出了个体轨迹预测的中心思想——当预测某个个体轨迹时，不但可以参考其自身的历史轨迹数据，还可以参考与其相似的轨迹数据。围绕这一中心思想，本文首先提出了轨迹聚类算法，以找到相似的轨迹并将它们聚为一类，形成一个簇。接下来又提出了轨迹预测算法，基于轨迹聚类结果，利用个体轨迹自身历史数据和同簇轨迹数据对每条轨迹进行预测，并将所有个体轨迹预测结果整合得到群体分布预测结果。最后，利用北京市出租车数据对算法进行实验。

论文具体章节安排如下：

1. 主要介绍了轨迹数据挖掘的研究背景、研究意义及相关工作回顾，并介绍了文章的组织结构。
2. 提出了一个轨迹聚类算法，包括该聚类算法中所需用到的相似度度量方法和K-Means聚类方法，并在出租车数据集上分析聚类结果。
3. 提出了一个轨迹预测算法，包括个体轨迹预测时会涉及的转移矩阵学习和权重学习，以及个体轨迹预测结果的整合。
4. 将把本文第二章和第三章提出的算法应用于出租车数据集，按整个算法流程进行实验，完成对出租车未来群体分布的预测。
5. 将对整篇文章进行总结和分析，并以此为基础对接下来的工作进行展望。

# 第2章 轨迹聚类算法

聚类分析是数据挖掘领域一项非常重要的方法，在许多领域都有重要应用[8]，包括市场研究、模式识别、数据分析和文档处理等。在生物学中，聚类能用来推导植物和动物的分类，根据相似功能对基因进行分类，获得对种群中固有结构的认识；在商务中，聚类能帮助市场分析人员根据购买模式从顾客中发现不同的顾客群，刻画顾客群的特征，以便商场向合适的人群推荐商品；聚类还可以帮助识别卫星观测数据中土地利用相似的区域，根据房子的类型和地理位置等对房屋的分组进行识别；聚类还能用于对Web上的文档进行分类，以发现各种信息等等。在本文中，则是通过对轨迹的聚类来寻找相似的轨迹，以便于进行后续的轨迹预测。

## 2.1 聚类基本原理介绍

随着大量轨迹数据不断被记录、保存下来，越来越多的数据挖掘手段逐渐应用在轨迹数据上，轨迹聚类便是聚类分析在轨迹数据上的一种应用，它可以帮助人们找到较为相似的轨迹，从而方便人们研究、分析不同轨迹簇的特征等。本节将会首先介绍聚类分析和其基本流程，之后再详细介绍聚类中的相似度度量。

### 2.1.1 聚类分析基本流程

聚类分析的目的是将相似的对象聚为一类，不相似的对象分到不同类，从而形成若干个“簇”，换句话说聚类要使同一个簇中的对象相似度尽量较高，而不同簇之间相似度尽量较低。一般来说，聚类分析的基本流程包括定义相似度、选择聚类算法和检验聚类结果等步骤。

**1. 定义相似度**

因为聚类是基于相似度进行的算法，所以为了衡量对象之间的相似度，首先要定义数据集中各个对象之间的相似度（或距离），两个对象之间的相似度越高（即距离越近），它们就越有可能被分在同一类中。具体的相似度度量方法将会在之后的2.1.2节中具体提到。

**2. 选择聚类算法**

在定义或得到对象之间的相似度后，就需要使用合适的聚类算法对这些对象进行聚类，常见的聚类算法有K-Means算法，DBSCAN算法，OPTICS算法和谱聚类算法等。

根据上文提到的数据类型的不同，聚类算法一般可以分为两类。第一类是适用于向量空间上的聚类算法，例如DBSCAN算法和OPTICS算法等，这类算法的研究对象需是可以用多维空间向量表示的数据。另一类是应用于测度空间上的聚类算法，适用于那些无法用向量空间表示的数据，例如K-Means算法和谱聚类算法等，只要提供对象之间的相似度，即可使用这类聚类算法将对象分为不同的类。显然，第二类算法也可以应用于第一类所适用的向量空间，因为上文中已经提到我们很容易在向量空间上定义对象之间的相似度（距离）。

本文中需要处理的数据类型为轨迹数据，故第二类聚类算法将比较适用。

**3. 检验聚类结果**

在进行完聚类后，如果有条件还可以进行对结果的检验。对于有标签数据（即已知数据的真实分类），常见的检验指标有NMI和ARI指标等。

假设数据集中有*N*个对象，它们的原始真实分类标签分别为（其中表示第*i*个对象的类别），在使用聚类算法进行聚类后得到的分类标签为，则该聚类结果的NMI指标为：

 （2-1）

其中









以上就是聚类分析的基本流程，在本文中重点将会放在相似度的定义和聚类分析算法上，因为轨迹数据集不存在原始的真实分类标号，所以无法进行量化得检验，只能通过实验来进行大致的检验。

由于轨迹数据的特殊性，对轨迹数据进行聚类的方法可能与一般聚类方法有所不同，故本文将在2.2节对轨迹聚类的相关工作进行回顾。

### 2.1.2 相似度度量

相似度度量又称距离度量，两个对象之间的相似度越高则距离越近，反之相似度越低则距离越远。衡量对象之间的相似度是一项非常重要的工作，相似度度量在许多问题中都有相关的应用[9]，其中一个很重要的应用就是在上文提到的聚类分析中。下面就简单介绍一下相似度度量（距离度量）的一般方法。

相似度度量方法的选择和数据类型有很大的关系，有些数据可用*n*维空间中的向量表示，对这类数据通常可以直接使用*n*维空间中的距离函数来定义对象之间的距离，常见的距离函数有：欧几里得距离、曼哈顿距离和闵可夫斯基距离等。

在这些距离函数中，最常用的是欧几里得距离（又称欧式距离），若距离函数选用的是欧几里得距离，那么对于给定数据集中的两个*n*维向量和，它们之间的距离可以表示为：

 （2-2）

但是对一些较为复杂的数据类型，并不能直接用*n*为空间中的向量表示，因此这些对象的相似度度量需要另辟他径，寻找更适合的方法，而不能直接简单得用这里的距离函数表示。

比如在日常生活中，就可能常常面对的是更加复杂的数据类型，如时序数据或网络数据等。且随着卫星、无线网络以及定位设备的发展，大量移动轨迹数据呈急速增长的趋势，也使得越来越多的轨迹数据被记录、保存下来，挖掘这些轨迹数据中隐含的有效信息有着非常重大的意义，因此，定义一个能有效度量轨迹数据相似度的指标成为了大家关注的课题。本文将在后文中回顾轨迹相似度度量的相关工作并提出本文的相似度度量方法。

## 2.2 相关工作回顾

大量轨迹数据的出现使数据挖掘方法有了新的应用领域，因此，一系列基于轨迹数据的聚类方法和相似度度量方法被陆续提出，本节将做一个简单的相关工作回顾。

目前，轨迹聚类方法主要可以分类两大类，一类是将整条轨迹当作对象进行聚类，这些聚类方法的主要区别在于相似度的度量方式不同，这里相似度度量方法包括轨迹间欧氏距离、DTW距离、最长公共子序列距离和编辑距离；另一类是将每条轨迹划分为很多轨迹段，然后对轨迹段进行聚类的方法。

1. **基于轨迹间欧氏距离的聚类**

轨迹间欧氏距离[10]首先需要计算每个时刻上对应两点的欧氏距离，再对这些距离求和即可。假设记录的点数为*n*个，待进行相似度度量的两条轨迹分别为*R*和*S*，则对于轨迹*R*来说，第*i*时刻的对应点坐标为，它们的轨迹间欧氏距离为

 （2-3）

1. **基于DTW距离的聚类**

DTW(Dynamic Time Warping)全称为动态时间规整[11]，常被用来度量时间序列的距离，由于轨迹也可以看作一串时间序列，故DTW也可以应用于衡量轨迹间距离。由于时间序列的长度往往不同，无法做到点与点之间一一配对，故DTW的基本思想是对时间序列做一定的“缩放”，从而找到两个序列间最适合的匹配。

假设两个时间序列*R*和*S*的长度分别为*n*和*m*，则它们可以分别表示为和，则它们之间的DTW距离可以表示为：

（2-4）

其中，表示轨迹*R*和*S*当前第一个轨迹点间的欧氏距离，和分别表示轨迹*R*和*S*去掉当前第一个点后剩余未匹配的轨迹点。从式中可以看出，如果两条轨迹都无剩余轨迹点，则表示匹配完成，DTW距离为0；如果只有一条轨迹无剩余轨迹点而另一条有，则DTW距离为无穷大；如果两条轨迹都存在剩余轨迹点，则需用递归的方式求取最小的距离作为DTW距离。

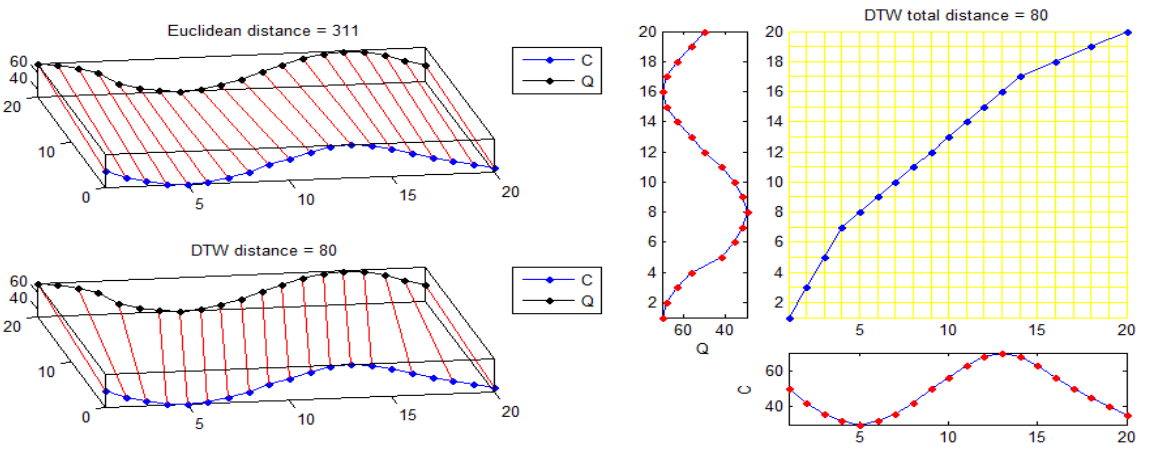


图2-1 DTW距离示意图[12]

1. **基于最长公共子序列的聚类**

在现实情况中，由于硬件设备的限制可能使轨迹数据的采样值中存在一定噪声，即与真实值之间存在偏差。为了避免对轨迹进行相似度度量时对噪声太过敏感，基于最长公共子序列（LCSS）的度量方法[13]被提出，它使用连个序列中都出现的最长公共子序列来度量这两个序列的相似度。

最长公共子序列的计算方法可以用下式表示：

 （2-5）

其中，是一个相似阈值，只要两个点的差值小于这个阈值即可认为这对记录点相似。

但是对于不同的轨迹来说，它们的长度不同，故如果直接用LCSS的结果当作相似度并不合理，因此还需要将它标准化，转化为可以进行相似度度量的形式：

 （2-6）

这个过程可以将LCSS转化为[0,1]区间内的值，其值越小则说明两轨迹间越相似。

**4. 基于编辑距离的聚类**

编辑距离最早用于衡量两个字符串之间的相似度[14]，它是指两个字符串在进行比较时，只进行增加、删除或替换操作，使一个字符串变成另一个字符串时所需的最小操作次数。后来编辑距离渐渐也被应用在时间序列和轨迹数据中。

以“123345”和“8233756”这两个字符串为例，它需要经过3次操作才能使一个字符串变成另一个，故它的编辑距离为3，具体得操作步骤为：

a. **1**23345 → **8**23345（“1”替换成“8”）

b. 8233**4**5 → 8233**7**5（“4”替换成“7”）

c. 823375 → 823375**6**（在最后增加“6”）

1. **轨迹段聚类法**

轨迹段聚类法[15]采用的是先划分再聚类的方法，它先把所有轨迹都划分成若干轨迹段，再用衡量线段距离的方法对这些轨迹段进行聚类，最终得到轨迹段的运动模式。在衡量轨迹段间的距离时，主要考虑三种距离并把它们加权求和从而得到轨迹段间距离，这三种距离分别为垂直距离、平行距离和角度距离。

以上就是对轨迹聚类相关工作的回顾，由于本文的预测是要基于个体轨迹的，故划分轨迹段一类的聚类方法显然不太合适，因此本文的做法是定义一种轨迹间的相似度度量，再用合适的聚类算法进行聚类。

## 2.3 基于JS相似度的轨迹聚类算法

本文进行轨迹聚类的目的是为之后的轨迹预测奠定基础，而通过上文中对轨迹聚类相关工作的回顾，显然可见划分轨迹段一类的聚类方法并不太合适，因为划分轨迹后无法保留每个个体轨迹的特征， 因此本文的做法与第一类轨迹聚类方法类似，首先定义一种轨迹间相似度度量方法，再选用合适的聚类算法进行聚类。

考虑到我们希望在预测某个个体时，与它同簇的轨迹数据也能提供有效的参考信息，故同簇的轨迹数据应该尽量具有相似的移动模式，且分布情况尽量相似，故本文选取的相似度度量方法是基于空间分布相似度的。本节将具体介绍这种空间分布相似度的衡量指标，并介绍选用的聚类算法。

### 2.3.1 网格划分

下文中用*T*表示一条轨迹，用表示第*i*条轨迹。其中，每一条轨迹又可以用一系列点表示，由于本文中不涉及轨迹点的具体时间，故忽略时间参数，因此一条轨迹可以表示为

 （2-7）

其中*n*为该轨迹中包含的点的个数。

为了更好得统计轨迹的空间分布和更方便于之后的轨迹聚类与预测，首先要对平面空间进行网格化，即把连续的平面空间划分为网格状的区域，这样就可以把原本轨迹中的一系列坐标序列转换为一系列轨迹经过的区域的序列，如下图2-2所示。假设网格的宽为，长为，一行的网格数为，则网格编号可以用以下公式计算：

 （2-8）

其中代表网格在*x*轴上的编号，代表网格在*y*轴上的编号：

，

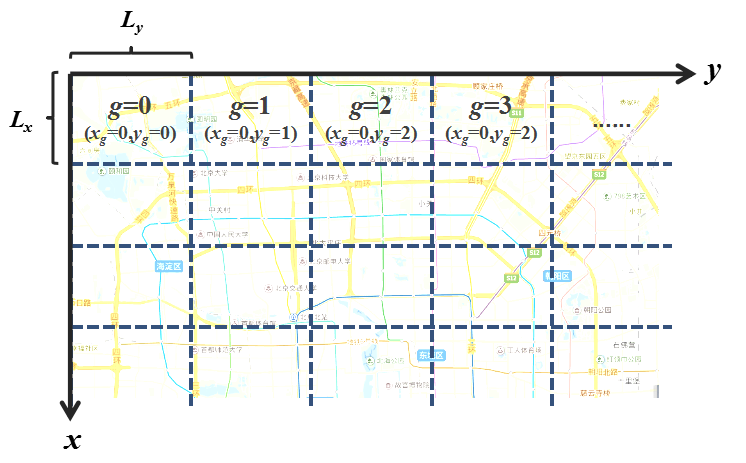


图2-2 网格划分示意图

对平面空间的网格化从另一角度来说相当于把空间精度进行降低，因为具体的坐标点是很难重复出现的，这不便于之后空间分布的统计和轨迹的预测，所以这么做可以便于之后做聚类和预测。

### 2.3.2 基于JS散度的相似度度量算法

#### 2.3.2.1 KL散度

根据香农的信息论，给定一个字符集的概率分布，我们可以设计一种编码，使得表示该字符集组成的字符串平均需要的比特数最少。假设这个字符集是*X*，其概率分布为，且对，其出现概率为，那么其最优编码平均需要的比特数等于这个字符集的熵

 （2-9）

在同样的字符集上，假设存在另一个概率分布。如果用概率分布的最优编码（即字符*x*的编码长度等于），来为符合分布的字符编码，那么表示这些字符就会比理想情况多用一些比特数。

因此，在统计学和概率论中，就常常用KL散度（Kullback–Leibler divergence）[16]来度量两个概率分布之间的差异。假设这两个概率分布分别为*P*和*Q*，若它们是连续随机变量，则分布*P*关于分布*Q*的KL散度为：

 （2-10）

如果*P*和*Q*为离散随机变量，则分布*P*关于分布*Q*的KL散度可以用下式表示：

 （2-11）

实际上，从编码的角度来说，分布*P*关于分布*Q*的KL散度可以理解为基于*Q*来编码来自*P*的样本时平均所需要的额外的位元数。

KL散度有以下特性：

1. **非负性**：KL散度值始终大于等于0，且由吉布斯不等式可以得出，当且仅当的时候KL散度等于0，且KL散度的值越大说明两个分布的相似度越低。该性质可以用下式表示：

 （2-12）

1. **非对称性**：尽管从直觉上KL散度是个[度量或距离函数](https://zh.wikipedia.org/wiki/%E5%BA%A6%E9%87%8F)，但是它实际上并不是一个真正的度量或距离，因为KL散度不具有对称性，即分布P关于分布Q的KL散度（或度量）一般不等于分布Q关于分布P的KL散度（或度量）。该性质可以用下式表示：

 （2-13）

举个实际的例子，假设有连个随机变量*X*和*Y*，它们的样本空间相同，且是离散随机变量，样本空间中可能的取值共有4个，则假如随机变量*X*对这四个取值的概率分布是0.1，0.2，0.3，0.4；随机变量*Y*对这四个取值的概率分布是0.4，0.3，0.2，0.1，那么分布*X*对于分布*Y*的KL散度为







#### 2.3.2.2 基于JS散度的相似度度量算法

本文希望选一种度量方式来对轨迹进行度量，发现轨迹的空间分布可以和概率中的分布进行类比，故可以考虑使用KL散度来对轨迹间的相似度进行度量。但是上节中提到，KL散度的性质中存在不对称性，因此它无法直接用来度量轨迹间的距离，会为之后的聚类带来一定困扰，可能会导致聚类效果不佳。

因此我们希望找到一种KL散度的改进算法，而有人提出可以使用JS散度（Jensen-Shannon divergence）[17]来改善KL散度不对称的问题。一般来说，*P*和*Q*的JS散度可以由以下公式来定义：

 （2-14）

其中，*M*表示*P*和*Q*分布的平均，可以用下式表示

 （2-15）

这样，就可以把KL散度转化为一个可以用于距离（相似性）度量的指标。JS散度有以下性质：

1. **有界性**：JS散度的值总大于等于0且小于等于1，当且仅当的时候JS散度等于0。该性质可以用下式表示：

 （2-16）

2. **对称性**：JS散度具有对称性，即分布P关于分布Q的JS散度（或度量）等于分布Q关于分布P的JS散度（或度量），这个性质对于相似度度量来说至关重要。该性质可以用下式表示：

 （2-17）

由于本文希望把JS散度应用在轨迹数据上，度量两个轨迹之间的相似度，那么就需要明确对于轨迹数据来说，样本空间和取值分别是什么。在上文的2.3.1节中，已经提到我们把空间划分为了一系列网格，因此，如果这一地理空间为样本空间的话，那么划分出来的每个网格就是可能的取值。那么由此，用JS散度来度量轨迹R和轨迹S之间的相似度就可以用下式来表示：

 （2-18）

其中

 （2-19）

 （2-20）

在式（2-19）中，*g*代表上文所提到的网格编号即对所有网格进行遍历，把对应值相加，的计算原理同式（2-19）。但是在计算过程中，可能会出现某网格*g*中或的情况，由于这种情况下式（2-19）是没有意义的，故若出现这种情况时令该网格的计算值等于0来处理。

这样，我们就定义了基于JS散度的相似度（距离）度量算法，两条轨迹之间的JS散度越小，就表示两条轨迹的相似度越大，当两条轨迹完全相同时JS散度等于0。

### 2.3.3 K-Means聚类算法

在用JS散度定义了轨迹间的相似度度量后，就可以选用合适的聚类算法来进行聚类。在上文2.1.1节中已经提到，聚类算法一般可以分为两类，第一类是适用于向量空间上的聚类算法，例如DBSCAN算法和OPTICS算法等，这类算法的研究对象需是可以用多维空间向量表示的数据；另一类是应用于测度空间上的聚类算法，适用于那些无法用向量空间表示的数据，例如K-Means算法和谱聚类算法等，只要提供对象之间的相似度，即可使用这类聚类算法将对象分为不同的类。

而由于轨迹数据不能在向量空间中表示，只能在测度空间中表示，故第二类聚类算法将比较适用，因此本文选用的聚类算法为比较常用的K-Means聚类算法，它是最为经典的基于划分的聚类方法，是十大经典数据挖掘算法之一。下面将首先对K-Means算法原理进行介绍，再说明如何利用K-Means算法对轨迹数据聚类。

#### 2.3.3.1 K-Means算法原理介绍

 K-Means算法首先需要用户指定*K*为参数，它可以将输入的*N*个数据对象划分为*K*个簇，使得所得到的簇满足同一个簇内的对象之间相似度尽量高，而不同的簇之间对象相似度尽量小。簇的相似度是利用簇中对象的均值来度量的，可以看作簇的聚类中心（或质心）。

K-Means算法的基本思想是，首先从*N*个数据对象任意选择 *K* 个对象作为初始聚类中心，而对于所剩下的其它对象，则根据它们与这些聚类中心的相似度（距离），分别将它们分配给与其最相似的簇（用聚类中心来度量）；然后再重新计算每个所获新簇的聚类中心（该簇中所有对象的均值），不断重复这一过程直到标准测度函数开始收敛为止，一般来说如果重新计算聚类中心时与上一次聚类相比无变化，则可以当作聚类结果已收敛。

K-Means算法的基本流程可以用以下几步描述：

（1）从*N*个数据对象随机选取*K*个对象作为初始的聚类中心；

（2）对于每个对象，计算它和*K*个簇的聚类中心的距离，并将它归到和它距离最小的簇中（距离最小说明它和这个簇的相似度最高）；

（3）重新计算每个簇的聚类中心（簇中所有对象的均值）；

（4）不断重复第2步和第3步，直到标准测度函数开始收敛为止（或直到每个聚类中心不再发生变化为止）。

以上就是K-Means算法的基本流程，上述的流程之后，就可以把原本的*N*个对象分为*K*个簇，每个簇是一个分类，簇内的对象相似度较高，簇之间的相似度较低。

#### 2.3.3.2 基于K-Means方法的轨迹数据聚类

上文介绍了K-Means算法的基本原理和流程，下面就要介绍如何把这个算法应用于轨迹数据。

一般的K-Means算法在重新确定聚类中心时，是直接计算簇内对象的均值，但是轨迹数据中因为没有定义“绝对距离”而只定义了轨迹对象之间的“相对距离”，故无法直接计算均值。所以在处理轨迹数据时，聚类中心定义为与簇内其他所有对象的距离之和最短的对象，即聚类中心被定义在了某一个具体的“对象”上。

在介绍轨迹数据对象上的K-Means方法前，首先需要定义一些数学表达，设需要进行聚类的轨迹对象集合为*D*，包含*N*个对象，聚类参数为*K*个，即需要聚为*K*个类。定义以下变量：

：第*t*次选出的第*k*类的聚类中心（, ）。

：第*t*次聚类的第*k*个簇中所包含的对象集合（, ）。

：轨迹对象*i*在第*t*次被分到的簇的编号（, ）。

：表示轨迹对象*i*和*j*之间的距离（），这里是沿用2.3.2节中提到的JSD散度当作轨迹之间的距离（相似度）度量。

故基于K-Means方法的轨迹聚类算法可以用以下流程表示：

1. **输入**：

聚类的簇数量*K*；

包含*N*个轨迹对象的数据集*D*。

2. **初始化**：

从轨迹对象中随机选定K个对象为聚类中心（）；

每个簇的集合中现在只包含自己的聚类中心，即：

  （2-21）

3. **核心算法**：

**Repeat**

（1）将每个对象归到和它距离最小的簇中：

  （2-22）

1. 重新计算每个簇的聚类中心：

  （2-23）

**Until**  

4. **输出**：*K*个簇的对象集合（）。

以上就是基于K-Means方法的轨迹聚类算法 ，通过该方法就可以把轨迹对象进行聚类，分为*K*个簇。

## 2.4 实验

在设计完轨迹相似度度量和轨迹聚类算法后，我们要对其进行实验验证，这里选取了25辆北京市出租车的轨迹数据，为了保证聚类的效果，这里选取的数据都在东经115.3~117.3和北纬39.3~41.3范围内的轨迹，且每条轨迹的采样点都不少于1000个。

首先利用2.3.1节中介绍的方法做准备工作，对平面二维空间做网格划分，这里把*x*轴和*y*轴都划分为100等份，则网格的总数量为

 （2-24）

之后，再根据2.3.2节中的公式（2-18）来度量轨迹之间的相似度，即计算每两个轨迹之间的JS散度，结果如表2-1所示。

表2-1 轨迹间JS散度实验结果

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | ... |
| 1 | 0.00 | 0.06 | 0.07 | 0.08 | 0.08 | 0.08 | ... |
| 2 | 0.06 | 0.00 | 0.02 | 0.01 | 0.03 | 0.01 | ... |
| 3 | 0.07 | 0.02 | 0.00 | 0.01 | 0.02 | 0.01 | ... |
| 4 | 0.08 | 0.01 | 0.01 | 0.00 | 0.24 | 0.03 | ... |
| 5 | 0.08 | 0.03 | 0.02 | 0.24 | 0.00 | 0.12 | ... |
| 6 | 0.08 | 0.01 | 0.01 | 0.03 | 0.12 | 0.00 | ... |
| ... | ... | ... | ... | ... | ... | ... | ... |

比如我们具体观察编号为16的轨迹，它和其他24辆出租车的JS散度分别为：0.119，0.048，0.049，0.276，0.070，0.095，**0.296**，0.114，0.113，0.130，0.101，0.125，0.106，**0.059**，0.311，0.000，0.251，0.267，0.113，0.083，0.279，0.053，0.058，0.051，0.142。显然可见，与该轨迹JS散度最小（即分布最相似）的轨迹是编号为14的轨迹，与该轨迹JS散度最大（即分布最不相似）的轨迹是编号为7的轨迹，因此，它与编号14的轨迹聚在一类的概率很大，而与编号7的轨迹聚在一类的概率很小。

得到轨迹之间的JS散度后，我们就可以利用2.3.3节中提到的K-Means聚类算法进行轨迹聚类，设聚类参数，即聚类的簇数为4，则聚类的结果如表2-2所示。

表2-2 轨迹聚类实验结果

|  |  |
| --- | --- |
| 簇编号 | 簇中轨迹编号 |
| 1 | 3, 5, 6, 10, 15, 19, 25 |
| 2 | 2, 4, 7, 8, 11, 18, 22, 24 |
| 3 | 14, 16 |
| 4 | 1, 9, 12, 13, 17, 20, 21, 23 |

由于多个轨迹数据在进行可视化时会比较杂乱，故从每个簇中选取一个轨迹进行可视化，可以看出这4个簇的大致分布区别，见下图2-3，图2-4，图2-5和图2-6。

由图2-3可见，簇1中的轨迹10主要分布在北京市城区的西北方向。

由图2-4可见，簇2中的轨迹11主要分布在北京市城区的东北方向。

由图2-5可见，簇3中的轨迹16主要分布在北京市城区的中部。

由图2-6可见，簇4中的轨迹21主要分布在北京市城区的中部及东南方向。

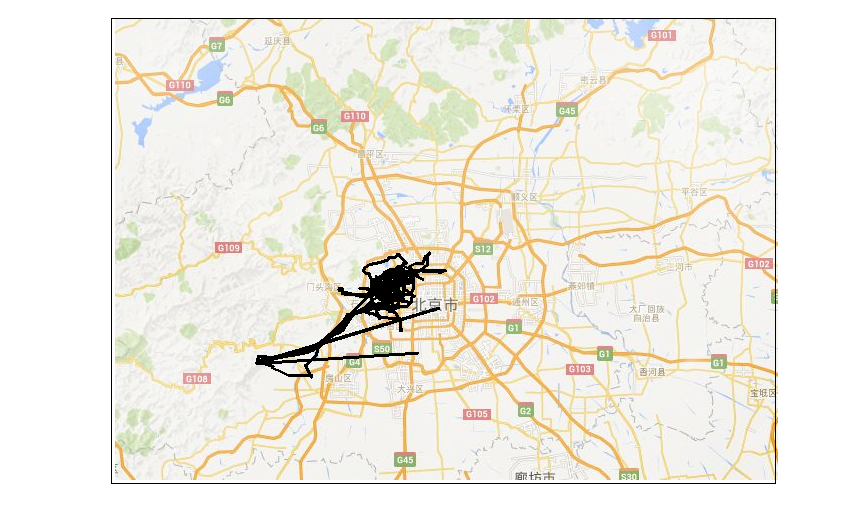


图2-3 轨迹10可视化图像

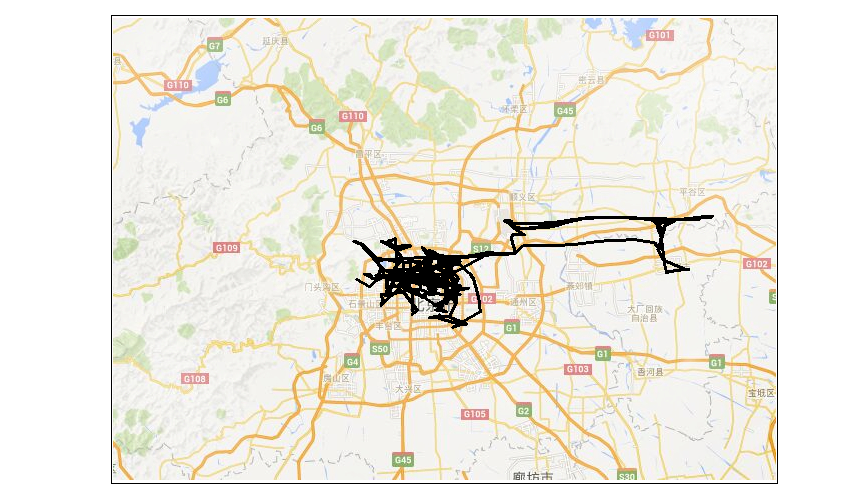


图2-4 轨迹11可视化图像

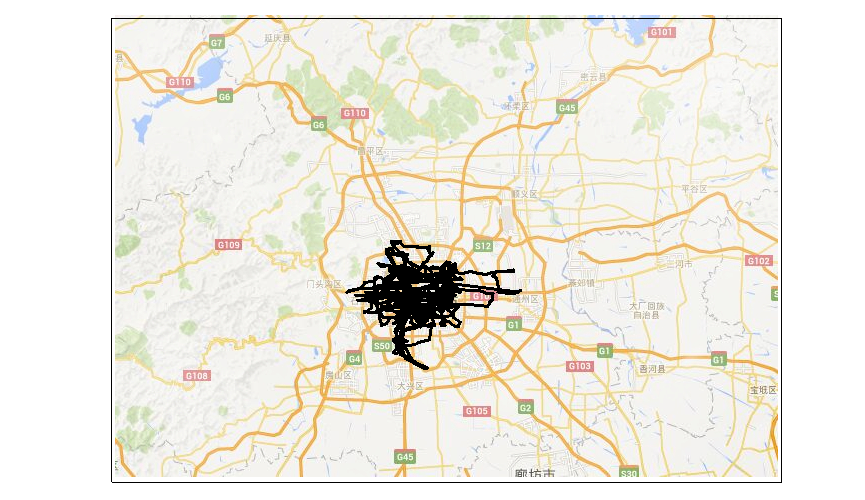


图2-5 轨迹16可视化图像

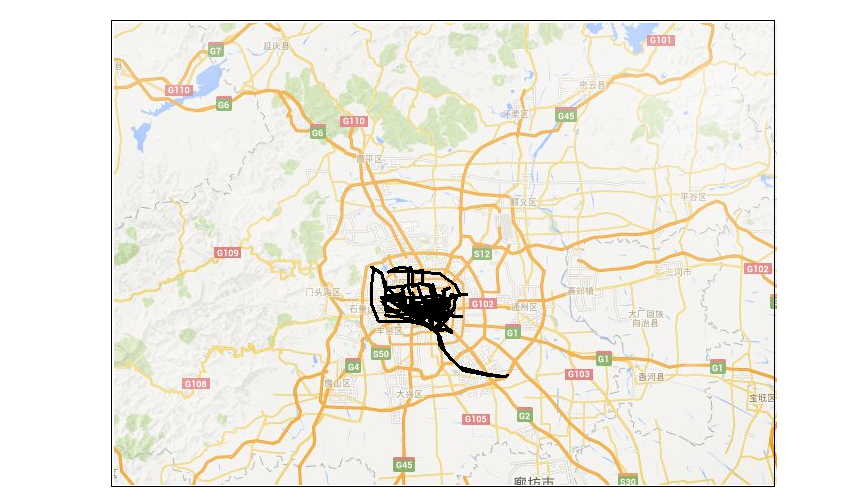


图2-6 轨迹21可视化图像

因此，从以上来自4个簇的轨迹可以看出，本文的轨迹聚类算法基本能将分布相似度较低的轨迹分在不同的簇中。

# 第3章 轨迹预测算法

随着卫星、无线网络以及定位设备的发展，大量移动轨迹数据呈急速增长的趋势，如交通轨迹数据、人员移动数据等。

通过轨迹数据挖掘发现隐含的知识，研究其行为模式并做出预测，可以帮助政府和用户做出更好的决策，甚至可以成为解决城市交通、城市环境、突发事件应急等重大社会问题的有效手段。例如，在交通方面，通过分析车辆的轨迹可以得知道路的拥堵情况，从而可以根据推测的拥堵情况来向车辆提示路况堵塞情况和最佳导航，方便城市交通的协调；另外，通过分析用户历史轨迹数据，还可以挖掘出人们之间的社交关系，从而为人们提供旅游、好友等推荐服务。

因此，近年来轨迹数据挖掘越来越受到各界的关注，包括计算机科学、社会学和地理学等在内的各个领域都将其列为重要研究课题。而移动对象的位置预测技术可以帮助提供更好的基于位置的服务， 有助于分析和理解轨迹数据，具有深远的意义和很大的发展空间。

本文的最终目的是预测出对象未来的群体分布，但是本文实现这一目的的方法与传统的群体分布预测不同，传统的群体预测方法一般直接使用群体历史数据来预测群体未来分布，而本文的研究思路是先基于个体轨迹数据预测出每个个体下一步可能到达位置的概率分布，再将个体预测结果整合起来，最终得到群体分布的预测结果。

在第2章中，我们已经提出了一个轨迹聚类算法来寻找相似的轨迹，那么本章将基于第2章的聚类结果，继而设计轨迹的预测算法。在轨迹预测中，我们会遇到马尔可夫链和转移矩阵的相关知识，故在这章中，会首先介绍所设计的轨迹预测算法的基本思想，再介绍马尔可夫链和转移矩阵，之后介绍的转移矩阵的学习方法以及权重学习方法，最终通过实验验证。

## 3.1 基本思想

在已知一系列个体轨迹数据的前提下，为了预测对象未来的群体分布， 本文希望先基于个体轨迹数据预测出每个对象下一步可能到达位置的概率分布，再将个体预测结果整合起来，最终得到群体分布的预测结果。

因此，如何预测出每个对象下一步可能到达位置的概率分布就成了重点研究的核心。为了实现这一目的，本文提出了个体轨迹预测的中心思想——当预测某个个体轨迹时，不但可以参考其自身的历史轨迹数据，还可以参考与其相似的轨迹数据。因为如果只考虑个体对象本身的历史数据，可能会产生“过拟合”现象，即预测结果在训练集表现很好，但是在测试时的表现却远不如训练时，换句话说就是预测的适应性不够好。所以为了改善这个问题，希望可以参考与该个体轨迹相似的其他轨迹，共同来预测该个体轨迹下一步可能到达位置的概率分布。

围绕这一中心思想，本文在第2章中已经做了轨迹数据的聚类，以找到相似的轨迹并将它们聚为一类，从而得到*K*个簇内对象相似度较高、簇间相似度较低的簇。

得到这些簇后，就可以进一步设计接下来的轨迹预测算法。这里用到了马尔可夫链的概念，假设每个轨迹下一步会到达的位置只与上一步所处的位置有关，于是可以把每一条轨迹都看作一条马尔可夫链，从而可以用到马尔可夫链中转移矩阵来描述对象的移动特征。这里的转移矩阵是一个描述从某个网格转移到其他网格的概率的矩阵，假如空间网格的数量总共有个，则转移矩阵为一个的矩阵。马尔可夫链和转移矩阵的基本原理将在之后的3.2节中具体介绍。

引入转移矩阵的概念后，我们就可以分别基于个体的自身轨迹数据和簇内轨迹数据，通过学习得到两个转移矩阵和，这两个转移矩阵分别代表个体轨迹自身的转移特征和簇内轨迹的整体转移特征。它们对于个体轨迹的预测都会产生一定影响，但是影响的大小不一定相同，所以最终的转移矩阵需要将它们按权重相加，如下式

 （3-1）

其中*a*,*b*满足

 （3-2）

得到最终的转移矩阵*Y*后，就可以预测出轨迹在下一步会到达位置的概率分布。这里所预测的位置就是上文2.3.1节中提到的网格，假如此时轨迹正处在网格*g*中，则它下一步会到达位置的概率分布为，表示转移矩阵*Y*的第g行，即从网格*g*出发下一步会到达其他网格的概率分布。

由于对每一条轨迹*i*来说，都可以得到一个概率分布，记为，故把这些概率分布相加起来即可得到群体的分布：

 （3-3）

以上就是基于个体轨迹来进行群体分布预测的算法思想，下面将详细介绍每一部分的原理及实现方法。

## 3.2 马尔可夫链与转移矩阵

在随机过程理论中，马尔可夫过程是一类占有重要地位、具有普遍意义的随机过程，它广泛应用于近代物理、生物学、公用事业、地质学和大气科学等各个领域。本节将介绍马尔可夫链和转移矩阵的相关原理。

### 3.2.1 马尔可夫链原理介绍

如果把每一条轨迹都看作一条马尔可夫链，就以通过马尔可夫链中的转移矩阵来描述对象的移动特征，故这里先介绍马尔可夫链的基本原理和性质。

马尔可夫过程[18]是一类有重要应用意义的随机过程，它具有如下特征：随机过程“将来”所处的状态仅与“现在”所处的状态有关，而与“过去”曾处于什么状态无关。

马尔可夫过程按其状态和时间参数是离散还是连续的可以分成三类：

（1）时间和状态都是离散的马尔可夫过程，称为马尔可夫链；

（2）时间连续、状态离散的马尔可夫过程，称为连续时间的马尔可夫链；

（3）时间和状态都连续的马尔可夫过程。

在本文中，由于轨迹数据由GPS等设备采样得到，故其时间和状态都是离散的，所以这里我们研究的应该是马尔可夫链，以下是马尔可夫链的定义。

设为随机序列，其状态空间为，如果对任意正整数*t*有

 （3-4）

则称此随机序列为马尔可夫链。

若将时刻*t*称为“现在”，将时刻*t*+1称为“将来”，而把称为“过去”，则定义中的等式可较通俗得解释为在已知“现在”所处的状态的条件下，“将来”所要达到的状态与“过去”所经历的状态无关，这一特性常称为马尔可夫的无后效性。在自然界与社会现象中，许多随机现象遵循这一演变规律。例如，微分方程的初始值问题描述的物理系统（或物理过程）就属于这类确定性现象；又如，某生物群体的大小只与现状的大小有关，而与过去该生物群体的大小无关，这一事实也属于这类随机性现象。

在本文中想用马尔可夫链来描述一条轨迹，则每个采样点就可以当作一次状态转移，而状态空间即网格空间，假如空间网格的数量总共有个，则有个状态。

### 3.2.2 转移矩阵

由马尔可夫链的无后效性公式（3-4）和乘法公式可以得出

 （3-5）

由此可以看出，马尔可夫链的统计特性完全由条件概率所确定，所以如何确定这个条件概率就显得非常重要，我们把这个条件概率称为一步转移概率。一般一步转移概率为，它表示系统在时刻*t*处于状态的条件下，到时刻*t*+1转移到状态的概率，记为。

一般，转移概率不仅与状态有关，而且与时刻*t*有关，但当它与时刻*t*无关时，表示马尔可夫链具有平稳转移概率，即与起点无关，此时我们称马尔可夫链是齐次的，此时转移概率可以直接表示为。

假设马尔可夫链是齐次马尔可夫链，设P为一步转移概率所组成的矩阵，状态空间，共*m*个状态，则马尔可夫链的一步转移概率矩阵可以表示为：

**** （3-6）

马尔科夫的一步转移概率矩阵（简称转移矩阵）表示了每个状态之间相互转换的概率，转移矩阵具有以下几个性质：

1. 非负性：转移概率矩阵中的每一项都不小于0。该性质用公式表示为

  （3-7）

1. 归一性：转移矩阵中，从某一个状态到其他所有状态（包括自己）的概率总和应该等于1，即矩阵中的每行之和应该等于1。由于矩阵中的每一行表示过程由一种状态向其他状态转移的所有可能性，所有的可能性加在一起就成为一个必然事件，而必然事件的概率恒是 1。该性质用公式表示为

  （3-8）

1. 非对称性：一般来说，由状态*i*转向状态*j*的转移概率不等于状态*j*转向状态*i*的概率。该性质用公式表示为

  （3-9）

同时，如果一个矩阵满足以上两个条件，它才可以被称为转移矩阵，所以这也是我们求解转移矩阵的标准。

假设对象正处在状态（），那么在下一时刻它有*m*种转移的可能性（包括继续保持此刻的状态），可以用表3-1表示。

表3-1 状态转移概率示意图

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 状态转移 |  |  | ... |  | ... |  |
| 转移概率 |  |  | ... |  | ... |  |
| 简记 |  |  | ... |  | ... |  |

### 3.2.3 转移矩阵的求解

由于马尔可夫链通过研究对象的状态转移概率来进行预测，因此状态转移概率的确定成为了马尔可夫模型预测的关键，较准确的算出马尔可夫状态转移概率矩阵能提高马尔可夫预测的准确性。 目前，许多学者已经针对状态转移概率矩阵的求解方法进行了很多研究，旨在通过有效的方法应用马尔可夫预测模型进行各个方面的预测，或者提高马尔可夫预测模型的预测精度。目前求解状态转移概率矩阵的方法中，最常见的是采用统计法利用先验知识估算状态转移矩阵。

统计法估算马尔可夫状态转移矩阵是一种最为传统且应用较广的方法，它的使用前提是状态转移的数据具有可统计性，故在进行统计法估算马尔可夫状态转移矩阵前首先需要获取统计数据，因此它一般有以下步骤：

（1）根据实际问题确定状态数量，并从数据集中提取所取得状态转移数据，统计每种状态转移出现的次数；

（2）筛选并处理所得数据；

（3）应用已有数据，计算状态转移概率的初始矩阵；

（4）通过统计到的状态转移数据，计算出状态转移概率矩阵。

（5）建立马尔可夫链模型进行预测。

设表示由状态转移到状态出现的次数，那么由状态转移到状态的频率可以由以下公式计算

  （3-10）

根据概率论与数理统计知识可知，当试验次数或统计量较大的时候，频率可以作为概率的一个近似值，或者说概率是可以通过频率来测量来。因此，在应用统计的方法估算概率矩阵时，可以用状态转移的频率近似地估计转移概率。设是从状态转移到状态的转移概率的估计值，则根据公式（3-11），可以由下式计算得到

  （3-11）

则状态转移概率矩阵的估计值为

**** （3-12）

至此，就已经完成了用统计法来估算马尔可夫状态转移矩阵的方法。显然可见，这种方法简单易行，因此在各个领域都获得了广泛得应用。

在本文中，我们利用马尔可夫链来描述一条轨迹，则每个采样点就可以当作一次状态转移，而状态空间即网格空间，假如空间网格的数量总共有个，则有个状态，因此转移矩阵为一个的矩阵。

而对每一条轨迹来说，我们需要计算出两个转移矩阵，一个是根据个体轨迹自身数据的统计值计算出的转移矩阵，一个是根据同簇轨迹数据的统计值计算出的转移矩阵，这两个转移矩阵分别代表个体轨迹自身的转移特征和簇内轨迹的整体转移特征。

## 3.3 权重学习

根据上文中的介绍，我们已经可以计算出基于个体轨迹自身数据的转移矩阵和基于同簇轨迹数据的转移矩阵，它们对于个体轨迹的预测都会产生一定影响，但是影响的大小不一定相同，所以在预测时需要给它们分配权重，再将它们按权重相加

 （3-13）

其中*a*,*b*满足。

而对于每一条轨迹来说，其个体特征和簇内整体特征对它的影响可能并不一样，因此我们需要想办法来尽量合理得分配这个权重，即*a*和*b*的取值。

如果把和看作两个变量，不难把公式（3-13）看作一个二元线性回归问题，因此这里我们希望利用与回归有关的方法来进行权重学习。由于和实际上是一个矩阵而非真正的变量，故而在求解回归问题时不能直接使用最常用的最小二乘法，而需要用到梯度下降发。因此，本节接下来会首先对回归分析和梯度下降法做一个介绍。

### 3.3.1 回归分析

现实生活中，一切事物都是相互关联、相互制约的。我们将变化的事物看作变量，那么变量之间的相互关系，可以分为两大类：一类是确定性关系，也叫作函数关系，其特征是一个变量随着其它变量的确定而确定，如矩形的面积由长宽确定；另一类关系叫相关关系，其特征是变量之间很难用一种精确的方法表示出来，如商品销量与售价之间有一定的关联，但由售价我们不能精确地计算出销量。不过，确定性关系与相关关系之间没有一道不可逾越的鸿沟，由于存在实际误差等原因，确定性关系在实际问题中往往通过相关关系来体现；另一方面，当对事物内部规律了解得更加深刻时，相关关系也可能转化为确定性关系。

回归分析[19]就是处理变量之间的相关关系的一种数学方法，它是最常用的数理统计方法，能解决预测、控制、生产工艺化等问题。由相关关系函数确定形式的不同，回归分析一般分为线性回归、非线性回归和逐步回归，在这里我们着重介绍线性回归，它是比较简单的一类回归分析，在实际问题的处理中也是应用得较多的一类。

回归分析中最简单的形式是

 （3-14）

其中，固定的未知参数，称为回归系数，自变量称为回归变量**，**我们称公式（3-14）为一元线性回归。

一般的，这些回归系数可以通过最小二乘法求解，它将最佳拟合直线估计为最小化实际数据与直线的估计值之间的误差的直线。设训练集由预测变量*x*的值和它们的相关联的响应变量*y*的值组成：，训练集共包含*L*个数据点，则回归系数可以用下式估计：

 （3-15）

 （3-16）

其中，是的均值，而是的均值。

一元线性回归问题的一个自然推广是为多元变量，形如

 （3-17）

此时，这类回归问题被称为多元线性回归问题。特别的，当时为二元线性回归问题，本文需要研究的就是二元线性回归问题，回归变量即为基于个体轨迹自身数据的转移矩阵和基于同簇轨迹数据的转移矩阵。

### 3.3.2 梯度下降法

梯度下降法[20]是一种[最优化算法](http://zh.wikipedia.org/wiki/%E6%9C%80%E9%80%9F%E4%B8%8B%E9%99%8D%E6%B3%95)，常用来优化参数，通常也称为最速下降法。

梯度下降法是一般分为如下两步：

（1）首先对参数*w*赋值，这个值可以是随机的，也可以让*w*是一个全零的向量；

（2）改变*w*的值，使得按梯度下降的方向进行减少。

以一个线性回归问题为例，假设要学习的函数表达式为：

 （3-18）

那么可以定义损失函数为：

 （3-19）

假设现在有*L*组样本数据，则可以写成如下方程组的形式：

 （3-20）

所有的可以用列向量*Y*表示，所有可以用矩阵*X*表示，一行代表一组样本，那么权重可以用列向量*W*表示。

那么此时，可以定义每个样本（每行）的损失变量为：

 （3-21）

那么这时损失函数就可以表示成：

 （3-22）

梯度下降法的目标很简单，就是希望求得使损失*J*最小值时候的解*W*。所以首先对*J*求导：

 （3-23）

带入公式（3-21）可得：

 （3-24）

整理上式为矩阵形式可以表示为

 （3-25）

令该导数等于0，可以求此时的*W*，故

 （3-26）

但是求解矩阵的逆复杂度有比较高，特别是当矩阵较大时，故可以用梯度下降的思路来求解，使参数*W*在不断向最优值改变（增加或减少），设此时为第*t*次迭代（改变），则参数改变的公式为

 （3-27）

其中就是下降的速度，一般是一个小的数值，可以从0.01开始尝试，越大下降越快，收敛越快。

一般来说，设梯度下降法的迭代终止的条件为

 （3-28）

其中，为一个较小的阈值，当参数的改变量小于这个阈值时即可停止迭代。

### 3.3.3 轨迹转移矩阵的权重学习

上文简单介绍了一般的回归问题和梯度下降法，那么接下来，我们希望把这种方法应用在轨迹的转移矩阵上。

根据公式（3-13），我们用代替，用代替，则可以把预测问题写成二元线性回归的形式

 （3-29）

但是与一般的二元线性回归不同，这里的都是矩阵，代表转移矩阵，而不是一个简单的变量，所以在定义损失函数时会有所改变。

确定此回归问题后，我们需要找*L*组样本来求解该回归问题，因此这里需要解决如何获取*L*组样本的问题。

这里采用“滑动窗口法”来获取*L*组样本数据，设滑动窗口的大小为，那么对于轨迹*T*来说，统计第一个样本中的个体转移矩阵时，使用的便是该轨迹第一个窗口（轨迹的第1个点至第点）的数据，记为。同理，统计第一个样本中的簇内转移矩阵时，使用的是簇内所有轨迹的第一个窗口（所有轨迹的第1个点至第点）的数据，记为。而统计第一个样本中的预测转移矩阵*Y*时，使用的是该轨迹轨迹相对的下一个窗口（第点至第点）的数据，记为。因此，第*k*组样本中的矩阵就可记为。

设表示第*k*组样本中矩阵的第*i*行第*j*列（即由状态*i*转移为状态*j*的概率），表示第*k*组样本中矩阵的第*i*行第*j*列，同理，则设

 （3-30）

此时损失函数可以表示为

 （3-31）

上式对求导可得

 （3-32）





同理对求导为

 （3-33）

此时即可用梯度下降法来求参数，方法如下

 （3-34）

 （3-35）

实际上为了更快更好得求解，在进行梯度下降法之前可以提前求出以下5个值：

，，，

，

在使用梯度下降法的过程中，这5个值相对于参数来说相当于是5个常数值，故可以提前求取。

由此，便可以求出每条轨迹的两个权重参数，但是这个值还不能直接当作权重使用，因为由公式（3-12）可知，权重之和相加应该等于1，故需要对其进行归一化，则最终的权重可由以下公式计算

 （3-36）

 （3-37）

得到个体转移矩阵和簇内转移矩阵的权重后，就可以计算某条轨迹下一步的空间概率分布，假设用于学习的数据集为所有轨迹的第1个点至第*v*个点，则此时分别由个体的第1个点至第*v*个点和簇内每条轨迹的第1个点至第*v*个点数据统计得到，这样就可以预测该轨迹第*v*+1个点的空间概率分布：

 （3-38）

其中，表示该轨迹第*v*个点所处的状态（即网格编号），表示矩阵的第行，表示矩阵的第行，表示矩阵的第行。

由于对每一条轨迹*i*来说，都可以得到一个概率分布，记为，故把这些概率分布相加起来即可得到群体的分布

 （3-39）

## 3.4 实验

在设计好轨迹的预测算法后，我们需要在这里对其做简单的实验验证。在2.4节中，我们已经对聚类算法进行实验，由于预测算法是基于聚类算法进行的，故这里沿用2.4节中的聚类结果，并基于此结果进一步运行预测算法。

首先，我们要计算每条轨迹基于个体轨迹自身数据的转移矩阵和基于同簇轨迹数据的转移矩阵，计算的方法为3.2.3节中提到的统计估算法。

对每条轨迹来说，在计算出和后，就可以将它们加权求和得到最终的转移矩阵*Y*，因此接下来要做的就是学习它们的权重*a*和*b*，这里权重学习的过程用到了3.3节中提到多元线性回归法和梯度下降法。最终可以学习出每条轨迹的这两个权重，结果可以用表3-2表示。

表3-2 权重学习结果

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 轨迹编号 | 权重*a* | 权重*b* | 轨迹编号 | 权重*a* | 权重*b* |
| 1 | 0.19 | 0.81 | 14 | 0.39 | 0.61 |
| 2 | 0.49 | 0.51 | 15 | 0.48 | 0.52 |
| 3 | 0.45 | 0.55 | 16 | 0.37 | 0.63 |
| 4 | 0.36 | 0.64 | 17 | 0.37 | 0.63 |
| 5 | 0.16 | 0.84 | 18 | 0.18 | 0.82 |
| 6 | 0.17 | 0.83 | 19 | 0.29 | 0.71 |
| 7 | 0.36 | 0.64 | 20 | 0.26 | 0.74 |
| 8 | 0.16 | 0.84 | 21 | 0.34 | 0.66 |
| 9 | 0.20 | 0.80 | 22 | 0.31 | 0.69 |
| 10 | 0.20 | 0.80 | 23 | 0.39 | 0.61 |
| 11 | 0.46 | 0.54 | 24 | 0.42 | 0.58 |
| 12 | 0.19 | 0.81 | 25 | 0.34 | 0.66 |
| 13 | 0.21 | 0.79 |  |  |  |

从表3-2可以看出，对不同轨迹来说，个体自身历史数据和簇内轨迹数据对它未来的影响比重是不同的。比如轨迹6的簇内数据影响度是最大，而想比之下轨迹12的自身数据影响度较大。但是有趣的是，对这些轨迹来说，簇内数据影响度都大于自身数据影响度，即权重*b*都大于权重*a*。

求得权重后，就可以得到每条轨迹的最终转移矩阵*Y*，以轨迹11为例，它的权重a为0.46，权重b为0.54，则根据公式（3-13）可知它的转移矩阵计算公式为

 （3-40）

轨迹11在第1000个采样点时所处的网格编号为4149（即第41行第49列网格），那么根据其转移矩阵，可知轨迹11下一步到网格4049（即第40行第49列的网格）的概率最大。

# 第4章 实验与讨论

## 4.1 数据集说明

关于轨迹数据，公开的数据集较少。这里，我们找到了2011年发布的T-Drive 轨迹数据集[21]。这个数据集包含10357辆出租车一周内的GPS采样数据。但是需要注意的是，这些出租车的轨迹采样范围可能超出北京市范围，还有些采样点甚至因设备问题而无法正常采样，另外这些出租车采样的点的数量也并不一致，甚至数量相差很大（例如编号为1131的出租车有九万多个采样点，而编号为9424的出租车只有几十个采样点）。因此在这里，我们通过对数据的预处理，筛选出了比较合适的25辆出租车作为待实验的轨迹数据。

## 4.2 算法流程

为了基于这些个体轨迹数据实现群体分布的预测，我们需要使用第2章中设计的轨迹聚类算法和第3章中设计的预测算法，算法的整体流程可以整理如下：

1. **数据预处理：**

首先要筛选合适的数据，筛选的条件是始终在北京市地界以内范围内移动、没有出现因设备故障而采样无效的情况、轨迹中至少包含1000个采样点。

**2. 网格划分：**

把连续的平面地图划分为网格状的区域，从而把原本的一条轨迹数据转化为一串经过的区域的序列。

**3. 相似度度量（距离度量）：**

计算每两条轨迹之间的JS散度，以度量轨迹之间的距离，若两条轨迹之间的JS散度越小，则表示两条轨迹的相似度越大。

**4. 轨迹聚类：**

基于相似度度量结果，利用K-Means聚类方法来对轨迹进行聚类，较为相似的轨迹将聚为一类，形成一个簇，从而使簇内轨迹相似度较高，簇之间相似度较低。

**5. 个体轨迹预测：**

预测每条轨迹下一步要去的位置的概率分布，这部分主要包括转移矩阵计算和权重学习两部分，先分别计算出基于个体自身数据的转移矩阵和基于簇内轨迹数据的转移矩阵，再学习它们所占的权重，最终加权求和得到最终的转移矩阵，由于权重是归一化的，所以这个最终转移矩阵的某一行实际上就是从对应网格出发的下一步概率分布。

**6. 群体分布预测：**

根据每条轨迹的下一步概率分布预测结果，把它们累加起来，就可以得到群体在下一步时的分布。

通过以上流程，就可以完成基于个体轨迹数据的群体分布预测算法。

## 4.3 结果及讨论

基于以上步骤进行实验，假设所有轨迹都处在第1000号采样点，则可以通过学习出的转移矩阵来求得群体在下一步（即1001号点）的分布，由于分布矩阵比较庞大且有很多零值，故只取了其中一小部分如下表4-1表示。

表4-1 实验所得群体分布

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 网格号 | ... | 40 | 41 | 42 | 43 | 44 | ... |
| 30 | ... | ... | ... | ... | ... | ... | ... |
| 31 | ... | 0.018 | 0.055 | 0.000 | 0.000 | 0.000 | ... |
| 32 | ... | 0.051 | 0.037 | 0.000 | 0.002 | 0.000 | ... |
| 33 | ... | 0.073 | 0.325 | 0.037 | 0.069 | 0.000 | ... |
| 34 | ... | 0.018 | 0.176 | 0.089 | 0.000 | 0.000 | ... |
| 35 | ... | 0.000 | 0.000 | 0.021 | 0.044 | 0.014 | ... |
| 36 | ... | 0.000 | 0.051 | 0.028 | 0.844 | 0.029 | ... |
| 37 | ... | 0.090 | 0.007 | 0.948 | 0.004 | 0.002 | ... |
| 38 | ... | 0.615 | 0.027 | 0.034 | 0.005 | 0.000 | ... |
| ... | ... | ... | ... | ... | ... | ... | ... |

用Matlab对上述分布矩阵进行可视化，可得到分布的三维图和二维俯视图如下图4-1和图4-2所示。

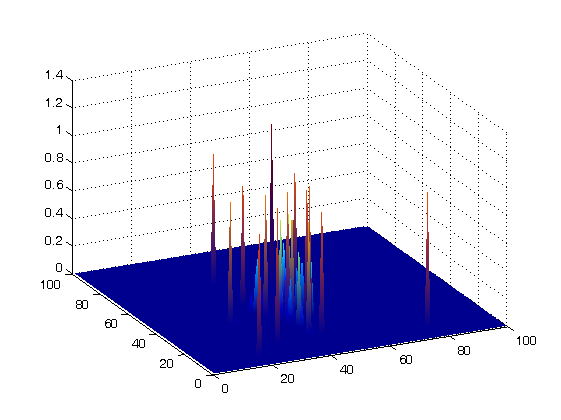


图4-1 实验所得群体分布三维图

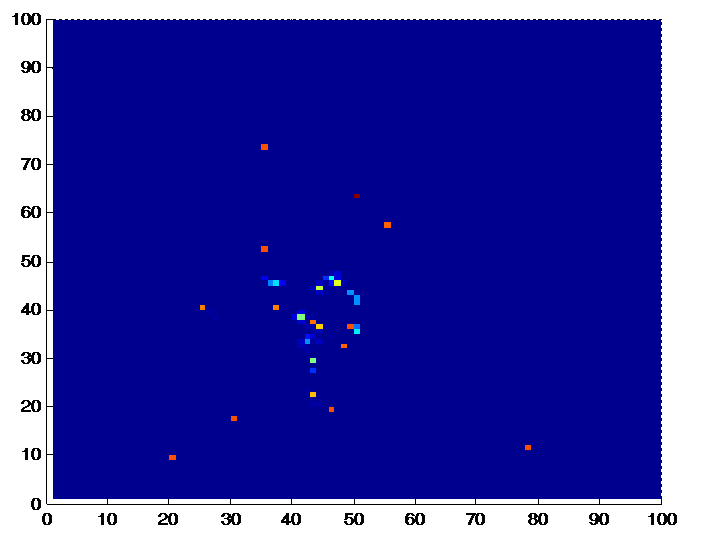


图4-2 实验所得群体分布二维俯视图

其中，颜色越红的部分表示其值越大，蓝色的部分表示值很小甚至等于0。可以看出，分布大多比较集中在某些很小的区域内，而其他大部分位置分布值都等于0。

实际上，根据这些轨迹数据，我们可以知道这25条轨迹下一步的真实分布，用Matlab画图可得，如图4-3。

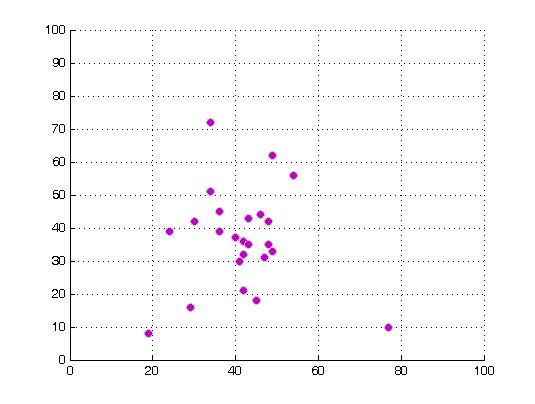


图4-3 真实群体分布图

比较预测所得分布图（图4-2）和真是群体分布图（图4-3）可以发现，预测结果和真实分布十分吻合。实际上，统计预测分布中值最大的25个网格点，发现其中有20个网格点与真实分布相符，因此此时正确率为80%。随着轨迹数据的增加，正确率可能还会有所提升。

由以上所有实验结果可以看出，无论是在轨迹聚类方面还是在轨迹预测方面，该算法都有其合理性。另外，本文的算法把群体知识和个体知识结合了起来，不但使群体分布预测中融入了更多的细节特征，也使个体轨迹的预测更加可靠。

# 第5章 结语

这一章作为本文的最后一章，首先总结了这篇文章的主要工作，然后以上述 工作为基础，讨论了接下来即将开始的工作。

## 5.1本文内容

本文的主要研究内容是从一系列个体轨迹数据中提取有效信息，设计数据挖掘方法，实现对它们未来群体分布的预测。这与传统的轨迹预测方法有所不同，传统的轨迹预测一般是利用个体轨迹历史数据预测个体的未来轨迹，或利用群体分布历史数据预测群体的未来分布。而本文将两者结合起来，先基于个体轨迹数据预测出每个个体下一步可能到达位置的概率分布，再将个体预测结果整合起来，从而得到群体分布的预测结果。

基于以上研究内容和研究思路，可以看出本文的研究核心在于如何可靠得对个体轨迹进行预测，于是本文提出了个体轨迹预测的中心思想——当预测某个个体轨迹时，不但可以参考其自身的历史轨迹数据，还可以参考与其相似的轨迹数据。围绕这一中心思想，本文首先提出了轨迹聚类算法，以找到相似的轨迹并将它们聚为一类，形成一个簇。接下来又提出了轨迹预测算法，基于轨迹聚类结果，利用个体轨迹自身历史数据和同簇轨迹数据对每条轨迹进行预测，并将所有个体轨迹预测结果整合得到群体分布预测结果。

本文第一部分主要介绍了轨迹数据挖掘的研究背景、研究意义及相关工作回顾，并介绍了文章的组织结构。

第二部分提出了一个轨迹聚类算法，包括该聚类算法中所需用到的相似度度量方法和K-Means聚类方法，并在出租车数据集上分析聚类结果。

第三部分提出了一个轨迹预测算法，基于马尔可夫链分别计算出个体数据的转移矩阵和簇内数据的转移矩阵，并利用多元线性回归和梯度下降法学习它们的权重，最终把个体轨迹预测结果整合得到群体分布结果。

最后一部分把本文第二章和第三章提出的算法应用于出租车数据集，按整个算法流程进行实验，完成对出租车未来群体分布的预测。

## 5.2改进方向

本文所提出的轨迹聚类和预测的方法可能还可以进一步改进，使聚类和预测的效果更好。

对于轨迹聚类方法，本文是基于JS散度相似度度量进行了K-Means聚类，但是K-Means聚类也有一定的局限性，如它的参数*k*（聚类簇数）需要人为设置，不能智能学习，故下一步可以寻找一些改进方法实现*k*值的自动调整。另外本文在度量轨迹间相似度时只考虑了轨迹在空间上的分布，而没有考虑在时间上的分布，故接下来可以把时间分布因素也考虑到相似度度量中。

对于轨迹预测方法，本文目前只是离线的预测下一步分布，故下一步可以考虑进行在线多步预测，即当轨迹数据不断被实时获取时，可以不断通过预测值和真实值的差值对预测参数进行修正，再继续预测下一步甚至一段时间的分布。

另外由于硬件等原因，本文只选取了较小的数据进行实验，如果实验设备允许，还可以进行较大数据的实验，可能预测结果会更好。

# 参考文献

[1] J. Han, M. Kamber, J. Pei. Data mining: concepts and techniques[M]. Elsevier, 2011

[2] U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, et al. Knowledge Discovery and Data Mining: Towards a Unifying Framework.[M]. KDD, 1996, 82–88

[3] I. H. Witten, E. Frank. Data mining: Practical machine learning tools and techniques[M]. Morgan Kaufmann, 2005

[4] S. CITIES. Trace analysis and mining for smart cities: issues, methods, and applications[J]. IEEE Communications Magazine, 2013

[5] Y. Zheng. Trajectory data mining: an overview[J]. ACM Transactions on Intelligent Systems and Technology (TIST), 2015, 6(3):29

[6] D. H. Douglas, T. K. Peucker. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature[J]. Cartographica: The International Journal for Geographic Information and Geovisualization, 1973, 10(2):112–122

[7] R. Bellman. On the approximation of curves by line segments using dynamic programming[J]. Communications of the ACM, 1961, 4(6):284

[8] S. Fortunato. Community detection in graphs[J]. Physics reports, 2010, 486(3):75–174

[9] H. Wang, H. Su, K. Zheng, et al. An effectiveness study on trajectory similarity measures[M]. Proceedings of the Twenty-Fourth Australasian Database Conference-Volume 137, 2013, 13–22

[10] R. Agrawal, C. Faloutsos, A. Swami. Efficient similarity search in sequence databases[M]. Springer, 1993

[11] E. J. Keogh, M. J. Pazzani. Derivative Dynamic Time Warping.[M]. Sdm, 2001

[12] P. Chen, J. Gu, D. Zhu, et al. A dynamic time warping based algorithm for trajectory matching in LBS[J]. International Journal of Database Theory and Application, 2013, 6(3):39–48

[13] D. Buzan, S. Sclaroff, G. Kollios. Extraction and clustering of motion trajectories in video[M]. Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on, 2004

[14] J. H. Martin, D. Jurafsky. Speech and language processing[J]. International Edition, 2000

[15] J. G. Lee, J. Han, K. Y. Whang. Trajectory clustering: a partition-and-group framework[M]. Proceedings of the 2007 ACM SIGMOD international conference on Management of data, 2007

[16] S. Kullback, R. A. Leibler. On information and sufficiency[J]. The annals of mathematical statistics, 1951, 22(1):79–86

[17] D. M. Endres, J. E. Schindelin. A new metric for probability distributions[J]. IEEE Transactions on Information theory, 2003

[18] J. R. Norris. Markov chains[M]. Cambridge university press, 1998

[19] J. S. Armstrong. Illusions in regression analysis[J]. Available at SSRN 1969740, 2011

[20] J. Snyman. Practical mathematical optimization: an introduction to basic optimization theory and classical and new gradient-based algorithms[M]. Springer Science & Business Media, 2005

[21] J.Yuan, Y.Zheng, X.Xie, et al. Driving with knowledge from the physical world[M]. Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining, 2011

致 谢

岁月荏苒，时光不息，随着毕业设计的逐渐完成，来到电子科技大学的四年也即将成为美好的回忆，在此，我要真诚得向所有指导、帮助和陪伴过我的人表示由衷的感谢。

首先我要感谢我的指导老师邵俊明老师，他对学术的不懈追求和对学生的认真负责令我如沐春风，每一次与他的讨论都能感受到他渊博的知识与精妙的思维方式，令我受益匪浅，邵老师将是我永远的学习榜样。

感谢所有在课堂上为我们传道授业解惑的老师和课下指导帮助过我的老师们，感谢为英才实验学院授课的每一位老师，感谢指导我数学建模的张勇老师，感谢教我《通信网理论基础》的王晟老师，你们的无私敬业使我的大学时光变得充实。

感谢我在电子科技大学认识的每一位同学和朋友，与辩论队队友一起的辩之皇皇，与数学建模队友一起的不舍昼夜，与室友们的深夜卧谈，与朋友们的相互鼓励，都将是我永远的美好回忆。

最后我要特别感谢我的父母和亲人，他们无微不至的关怀和鼓励是我永远的精神支柱，在我失落无助的时候给予我慰藉，让我可以无忧无虑得继续成长。