高时效性下的轨迹局部相似性查询算法的设计与实现

目录

[第一章 绪论 3](#_Toc527744550)

[1.1研究背景及意义 3](#_Toc527744551)

[1.2本文研究内容 6](#_Toc527744552)

[1.3本文组织结构 7](#_Toc527744553)

[第二章 相关理论与关键技术 8](#_Toc527744554)

[2.1轨迹表示方法 8](#_Toc527744555)

[2.1.1仅基于位置信息的轨迹表示方法 8](#_Toc527744556)

[2.1.2基于时间信息的轨迹表示方法 8](#_Toc527744557)

[2.1.3基于文本信息的轨迹表示方法 9](#_Toc527744558)

[2.1.4基于网格的轨迹表示方法 9](#_Toc527744559)

[2.2现有轨迹相似性计算方法 9](#_Toc527744560)

[2.2.1欧氏距离（EU） 9](#_Toc527744561)

[2.2.2动态时间弯曲距离（DTW） 10](#_Toc527744562)

[2.2.3最长公共子序列距离（LCSS） 11](#_Toc527744563)

[2.2.4编辑距离（EDR） 12](#_Toc527744564)

[2.2.5带真实惩罚的编辑距离（ERP） 12](#_Toc527744565)

[2.2.6路网上的轨迹相似性查询 13](#_Toc527744566)

[2.2.7城市运输系统中基于段的轨迹相似性计算 15](#_Toc527744567)

[2.2.8简化的轨迹相似性计算 18](#_Toc527744568)

[2.2.9基于签名的轨迹相似性计算 19](#_Toc527744569)

[2.3 高效的轨迹索引技术 20](#_Toc527744570)

[2.3.1 R树索引 20](#_Toc527744571)

[2.3.2 Voronoi图 21](#_Toc527744572)

[2.4本章小结 22](#_Toc527744573)

[第三章 标准空间下的轨迹段相似性 23](#_Toc527744574)

[3.1 数值不均衡问题的提出 24](#_Toc527744575)

[3.1.1 轨迹的表示形式 **错误!未定义书签。**](#_Toc527744576)

[3.1.1 数值不均衡问题的定义 24](#_Toc527744577)

[3.1.2 数值不均衡问题的影响 24](#_Toc527744578)

[3.2 标准空间 25](#_Toc527744579)

[3.2.1 标准空间的定义 25](#_Toc527744580)

[3.2.2 标准空间的构建方法 26](#_Toc527744581)

[3.3标准空间下对应轨迹段的相似性 27](#_Toc527744582)

[3.2.1点距离的定义 **错误!未定义书签。**](#_Toc527744583)

[3.3.1轨迹段形状相似性 30](#_Toc527744584)

[3.3.2对应轨迹段的空间距离 37](#_Toc527744585)

[3.3.3对应轨迹段相似性 40](#_Toc527744586)

[第四章 基于标准空间的局部轨迹相似性算法 43](#_Toc527744587)

[4.1局部轨迹相似性问题的提出 43](#_Toc527744588)

[4.2轨迹段相似度及代价定义 43](#_Toc527744589)

[4.2.2 轨迹段的相似度 **错误!未定义书签。**](#_Toc527744590)

[4.2.2 获取轨迹段相似度的代价 **错误!未定义书签。**](#_Toc527744591)

[4.3基于代价的局部轨迹相似性算法 43](#_Toc527744592)

[第五章 实验设计与分析 44](#_Toc527744593)

[第六章 总结与展望 44](#_Toc527744594)

# 第一章 绪论

## 1.1研究背景及意义

近年来，随着移动设备和 GPS 的不断发展，人们已经可以很轻松地获取移动物体的地理位置信息，为了可以更好地利用这些信息，将需要使用一些技术手段去对这些信息进行处理，而更好地处理前提是需要更多的数据去支持算法的运行，从而又带动了地理位置的采集，形成一个良性循环。

数据采集中获取到的数据有很多种类。比如用户手持移动电话，通信公司就会根据手机信号和信号发射基站的位置去确定用户的具体位置，根据一个制定好的采样策略，收集用户所在位置的经纬度、当前时间，并根据多次采集数据去计算得到用户的平均移动速度和移动方向等。根据用户的移动数据，可以获取到大量用户的个性化信息，比如该用户经常去某家餐厅就餐，那么一些手机客户端可以根据用户喜好，为用户推荐类似口味的餐厅。或者根据用户一周内频繁出现的场所，为用户推荐周边的美食或娱乐场所。还可以根据用户的移动速度的变化，判断用户在某段路程里打了出租车，可以为用户推荐上车周边更好打车的地点。让用户可以不刻意得去记录自己的日常行为，仅仅被记录下行为轨迹，便可以获得个性化的推荐。

除了用户轨迹，现在很多出租车和私家车上都安装了车载GPS，通过车载GPS的工作可以将汽车每日的移动轨迹上传到服务器，然后通过对一个城市大量出租车、私家车的轨迹分析，可以得到很多信息。比如通过分析一天的轨迹信息中道路上车辆行驶速度，可以得到该城市每日早高峰晚高峰大约会出现在什么时间段，建议不赶时间的司机错峰行驶。还可以通过实时轨迹数据得到当前时间道路的车流量，判断该条道路在该时刻的拥堵状况，道路拥堵信息可以在广播频道里司机进行实时指导路线，或者在手机的出行app里动态展示，为司机挑选相对通畅的道路。

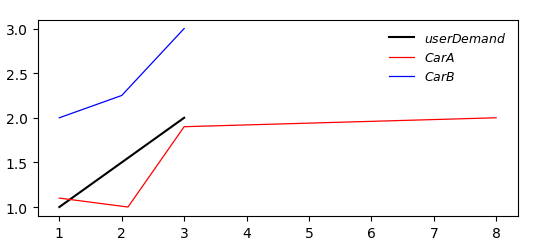
除了对用户位置的信息采集，对出租车移动路线信息的采集之外，还有对野生动物行为轨迹的采集，军事领域对地方目标轨迹的实时监测以实现精准打击，对飓风移动路径数据的采集来预报和预防自然灾害等等。随着数据采集设备的改良和采集方式的优化，各个领域都产生了海量的轨迹数据。所以对轨迹数据的分析利用变得十分重要。人们为了发掘海量数据中隐藏的价值，得到丰富的数据特征空间以及用户轨迹的规律性信息，开发了聚类分析、隐私保护和行为预测等一系列的应用技术，而这些技术的实现都得益于移动对象轨迹的相似性查询技术的发展。

轨迹数据展示了移动对象的时空动态，以数据的形式存储了空气、动物、车辆和人类的运动信息，在预测风暴移动、研究动物迁徙、规划城市建设和提供出行路线等方面有着重要的应用。而这些应用都需要轨迹数据库提供一个高效的轨迹查询功能，在移动对象的轨迹相似性查询中，相似性计算函数是核心。本文将在后面给出轨迹相似性查询以及相似性计算函数的定义。

当前该领域主要围绕两个问题进行研究，一是研究合适的相似性计算函数，二是研究高效的检索机制。选择一个合适的相似性计算函数和利用函数制定高效的检索机制至关重要，这些因素同时决定了查询方法的好坏。比如有时候我们无需对采样得到的整段轨迹计算与其他轨迹的相似度，只需要对一小段子轨迹选取合适的函数进行相似性计算即可，这样就可以在一定程度上减少运算时间，并获得相对而言更重要的轨迹相似性信息，因为相似的那段比不相似的那段更有价值。因此在面对不同场景时我们需要根据具体情况采用合适的相似性计算方法。

大多数对轨迹相似性的查询研究和常用的一些轨迹相似性计算函数，比如动态时间规整算法，最长公共子序列算法和编辑距离算法一般针对的是完整轨迹，最后计算结果得到的是两条完整轨迹的距离或者表达轨迹相似程度的数值。比如给定一条查询轨迹Q，一条数据库中的数据轨迹R，通过相似性计算函数得到的是轨迹Q和轨迹R从起始位置到终止位置所有样本点按照时间顺序连成轨迹段的相似性。但是实际情况下，轨迹数据库中的单条轨迹在很大部分轨迹段上与查询轨迹Q并不相似，但是有一小部分，比如三分之一的长度和Q是很重合的，有的时候这三分之一的轨迹包含的信息远大于另外三分之二的轨迹，但是之前整段轨迹比较相似性的一个缺点就是其余三分之二的不相似的轨迹段容易掩盖掉这三分之一的特征，因此我们需要额外使用一个方法，将这最相似的三分之一的轨迹段找出来。

举一个例子来说明这个问题。比如为用户推荐顺风车，用户需要的路线是顺风车CarA预计行驶轨迹的三分之一，另外一辆顺风车CarB的预计行驶轨迹与用户需要的路线很接近，但是大于CarA与用户接近的那部分轨迹距离，那么最好的情况应该是为用户推荐CarA。但是如果使用之前轨迹相似性算法计算轨迹全段的相似程度，可能出现由于CarA的轨迹很大部分与用户轨迹不相似，导致掩盖了相似的也是用户最需要的那部分结果，而为用户推荐了CarB的结果。这并不是使用相似性算法的本意，而计算完整轨迹的相似性计算没法解决这个问题。此外，顺风车的推荐还需要考虑一个时间因素，如果下图中CarA和CarB的轨迹是一小时之前，而用户发出呼叫顺风车的请求是在现在，那么任何一辆车都不能推荐给用户。针对这种时效性的问题，还必须考虑轨迹之间的时间距离。之前的许多研究都涉及到了时间方面的因素，比如将两点之间时间戳的差值作为时间距离，最后使用一个系数和空间距离结合起来，得到一个轨迹间的相似程度。但是大多数情况是根据算法研究人员经验值去确定参数值，并且不同规模的轨迹可能还需要调整参数的大小，参数的确定很难给出一个明确的意义。



国内外很多专家学者对轨迹相似性进行了深入的研究，使用了不同的空间网络、不同的轨迹格式表示以及不同的维度企图去找到一种更好地方法去表示出轨迹之间的相似程度。但是由于研究的问题会涉及到具体的场景，由于大家研究的问题不尽相同，所以研究出了很多的相似性表示方法。首先空间网络有欧式空间下和路网下的轨迹相似性，这两种研究场景最大的区别在于欧式空间不考虑道路对移动的限制，可以使用一条直线距离去衡量任意两个地点的距离，而由于路网空间对人员和车辆的限制，两个地点之间的距离必须使用真实的道路距离去衡量。还有轨迹格式会根据研究的问题不同而选取不同的格式，比如在欧式空间下采用网格表示轨迹，还有不考虑时间因素的情况下，轨迹点信息中仅包含经纬度的信息，如果考虑时间、空间、速度和运动方向的话，轨迹点的格式就会同时包含经纬度、时间戳、瞬时运动速度以及运动的方向。而本文考虑到欧式空间对于研究的便捷性以及计算的高效性，所以不考虑道路交通情况，而是研究欧式空间下包含时间、空间、运动方向等信息的轨迹相似性。

## 1.2本文研究内容

尽管在轨迹相似性方向上已经有很多研究成果，但是上文中提到的两个问题，几乎没有一个很好的解决方案。第一个问题是轨迹的时间距离在与空间距离结合的时候，普遍使用的参数结合的方法不能赋予参数一个明确的含义。第二个问题是以往的相似性函数忽略了局部相似的情况。为解决以上两个问题，本文采用了标准空间去结合包括时间和空间在内的各个维度，并提出基于标准空间的相似性计算方法。具体包括以下几个方面：

（1）标准空间的定义。本文基于统计学知识和欧式空间构造出了标准空间，并阐述了标准空间的特点，以及如何解决之前的研究工作中存在的问题，可以更好地研究轨迹的相似程度。

（2）在标准空间的基础上，给出各种距离的定义。包括点与点之间的标准距离，轨迹段与轨迹段之间的间隔距离，以及轨迹段在形状上的改进余弦距离。标准空间中的距离可以包含构造标准空间中的每一个维度表达的信息，更好地表示空间中的轨迹段的相似程度。

（3）基于标准空间的局部轨迹相似性算法的研究。研究的核心问题是如何利用标准空间的特性，在很长的数据轨迹中找出与查询轨迹最相似的部分。设计了局部轨迹相似性查询算法，找出最相似部分轨迹并表示数据轨迹与查询轨迹的相似程度。

（4）算法实现及实验设计。实现局部轨迹相似性查询算法，并设计实验，使用微软提供的真实轨迹数据集去验证算法的高效性和准确性。

## 1.3本文组织结构

# 第二章 相关理论与关键技术

本章主要介绍一些轨迹相似性相关理论与关键技术，包括轨迹的表示方法、现有的一些轨迹相似性查询算法以及一些加速轨迹相似性查询的索引技术。

## 2.1轨迹表示方法

移动对象在移动过程中，我们对其按既定的规则进行采样，会获取一系列采样点，这些采样点再按照时间先后进行排序，就会大致还原移动对象的移动过程。实际情况中，轨迹是移动对象的连续移动过程中采样得到的一系列离散的位置序列。并且在不同的场景下，为了达到不同的目的，我们需要采用不用的轨迹表示形式。下面将介绍一些常用的轨迹表示方法。

### 2.1.1仅基于位置信息的轨迹表示方法

一般情况下，轨迹可以表示为一系列包含信息的点组成的有序集合，即T = <p1, p2, …, pn>**错误!未找到引用源。**。在欧式空间下，点pi是一个坐标的形式，pi = <loni, lati>，loni表示经度，lati表示纬度。在路网空间下，我们可以将路网模型化为图的数据结构的形式，即G = (V, E)，此时点pi表示的就是图G的顶点集合V中的一个点。其实在真实的路网环境下，采样得到的点也是用坐标形式表示的，我们会使用map-matching算法将采样点映射到路网模型的顶点中。单纯地记录轨迹位置信息的优点是简单方便，没有太多的数据冗余，并且可以使用多种简单的相似性度量方法来计算轨迹相似性，比如DTW、LCSS和EDR，这些相似性度量方法针对的都是仅包含空间信息的轨迹，简单高效。

### 2.1.2基于时间信息的轨迹表示方法

我们都知道，轨迹信息有空间尺度和时间尺度，但是上面对轨迹的表述方法仅仅考虑了轨迹的空间尺度，没有考虑到移动物体在采样点空间位置下的时间信息。在研究用户的移动模型或者预测用户下一时刻的位置等情况下，为了更准确地得到研究结果，我们一般会在记录用户位置信息的同时，记录下用户处于该位置的时间信息。我们可以将含有时间信息的轨迹表示为T = <(p1, t1), (p2, t2), …,(pn, tn)>，pi表示坐标或者顶点集合中的点，ti表示的是用户处于pi位置的时刻。记录下轨迹的时间信息比单纯记录空间位置的轨迹应用地更广泛，不但可以更加详细的描述原始轨迹，还可以使用一些对时间信息敏感的轨迹相似性度量方法对轨迹进行相似性度量，获得更为准确和有效的相似性结果。

### 2.1.3基于文本信息的轨迹表示方法

在某种特殊情况下，我们可能无需考虑轨迹的时间信息，甚至空间信息也不是首要考虑的，但是我们需要好好利用轨迹的文本信息。比如我们要开发一个推荐系统，通过研究用户的个人偏好和个性化的要求，然后给出合理的推荐方案。在生成推荐方案时，为了结合用户的偏好和要求，我们需要考虑轨迹的文本特性，由此产生了基于文本特性的轨迹信息。基于文本特性的轨迹可以表示为T = <(p1, k1), (p2, k2), …,(pn, kn)>，pi表示坐标或者顶点集合中的点，ki表示pi位置的文本描述**错误!未找到引用源。**。

### 2.1.4基于网格的轨迹表示方法

网格表示法也是一种常见的轨迹表示方法。将图平面按照一定规则划分成网格，采样点的id用所在网格的id表示，同时记录下该采样点进入所在网格和离开所在网格的时刻。网格表示法的一般表示形式为：TR={(c, I)|cC, I=(tin, tout)}，其中c代表整个网格集合C中的一个网格，tin和tout分别表示进入和离开网格的时刻。

## 2.2现有轨迹相似性计算方法

### 2.2.1欧氏距离（EU）

欧氏距离的计算首先要得到两条轨迹的对应点，按照时间先后，一一对应，如图2.1所示。



图2.1 欧氏距离一对一的对应关系

然后将所有对应点的欧氏距离进行综合处理，可以求和、求均值、取中值等，下面以求和的方式举例。给定两个一维的时间序列，，序列A和序列B的欧氏距离表达式如式(1)所示。欧式距离实际上是Lp-norms在p=2情况下的一个特例。Lp-norms定义如式（2）所示。当p=1时，L1-norms叫做曼哈顿距离。

(1)

(2)

欧氏距离有很多优点，比如计算简单，长度为n的两条轨迹，可以在时间内计算出它们的相似度，而且它满足三角不等式，如式（3）所示，运用三角不等式可以计算两条轨迹之间的距离下限，从而可以进行高效的轨迹查询。

(3)

虽然欧氏距离计算十分简单，时间复杂度低，但是缺点也是显而易见的。第一，使用欧氏距离的前提就是两条轨迹必须要拥有相等的长度，因为欧氏距离的公式决定了两条轨迹必须使用相对应的点来进行计算二维距离。第二，欧式距离不能处理局部时间偏移，局部时间偏移是指由于采样策略或对象移动速度的不同，轨迹上的样本点不能在时间上一一对应，在另一条轨迹上的对应点可能是一段时间之前或者一段时间之后的。第三，使用欧氏距离进行相似性计算容易受到噪声的影响，因为在欧氏距离的计算中，轨迹中的每个点对应到另一条轨迹上的点，如果有噪声点，那么噪声点对最后结果会产生一定的影响，带来更大的距离。随着数据量的变大和研究的深入，我们现在一般用此函数对轨迹数据进行预处理，利用其时间代价低的优点，起到一个初步筛选的作用。

### 2.2.2动态时间弯曲距离（DTW）

由于样本点采集设备的误差等原因，两条轨迹数据的样本点在时间上不能一一对应，会产生局部时间偏移的问题，只有将轨迹在时间维度上进行拉伸之后才能进行有效的相似性计算。动态时间弯曲距离将计算两条轨迹中最小对应距离之和，而不是按照时间关系一一对应，如图2.2所示。



图2.2 DTW一对多的对应关系

DTW的二维空间上的计算公式如式(4)所示。其中m和n分别表示轨迹A和轨迹B的采样点的个数，即轨迹长度。Head()函数表示轨迹的第一个采样点。d(Head(A),Head(B))表示轨迹A和轨迹B的第一个采样点之间的欧氏距离。而Rest()函数表示轨迹除去第一个点剩余的部分。动态时间弯曲距离的公式是用递归定义的，公式的含义是轨迹A和B的第一个采样点之间的欧氏距离加上轨迹剩余部分的最小的一个DTW值，直到轨迹剩余部分长度为零。下面所有公式中涉及到的Head()和Rest()函数和动态时间弯曲距离中的Head()和Rest()函数意义相同。

(4)

由于动态时间弯曲距离可以通过复制某些点来解决局部时间偏移的问题，弥补了欧氏距离只能处理等长的轨迹数据的缺点，所以应用范围比欧氏距离更广。但是动态时间弯曲的时间复杂度是，计算代价比欧氏距离大。此外，与欧氏距离一样，计算动态时间弯曲距离时，每一个点都会被强制性找出其对应点，所以也会产生噪声干扰的问题。

### 2.2.3最长公共子序列距离（LCSS）

顾名思义，最长公共子序列距离计算的是两条轨迹中最长的公共子序列的长度，以此来表示两条轨迹的相似度，计算公式如公式(5)所示。

(5)

其中：

实际上，最长公共子序列距离表示的并不是空间距离，而是“得分”，两条轨迹的得分越高，表示它们相似度就越高。由计算公式可知，在递归过程中，每当两条子轨迹的第一个采样点的欧氏距离小于一个阈值，并且两段子轨迹的长度在一定的阈值以内，就认为这两个点是匹配的，可以给当前结果加一分，继续取二者的子轨迹进行递归，否则就取子轨迹组合中最大的得分，直到子轨迹的长度为零。

相比较前面介绍的两种函数而言，最长公共子序列距离可以有效地避免噪声的干扰。因为噪声点对应到另一条轨迹上时，距离会大于阈值，噪声点将不会匹配上另一条轨迹上的点，从而排除了噪声点的干扰。在时间复杂度上，最长公共子序列距离和动态时间弯曲距离一样，也需要的时间开销。

### 2.2.4编辑距离（EDR）

EDR**错误!未找到引用源。**的核心思想是从字符串领域借鉴来的。为了判断两个字符串之间的相似程度，根据对其中一个字符串做增加、删除和修改操作，其中删除一个字符串中的字符可看做是在另一个字符串的增加字符。增加字符的操作是为了使两个字符串序列长度相等，我们把增加的字符叫做间隙元素（gap）。两个字符串之间的距离如式（6）所示。

(6)

然而时间序列中的元素是实数，有时候不会像字符那样完全相等，所以当两个实数之差小于阈值时，我们就认为这两个实数相等，因此时间序列中元素之间的距离如式（7）所示。EDR是基于时间序列中元素的距离distedr得到的，如式（8）所示，其中序列R和S的长度分别是m和n，Rest（R）和Rest(S)是序列R和S除第一个元素以外的剩余元素。EDR能够处理时间序列偏移的能力就是由于当r1和s1不相等时，取值为R、S和其剩余部分相结合EDR的最小值，从而匹配了最合适的点对。

(7)

(8)

### 2.2.5带真实惩罚的编辑距离（ERP）

ERP**错误!未找到引用源。**是L1-norms和EDR的一个结合，在计算两个元素之间距离的时候，当遇到两个非间隙元素时采用元素间真实的L1-norms距离而不是0，当其中有一个元素是间隙元素时，利用一个常数g来参与L1-norms距离计算，因此ERP的计算结果中包含了两条轨迹之间真实的距离。ERP中两个序列中元素的距离表示如式（9）所示。基于disterp的ERP计算公式如式（10）所示，类似于EDR的计算方法，当序列R和S长度均不为0时，ERP将计算R、S与其剩余部分结合的ERP最小值，因此ERP同样可以处理局部时间偏移。

(9)

(10)

### 2.2.6路网上的轨迹相似性查询

当将一段轨迹展示在路网中时，考虑到实际道路情况，点与点之间不一定有直线道路相连，使用二维空间的Lp-norms即欧氏距离来计算两条轨迹的相似度和实际相似情况可能相差较大。因此我们需要重新定义一个适用于路网的相似性函数。

在计算相似性之前，需要将轨迹映射到路网上去，如图2.5所示。给定移动对象a和b的移动轨迹Ta和T b，轨迹格式为T={(l1, v1, t1), (l2, v2, t2),…(lm, vm, tm)}，其中li=(lgi, lai)表示样本点坐标，vi表示对象在ti时刻的移动速度。我们用da(lai,Tb)表示从样本点lai到轨迹Tb的路网距离。lai到Tb路网距离指，Tb上距离lai最近的点到lai的路径距离。用DG表示路网G的直径。在不同条件下，对图G中的轨迹进行相似性查询，我们有不同的距离函数。

图2.5 将道路转化为路网

路网上Ta和Tb的距离dN(Ta, Tb)的计算公式如式（14）所示。其中，轨迹Ta叫做查询轨迹，轨迹Tb叫做目标轨迹，m代表轨迹Ta的样本点个数。

(14)

当查询用户有对兴趣点的查询需求时，式中的样本点将会拥有权值wai，wai的大小代表样本点在相似性查询操作时不同的重要性。带权值的轨迹距离公式如式（15）所示。

(15)

当需要对路况信息或者交通拥堵信息进行分析时，轨迹的实时速度信息就十分重要了。令SG代表当前道路最高限速，ds(lai,lbi)代表轨迹A上的第i个样本点lai与轨迹B上的第i个样本点lbi的速度之差，其中lbi是轨迹B上到lai最近的样本点。带速度信息的轨迹距离公式如式（16）所示。

(16)

分析路况信息时，时间信息也是一个很重要的因素。我们令代表轨迹A上的第i个样本点lai与轨迹B上的第i个样本点lbi的时间之差，其中lbi是轨迹B上到lai最近的样本点。la1是轨迹Ta的第一个点，lam是轨迹Ta的最后一个点，lb1是轨迹B上距la1最近的点，lbm是轨迹B上距lam最近的点。带时间信息的轨迹距离公式如式（17）所示。

(17)

当结合权值、时间和空间来计算轨迹距离，我们有公式（18），其中和分别代表对应的子计算方法的权值参数，并且两个。如果将速度信息加入公式，公式如式（19）所示。

(18)

(19)

上述五种情况的基本思想是首先找出对应点，给定查询轨迹上的点，它的对应点是目标轨迹上到该点路网距离最近的点。然后考虑每一组点对之间距离、权值、速度和时间占总体的比例，最终得出以上公式。优点是并没有采取按照时间来匹配对应点的想法，计算结果可能更符合原始相似情况，并且计算思路简单、清晰，扩展方便。缺点是样本点权值的选取没有给出一个具体的方法，可能会使计算效果不好。

### 2.2.7城市运输系统中基于段的轨迹相似性计算

轨迹点是根据一个给定的采样方法获取的，不同的采样方法给轨迹相似性计算带来了很大的影响。比如两条完全相同的轨迹，但是由于采样开始时间不同，就造成了样本点的错位，如图2.6所示。



图2.6 不同采样方法造成的样本点错位

传统的相似性计算方法，比如DTW就没有考虑这个问题。LCSS忽略了轨迹的空间距离，EDR没有考虑到轨迹的形状因素。由于传统轨迹相似度计算方法计算结果的不准确，所以Mao等人提出了基于段的轨迹相似性计算方法**错误!未找到引用源。**。

首先介绍点段距离的概念。点段距离是两条轨迹对应点之间的特殊距离，表示为图2.7中的阴影面积，由样本点R、S以及各自前后样本点的中点连接而成。由于不规则阴影面积的计算比较复杂，而阴影部分正比于图2.8中两个虚线三角形的面积之和，所以将阴影部分面积的计算转换为图2.8中三角形面积的计算。但是当seg1和seg2很长时，效果并不好，所以用三角形的高，p1到seg2的距离和p2到seg1的距离来代表p1与p2之间的距离，如图2.9所示，其中为p1到轨迹S的点段距离。对应的距离公式如式（20）所示。

图2.7 使用轨迹段面积代替点点距离 图2.8三角形面积正比于阴影面积



图2.9 使用三角形的高代替三角形面积

(20)

其中，

然后介绍预测距离。给定一组对应点和，时间戳分别为ti和tj，ti和tj不相等，令时间戳大的点等于A，时间戳小的点等于B，如图2.10所示。



图2.10 预测B在与A相同时刻的位置

然后利用轨迹R在ti-1时刻的位置和ti-1与ti间的平均速度，来预测tj时刻的位置，得到，的预测距离计算公式如式（21）所示。

(21)

其中，

那么这两个点的时间距离可以转化为tj时刻，轨迹R的位置到的距离，如图2.10所示。的预测距离计算公式如式（22）所示。

(22)

图2.11中，融合点段距离和预测距离，我们可以得到对应样本点和之间的时空距离公式，如式（23）所示，其中t是时间距离的敏感参数，值越大表示时间距离越重要。

(23)



图2.11 利用夹角计算形状相似性

利用样本点之间的时空距离，可以得到该样本点和后一个样本点形成的轨迹段之间的距离，即段段距离，来计算轨迹段和轨迹之间的形状相似程度，如式（24）所示，其中是与之间的轨迹段，是与之间的轨迹段。

(24)

在判断两条轨迹的相似程度的时候，形状上的相似也十分重要。结合形状因素的段段距离如式（25）所示，其中是两条轨迹段之间的夹角，和的计算公式如式（26）和式（27）所示。

(25)

(26)

(27)

前面得到了结合形状因素的轨迹段之间的时空距离，我们将其当做一个距离计算因子，代替DTW函数中使用的对应点之间距离，可以得到式（28），其中指的是轨迹R的第一个样本点和第二个样本点之间的轨迹段，指的是出掉第一个轨迹段剩下的所有轨迹段。

(28)

该方法主要优点有三个：

（1）改进DTW函数，采用轨迹段到轨迹段的距离来代替点到点的距离计算，可以减少对轨迹采样方法的敏感程度。

（2）利用预测的方法，对于一个点对，预测时间戳靠前的点在下一刻的位置，使得两个点时间戳相同，将时间距离转换为空间距离，考虑到相似性计算中去。

（3）将形状因素加入到相似性计算中，提高形状相似性方面的精度。

### 2.2.8简化的轨迹相似性计算

由于利用轨迹相似性可以进行未来某一时刻的位置预测，Liu等人首先提出了基于社会传染理论的位置预测算法，然后提出了一个简化的轨迹相似性计算方法来找出带预测用户的相似用户组，以此支撑社会传染理论的运行，

根据用户在某地活动消耗的的时间比仅仅路过该地的时间长，并减少相似性计算的复杂度，我们将完整的轨迹分解成小轨迹。分解条件就是用户的时间跨度，是一个时间阈值。分解后得到用户的小轨迹集。我们可以得到所有用户的小轨迹集。

然后两个小轨迹集之间的重合程度来计算他们的相似程度。令代表用户的布尔型向量，的值如式（29）所示。

(29)

当比较用户和用户之间的相似程度时，将和做与运算（），取1的个数作为轨迹之间的相似度值。

这个方法的优点是相似度计算过程简单快速，没有过多地求轨迹之间的空间距离以及考虑时间、速度等因素。缺点就是计算结果可能随阈值设置的好坏而变化，若设置过大，导致小轨迹过长，会让两条相似的轨迹重合的小轨迹变少，从而不能反映真实的相似程度，精度不够。

### 2.2.9基于签名的轨迹相似性计算

当前的轨迹相似性函数很大程度依赖两条轨迹的对应样本点，然后计算对应样本点之间包含各种信息的距离，但是由于采样频率或者物体移动速度不同，样本点很可能不能一一对应，如图2.12所示。因此Ta等人提出了BDS**错误!未找到引用源。**计算两条轨迹之间的相似程度。与找轨迹的对应点的方法不同，BDS在计算轨迹Ti和Tj的相似度时，通过累加Ti的每个样本点到Tj的最短距离，如图2.13所示。

图2.12按时间匹配点对 图2.13按点到轨迹段最小距离匹配

这个最短距离的定义是将Tj上的所有样本点按照时间顺序连线，形成Tj-1条轨迹段，Ti上的第k个样本点到这Tj-1条轨迹段的最短距离就是Ti上样本点到Tj的距离，其计算公式如式（30）所示。

(30)

而Ti上样本点到轨迹段的最短距离分为两种情况，如果点到轨迹段的垂线与轨迹段相交，距离就是垂线的长度，否则就是样本点到轨迹段里自己最近的端点的距离，如图2.14所示。



图2.14点到轨迹段的距离

BDS的计算公式如式（31）所示，其中是一个归一化的距离，最终结果是一个介于0到1之间的值，表示轨迹和轨迹之间的相似程度，值越大，表示两条轨迹越相似。

(31)

其中，

该相似性函数的计算量很大，对于两条轨迹，要计算出每一个样本点到另一条轨迹所有段的最短距离。为了弥补这个缺点，使用该方法之前需要利用网格作为轨迹签名，先进行一次筛选，然后在计算轨迹上第k个点到轨迹的距离时，只需要计算点以该点为中心的一定范围内的网格中存在的轨迹段的距离，极大程度地减少了计算次数。

这个相似性计算方法方法的优点是不用硬性地将两条轨迹中的点进行配对，一个样本点的对应点可能在另一条轨迹段中两个样本点之间，将两条相似的轨迹更好地进行吻合。缺点就是由于仅考虑了两条轨迹的空间位置，没有考虑其他信息，比如时间和速度信息，因此仅适用于比较两条道路的相似性，不能完整地反映移动对象的详细信息。

## 2.3 高效的轨迹索引技术

### 2.3.1 R树索引

R树是一种针对空间数据建立的树形索引，其核心思想是将每片需要建立索引的区域使用一个外接矩形框起来，作为树的叶子节点，即叶子节点包含着指向该片区域的指针。然后在建树的过程中，矩形之间允许多层嵌套，每个矩形都是一个节点，外层的矩形是被嵌套的矩形的父节点，最终构造成一颗高度平衡树。解决了传统数据库对空间数据检索效率低下的问题。

由于R树需要构建成一颗高度的平衡树，所以有一个构建规则。假设M是树中所有节点允许包含的最大的条目数，m是节点包含的最小条目数，通常情况下m=M/2，此时R树的构建规则如下：

（1）所有叶子节点包含m至M个记录索引。如果一个叶子节点是树的根节点，那么它包含的记录数可以小于m。

（2）叶子中存储的指针指向的记录，这些记录均为可以覆盖其代表空间中区域的最小外包矩形。

（3）一个非根且非叶子的节点，其包含的孩子数目在m到M之间。

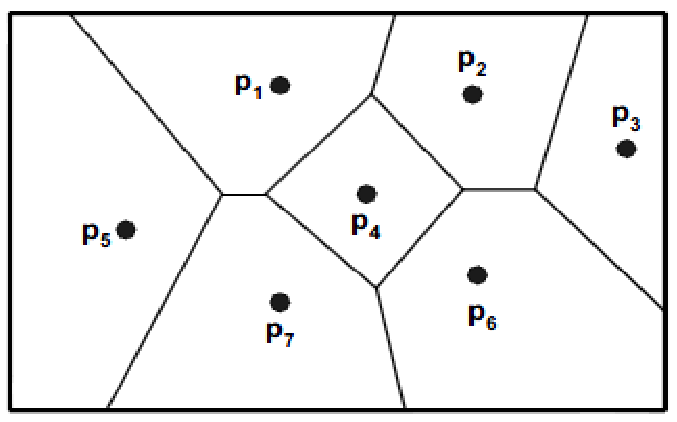
（4）对于一个非叶子节点，节点中的记录是节点所有孩子节点代表区域的最小外包矩形。

（5）R树的所有叶子节点均在同一层。

由于以上特性，相比传统数据库，R树可以很快检索空间数据。但是R树允许外包范围重叠，所以使得一个区域可以对应多条搜索路径，严重影响了R树性能，使空间搜索效率降低。因此R+树和R\*树和三维R树被提出来解决搜索区域的重叠问题。

### 2.3.2 Voronoi图

Voronoi图根据空间中的点集的分布，将一个欧式空间划分为多个凸多边形，凸多边形的每一条边都是一个点与周围点的垂直平分线，该点叫图多边形的质心。每个凸多边形都有一个质心，任意一个凸多边形内的点到其他凸多边形质心的欧式距离都小于到本凸多边形质心的欧氏距离，如图所示。后来有学者对Voronoi图进行了进一步研究，根据发生点和生成面的不同，提出了广义Voronoi图，其中包括有序k阶Voronoi图，最远点Voronoi图，权重Voronoi图等。Voronoi图作为空间数据索引中的一类重要索引结构，解决了很多传统数据库不能解决的问题。



## 2.4本章小结

本章主要介绍了不同的轨迹的表示方法以及为解决不同问题而提出来的一些轨迹相似性计算方法。有些相似性使用范围较为广泛，比如DTW、LCSS和EDR，但是这些算法不针对某个具体场景或具体问题，并没有研究具体情况下该引入哪些特征去描述轨迹或者该怎么去优化轨迹间的相似性表示。因此现在轨迹相似性的研究更加偏向于研究某个具体场景下的轨迹相似性，比如城市运输系统中基于段的相似性查询，和路网上的轨迹相似性查询等。此外，本章还介绍了加速空间数据搜索的几个应用比较广泛的索引结构。

# 第三章 问题定义

3.1

# 第三章 标准空间下的轨迹段相似性

## 3.1 数值不均衡问题的提出

### 3.1.1 数值不均衡问题的定义

数值不均衡问题，即时空数据某一维度数值跨度很大，而相比而言另一维度的跨度很小。例如轨迹开始与终止时间跨越为十二个小时，即43200秒，空间距离跨度为4000米，从数值上看，时间维度的数值远大于空间维度的数值。或者汽车从北京（116.408598, 39.923079）开往南昌（115.700302, 28.934003），在经度上跨度很小，约为120千米，而纬度上跨度很大，约为1320千米，纬度的跨度远大于经度跨度。

### 3.1.2 数值不均衡问题的影响

前人的一篇论文（Personalized trajectory matching in spatial networks）中使用以下的方法计算轨迹间距离，假设查询轨迹q和数据轨迹t上对应点为v1和v2。论文使用空间距离为v1和v2的路网距离，时间距离为v1和v2的时间戳之差，然后将空间距离和时间距离作为指数函数的指数部分，获取空间影响因子Is（v1,v2）和时间影响因子It（v1，v2），如公式（2）和（3）所示，该影响因子代表样本点之间的时空相似程度。在相似度函数的计算中，利用LCSS的方法，找出所有样本点对以获得最大空间和时间相似度相似度，如（4）和（5）所示，其中权重q.head.w代表点q在相似度计算中的重要程度，也是文章的创新点之一。最后使用参数将相似度结合起来，得到轨迹相似度，如公式（6）所示。









该方法的问题是没有考虑到相同差距值在不同维度具有不同的信息，忽视了空间上的单位米和时间上的单位秒的概念，将距离和时间简单作为一个数值带入公式中进行影响因子的计算。然后论文认为可以仅根据用户的时间相似度和空间相似度的重要程度确定参数。但是数值上大小相同的空间和时间影响因子，可能描述了轨迹不一样的相似程度。

假设两条轨迹的一对对应点v1和v2，空间距离是5米，时间戳差距是0分钟，另外一对对应点v3和v4空间距离是0米，时间戳差距是5分钟。此外两条轨迹在距离上首尾跨度1200米，时间跨度是10分钟。如果的确定不考虑轨迹时空跨度，即时空数据的不均衡，那么假设用户认为时间和空间因素同等重要的情况下为0.5，通过计算得到的Is(v1,v2)和It(v3,v4)数值相等，It(v1,v2)和Is(v3,v4)数值相等。但是1200米的轨迹上对应点之间5米的差距是很小的，而相比而言10分钟的跨度内，有5分钟的时间差距是很大的，计算结果很明显与用户的需求不符合。

与之类似的另一个问题，在欧式空间下，很多方法直接使用欧氏距离作为点与点之间的空间距离。而欧式空间中的x轴和y轴是两个不同的维度。如果轨迹在x轴上跨度很大，而在y轴上跨度很小，那么对应样本点在x轴上的距离差距和在y轴上的距离差距也不可以简单使用各自数值去计算。

## 3.2 标准空间

### 3.2.1 标准空间的定义

出现上述问题的关键在于之前的方法忽视了不同维度的数值传达信息的不同。上述论文中的方法在下面的情况下，影响不是很大，以二维空间举例，当轨迹沿着与经线和纬线夹角为45度的方向延伸时，轨迹在两个维度有相同的计量单位，并且有着相差不大的数值范围，此时使用欧氏距离可以描述轨迹的相似程度。但是下面两种情况，之前的方法不能很好的适用。第一种情况是轨迹与经线或纬线夹角很小，会使得轨迹在经纬度上的距离值相差很多倍。第二种情况是加入了时间维度，首先与空间维度的计量单位不同，那么相同的差值，表示的相似度信息也是不同的。

为了解决上述问题，本文提出一个标准空间的概念，在标准空间中，所有维度的计量单位均相同，不同维度可以在点与点的距离计算中直接进行运算，包括空间和时间。目的是消除量纲和数值不均衡对距离计算的影响。

### 3.2.2 标准空间的构建方法

在介绍标准空间的构造方法之前，我们需要规定一下本文中使用到的轨迹格式，本文需要使用时间和空间信息将轨迹映射到标准空间，所以轨迹的样本点信息中需要包含经纬度和时间戳信息。后文虽然还用到了方向，但是方向是依据前后两个样本点计算得到的，所以样本点信息中不需要方向信息。因此轨迹中样本点的信息格式为

假设有查询轨迹Q和数据轨迹R，，，有下面两个方法可以将Q和R向标准空间转化。

第一个方法是正态映射法，其构造方法是首先获取两条轨迹所有样本点，计算所有样本点在三个维度下各自的均值和方差，经度、维度和时间戳的均值和方差分别为、、、、、。然后针对点两条轨迹中的每一个点p进行标准化，如公式所示。该方法得到的标准空间中，每个维度均服从均值为0，方差为1的正态分布。

第二个方法是最值映射法，其构造方法是先获取三个维度的最大和最小值，经度、维度和时间戳的最大值和最小值分别为、、、、、。然后针对所有点p进行最值映射，如公式所示。该方法得到的标准空间的特点是所有数值均分布在0到1之间。

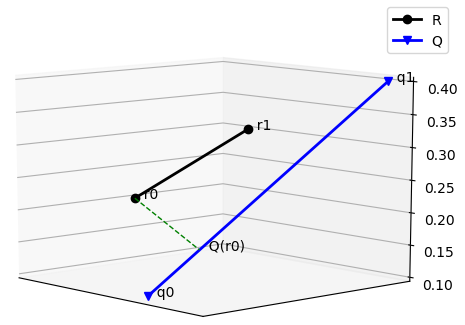
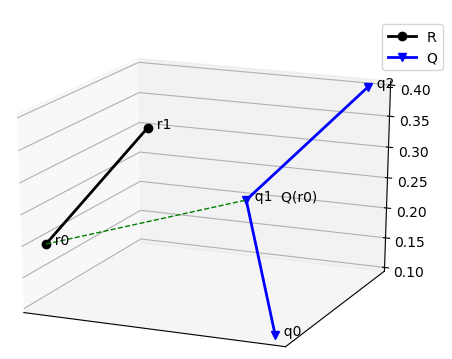
转化后点的格式为。使用上述的正态映射法或最值映射法均可得到属于轨迹Q和轨迹R的标准空间。

## 3.3标准空间下对应轨迹段的相似性

由于定义了标准空间，统一了时间维度和空间维度，允许了时间维度和空间维度进行统一运算，所以点与点之间的时空距离的计算在这里变得很方便，标准空间下所有维度的计量单位都是相同的，可以直接将时间维度纳入点对的距离计算之中，样本点之间的距离计算可以使用Lp-norm距离。本文以p=2，即欧氏距离来举例。样本点和之间距离的计算如式所示。其中和是时间维度转换过来的数值，在三维的时空标准空间中使用了与三维空间欧氏距离计算相同格式的公式，体现了标准空间对时间和空间的融合。

在介绍对应轨迹段的相似性之前，需要介绍到对应点和对应轨迹段的概念。

轨迹上的对应点指的是使用一个样本点对齐算法，将数据轨迹R上的所有样本点对齐到查询轨迹Q上去，那么R上的某个样本点会获得轨迹Q上的一个对应点，我们将在轨迹Q上的对应点记作，如图所示，的对应点是轨迹Q上的样本点。这里需要明确的是，并不一定是数据轨迹Q上的一个样本点。因为考虑减少到采样策略的不同对轨迹相似性计算造成的影响，本文中的样本点对齐算法的设计将有可能使对齐到查询轨迹Q两个样本点之间的连线上，如图所示，的对应点在轨迹Q的样本点和之间。这里需要用到的样本点对齐算法将在后文中进行介绍。



在获得对应样本点的基础上，我们可以得到对应轨迹段的概念。数据轨迹R上的连续两个样本点和，会在查询轨迹Q上分别获得他们的对应点和，那么轨迹段的对应轨迹段就是。

对应轨迹段可能会出现多种情况，我们对每种情况的处理方式不尽相同。第一种情况是和之间的轨迹段是一条直线段。其中和有可能是轨迹Q的样本点，也有可能是轨迹Q中某两个样本点连线上的点。这个情况在讲对应点已经讨论过了，如图所示，的对应点是轨迹Q的样本点，的对应点在样本点和之间。

第二种情况是由于数据轨迹Q移动速度和移动距离等因素，和可能会对齐到Q上的同一个点，即和有可能是轨迹Q中的同一个样本点，如图所示，和的对应点和为同一个点，并且该点还是轨迹Q的样本点。

第三种情况是虽然和是两个连续的样本点，但是和中间可能隔着多个样本点，如图所示，和的对应点和中间夹着轨迹Q的样本点和。







由于后面计算轨迹相似性的需要，本文提出标准空间下的对应轨迹段相似性的概念。在标准空间中，数据轨迹段和对应的查询轨迹段之间的相似性由两部分组成，第一部分是其形状上的相似性，第二部分是轨迹段之间标准空间距离。下面依次对这两部分进行讲解。

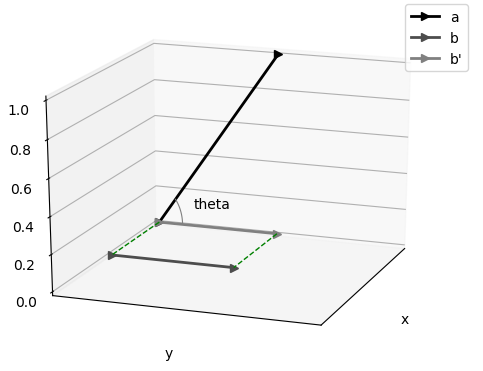
### 3.3.1对应轨迹段形状相似性

轨迹段形状相似性和轨迹段之间的夹角以及轨迹段的长度相关，轨迹段之间夹角越小的情况下，两条轨迹段长度越长，那么轨迹段形状越相似。下面将从夹角和轨迹段长度两方面讨论轨迹段形状上的相似性。

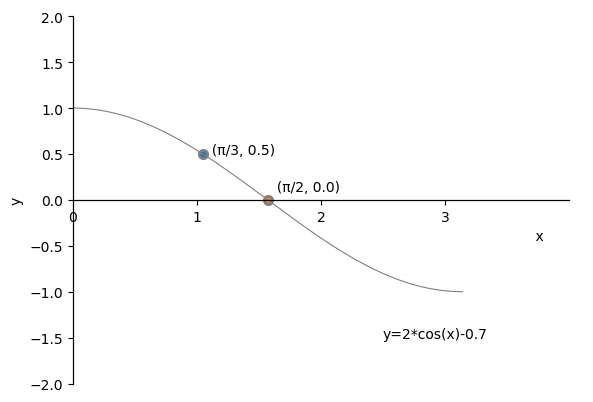
#### 3.3.1.1改进余弦距离

在我们建立的三维标准空间中，两条轨迹段可能存在异面的情况，如图所示。我们在这一小节采用向量的方式去讨论轨迹段的夹角，那么这里需要利用三维空间中的向量去求两个异面线段的夹角的余弦距离。三维空间中向量和向量的余弦距离如公式所示。在轨迹段余弦距离中，的取值范围为[0, ]，的取值范围为[-1,1]。

，其中。



但是使用余弦距离衡量一条轨迹段与其对应轨迹段之间的形状相似程度有两个问题，第一个问题是余弦距离对角度差介于和之间的轨迹段不敏感，即不能有效地对较大的锐角的轨迹段做出惩罚，如图所示，因为在轨迹的相似性里面，如果两条轨迹朝着一个方向走的时候，夹角过大，会导致轨迹形状的不相似，同时物体距离会随着时间推移越来越大，这是需要抑制的一种情况。第二个问题是两条轨迹段的相似程度不仅和角度相关，还和轨迹段长度相关。当两条轨迹段夹角为0时，在一定范围内，轨迹段长度越长，表示这两条轨迹段越相似。当两条轨迹段夹角为时，轨迹段长度越长，表示两条轨迹段不相似的程度越高。

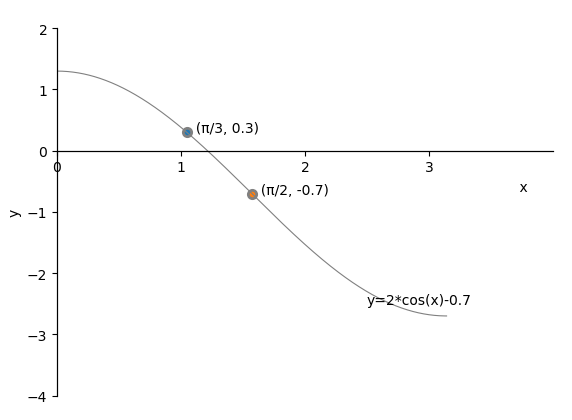


为了解决对较大锐角不能做出惩罚的问题，可以将余弦距离减去一个惩罚系数，的取值范围为，如公式所示，使得两向量夹角为较大锐角的情况下，他们的余弦距离为负值，达到了对大锐角惩罚的目的。但是这个方法一个隐藏的问题，因为时，所以的取值范围为[-1, 1]，则的取值范围为，在取0的时候，原本余弦距离为1，现在为，大于0，所以小于1。因此这个方法不但加大了对大锐角的惩罚，同时也减少了对小锐角的激励，存在一定的缺陷。

，其中

基于以上讨论，为了既可以加大对大锐角的惩罚程度，同时不降低对小锐角的激励程度，本文提出了改进余弦距离，如公式所示。改进余弦距离通过前面的倍数2加大了对小锐角情况的激励，同时使用惩罚系数也加大了大锐角以及钝角情况的惩罚。假设取，从改进余弦距离的函数图像中，我们可以看出当夹角在时，改进余弦距离为正数，当夹角在时，改进余弦距离为负数，且改进余弦距离在定义域内会随着角度的增大而减小，是一个单调递减函数，值域为[-1.7, 1.3]。

，其中



#### 3.3.1.2轨迹段形状相似距离

上一节介绍的改进余弦距离可以用来将对应轨迹段之间的夹角的相似程度，但是两条轨迹段的形状相似除了与夹角有关，还和轨迹段长度相关。

我们首先仍然假设两条轨迹段均为直线段，即如图所示的第一种情况。因此在获得可以获得对应轨迹段之间的夹角和轨迹段的长度后，我们得到了轨迹段与其对应的直线段的轨迹段的形状相似距离，如公式所示。

在轨迹段之间夹角很小的情况下，在一定程度上，数据轨迹段的长度越长，代表两条轨迹越相似。但是如果超过了这个程度，即比的长度大很多倍时，对应轨迹段之间的相似性如果再随着轨迹段长度的增长而增加，就不符合我们对轨迹形状上的相似性的要求了，因为在轨迹段形状相似性的要求是，在小锐角的情况下，两条轨迹段的长度越长越相似。因此不能仅根据一条轨迹段的长度边长去延伸。也就是说对数据轨迹段长度的激励需要有一个限制。

而依据余弦距离的几何意义，为映射到方向上的距离，这个距离带有方向性，如果夹角为钝角，距离就是负数。那么如果将余弦距离替换为改进余弦距离，也是映射的一种方式，那么的值就是由映射到方向后的值，使用改进余弦距离投影后，如果距离特别长而对应轨迹段的距离特别短，投影结果不能反映轨迹段之间长度的差异，为了抑制这种情况，在轨迹段形状相似距离计算时，我们将的长度规定为轨迹段形状相似距离的上限，即形状相似距离允许的最大激励不得超过轨迹段的长度，再大的部分我们认为是无效部分。

基于以上讨论，我们给出与对应轨迹段的轨迹段形状相似距离的计算公式，如公式所示。由公式可以看出，轨迹段之间夹角为一个小锐角时，两条轨迹段长度越长，二者相似距离越大。

其中，

上面讨论了轨迹段的对应轨迹段是直线段的情况，我们还需要对另外两种情况进行讨论。

当出现第二种情况，即和是轨迹Q中的同一个样本点时，由于此时为0，所以这种情况对应的轨迹段形状相似距离的最大值不超过0。计算具体的值时，由于和是同一个点，不能直接使用该点去计算余弦距离，首先需要找到轨迹中与相邻的两个样本点，分别记为和，由于在这两个样本点的中间，所以这一点处的方向应当看做前后两条轨迹段方向的平均值。因此当和处的夹角是一个小于的角时，由于受限于为0，所以轨迹段形状相似距离为0，当和处的夹角是一个大于的角时，轨迹段形状相似距离为负数。

当出现第三种情况，即和轨迹段中间经过一个或多个样本点，此时不是一条直线段，不能使用上面的公式直接进行计算。如图所示，和中间间隔着和，此时需要利用寻找对齐样本点的算法，在直线轨迹段上找到和的对应点，记为和，然后将、和视为三个独立的轨迹段，其对应轨迹段分别为、和。此时所有对应轨迹段均为直线段，然后计算每一对对应轨迹段的轨迹形状相似距离。如果与某一个端点重合，比如就是，那么根据的公式，因为为0，所以此处轨迹段形状相似距离为0。最后，将每一段独立的对应轨迹段之间的轨迹段形状相似距离相加，得到轨迹段和之间的轨迹段相似距离，如公式所示，其中，至是与之前间隔的样本点，m<n。



之前也有很多论文研究过轨迹段形状上的相似性给轨迹相似性带来的增益，比如前人的一篇论文中，提出Segment-Segment Distance的概念，使用一个与轨迹段之间形状相似程度相关的函数作为轨迹段距离的参数，如公式所示，描述的是对应轨迹段之间形状上的不相似程度，轨迹段越不相似，的值越大，就会让时空距离乘上更大的系数，使最后描述轨迹之间距离的变大，代表轨迹相似性越低。其中和是对应的轨迹段，是轨迹段，是其对应轨迹段。公式中使用到的角度和本论文中使用的角度含义相同，均为两条有向轨迹段的夹角，范围在[0,]，只不过该论文描述的是二维空间的情况，而本篇论文是在三维标准空间中进行讨论，无其它区别。

其中，



该论文中使用的Segment-Segment Distance与本论文中提出的轨迹段形状相似距离有两个不同点。第一个不同点在于夹角的使用，由于该论文中形状相似函数是作为距离的参数，那么形状越相似，的值应该越小，使得轨迹段之间距离越小，因此与是正相关的关系，关于的单调递增。而本论文中由于需要获得与形状相似程度成正相关的值，采用了余弦距离的形式，使改进余弦距离是关于的单调递减函数，余弦距离的值越小表示形状越相似。除此之外，与改进余弦距离最核心的区别在于无论是锐角还是钝角，一直作为一个正数，角度大点，就大一点，角度小一点，就小一点，没有明显的对大角度的抑制。而本文需要找到轨迹段中最相似的部分，就需要使用较大的惩罚去除掉相似度较差的部分，因此改进余弦距离以弧度值作为对角度激励和惩罚的分界点，角度大于该值的，改进余弦距离为负数，最后得到的形状相似性距离也是负数，明确了对夹角过大的惩罚，最后使用局部相似度查询算法的时候，会在很大程度上过滤掉距离为负数的轨迹段，留下就的是最相似的部分轨迹段。

第二个不同点在于Segment-Segment Distance中没有考虑对应轨迹段长度这方面，仅考虑了所夹角度。本文的形状相似性距离考虑到了对应轨迹段的长度给形状相似上带来的影响。因为针对两条轨迹，需要找到其中最相似的部分轨迹段，找出的相似部分越长越好。那么本文中的形状相似就不完全由轨迹段所夹的角度决定，对应轨迹段所夹角度相同的情况下，两条轨迹段长度越长，对最后的结果越有利，考虑到这种情况，本文使用了数据轨迹段通过改进余弦距离映射到其对应轨迹段中的长度来计算其对形状相似产生的影响。

### 3.3.2对应轨迹段的标准空间距离

下面介绍一下标准空间下，轨迹段间的空间距离的计算方法。

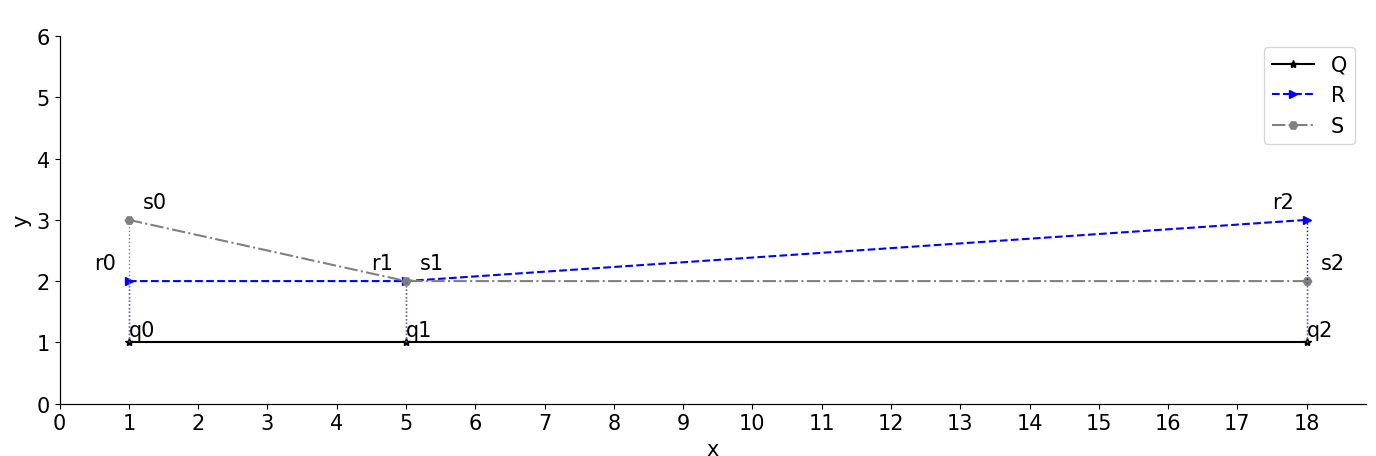
#### 3.3.2.1DTW算法存在的问题

前人对轨迹在空间距离上的计算大多从对应样本点的距离入手，比如DTW算法中是使用动态规划思想先找到最优的对应样本点分布，然后计算所有对应点的距离之和，作为轨迹之间的距离。该方法的好处是利用了动态规划思想的优点，改进了使用轨迹间的欧氏距离算法中的样本点必须一一对应的缺点，使得样本点之间允许“一对多”，解决了时间偏移的问题。这是一个最典型的求轨迹空间距离的做法，后面又有很多基于DTW算法的研究工作，将DTW中使用欧式距离去计算对应样本点的方法做出改进，其中包括引入时间和速度等其它维度的特征，构造更为复杂的距离函数，从而可以多方位地描述了样本点信息。但是这些改进算法并没有改变DTW算法的核心部分，依然是先寻找最优的对应样本点分布，然后计算所有对应样本点的距离之和作为轨迹间的距离。

虽然DTW算法应用广泛，但是也有一定的不足之处，DTW算法只考虑不同轨迹样本点之间的最优对应关系，对样本点产生了严重的依赖，并忽视了轨迹段的概念。如果一条轨迹使用无穷多个点去记录，当然DTW算法有着比较不错的表现，不但可以找到轨迹之间的最优对应点，并且最后得到的轨迹间的距离值可以准确地描述轨迹之间的距离以及相似程度。使用DTW算法计算原始轨迹的相似性的缺点是算法时间复杂度大，耗时严重，如果不考虑耗时的影响，DTW看起来是一个还不错的计算方法。但实际情况是，首先我们采样设备的采样频率是有一定限制的，不可能使用无穷多个点去描述一条轨迹。还有就是在轨迹相似性查询中，经常使用兴趣点和转折点提取算法，将一条有成百上千个样本点的轨迹数据转化为仅包含十几个或几十个点的轨迹数据，这样做的好处是在相似性查询时很大程度地保留了原始轨迹的形状，并且更关注轨迹中的兴趣点，同时减少数据量，降低了相似性计算的时间。下面我们来探讨一下因为采样策略或者提取兴趣点和转折点算法不同的原因，导致同样的轨迹，使用不同的样本点进行表示，会对DTW算法产生多大的影响。

因为DTW只考虑了空间维度，不考虑时间维度，而本文的标准空间中包含的是二维空间和一维时间，因此，为了尽量减少二者场景的不同，我们给出一个使用DTW算法计算二维空间中两条轨迹空间距离的例子。

假设我们的采样设备以一定的采样策略采集到了三个物体的运动轨迹，包括轨迹Q和轨迹R和S。其中轨迹Q包含了起点、终点以及中间的一个兴趣点，轨迹R包含了起点、转折点和终点，轨迹S也是包含了起点终点和转折点。这是三条比较简单的轨迹，该采样设备通过很少的样本点，将三个物体的移动路径准确地表达出来。



我们令轨迹Q为查询轨迹，我们想获取Q和R的相似性以及Q和S的相似性。下面通过DTW算法分别对查询轨迹Q和数据轨迹R以及S进行相似性计算，得到的距离矩阵如表所示。根据DTW距离矩阵，我们可以得到查询轨迹Q和数据轨迹R的DTW距离为4，Q和数据轨迹S的DTW距离也为4，因此，轨迹Q和轨迹R的相似程度与轨迹Q和轨迹S的相似程度相等。并且可以得到对应点的关系，分别对应和,分别对应和,分别对应和。我们会根据相似性计算结果中获得Q和R的相似程度与Q和S的相似程度相同的结论，并且根据样本点的对应关系看起来好像没什么问题。

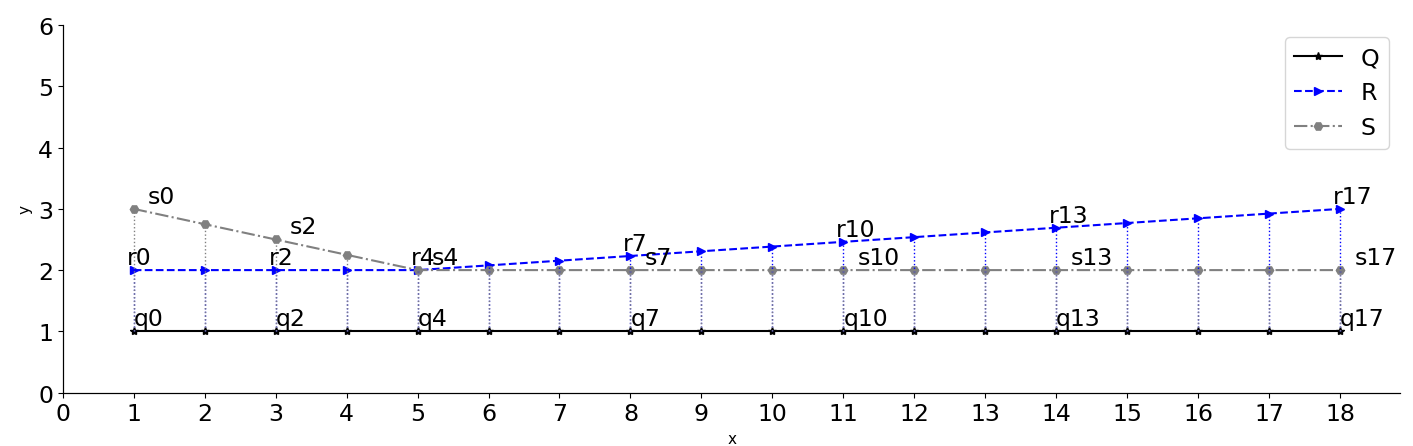
dtw(Q,R)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 0 |  |  |  |
| 0 | 0 | inf | inf | inf |
|  | inf | 1 | 5.12 | 22.24 |
|  | inf | 5.12 | 2 | 15.15 |
|  | inf | 22.15 | 15.04 | 4 |

dtw(Q,S)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 0 |  |  |  |
| 0 | 0 | inf | inf | inf |
|  | inf | 1 | 5.12 | 22.24 |
|  | inf | 5.12 | 2 | 15.15 |
|  | inf | 22.15 | 15.04 | 4 |

假设刚才产生Q、R、S轨迹的移动对象上还放着另外一套采样设备，这套采样设备的采样策略和刚才的设备稍有不同，最终采样得到的样本点稍微多一点，如图所示。虽然样本点多了，但是可以从坐标上看出，描述的还是刚才三个物体的运动，包括起点终点和转折点。



现在我们再使用一次DTW算法，分别计算出Q和R的距离，以及Q和S的距离。由于矩阵长度过大，展示不太方便，我们直接给出最终计算结果。查询轨迹Q和数据轨迹R的DTW距离为25，查询轨迹Q和数据轨迹S的DTW距离为20.5。根据计算结果，我们可以得出查询轨迹Q和数据轨迹S的距离更小，相似程度更高。

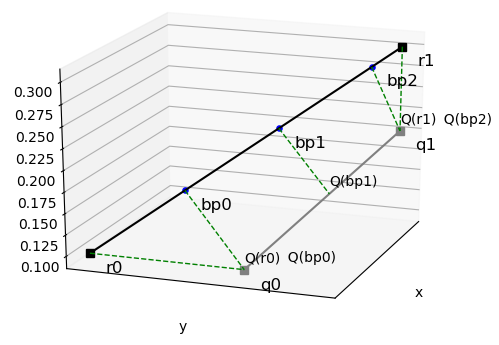
对比上述实验，使用DTW算法，对不同采样设备获取相同移动对象的轨迹数据进行相似性计算，第一份数据的结论是Q和R的相似程度等于Q和S的相似程度，第二份数据的结论是Q和R的相似程度小于Q和S的相似程度。得到了两个不同的结论。出现这个问题的原因表面上是由于采样设备的采样策略造成不同的计算结果，根本原因还是由于DTW算法对样本点以及对应样本点之间的距离过于依赖，完全使用样本点而忽视采样间隔，从而导致对采样策略十分敏感，不同的采样策略产生不同的相似性结论。

因此，仅根据轨迹数据中的少量样本点、兴趣点或转折点来计算轨迹间距离的做法是不合适的。因为单个的点只能表达了它自己的所具有的信息，没有描述出移动对象在两个点之间的移动的状态，比如移动距离。如果像第二个实验那样，样本点更加密集一点，使用更多的点去描述移动对象在和之间的移动，则会对这段直线段所包含的信息有更好地表达。

#### 3.3.2.2断点

基于以上讨论，提出一个断点（break point）的概念。我们使用一定的策略在一条轨迹段中获得其断点，断点是轨迹段上的一个点，用于将一条轨迹段分割为更短的子轨迹段。根据策略不同，可以将断点分为基于距离的断点和基于时间的断点。

首先介绍基于距离的断点。我们设置一个距离阈值，在轨迹段上，从样本点开始，每隔距离设置一个断点，使用断点将相邻样本点组成的一条轨迹段切分为条轨迹段，最终得到轨迹段上所有的断点集合。需要注意的是，这里的距离指的是标准空间下的距离，包括全部三个维度。如图所示，是查询轨迹R中的一条轨迹段，我们使用三个断点，和将轨迹段分成四段，其中前三段的长度、和的长度均为，而末尾的子段的长度小于。



在得到轨迹段中的断点之后，我们使用寻找对应样本点的算法，在轨迹段上找到与断点集合中所有断点的对应点。由于对应轨迹段存在三种空间情况，我们首先讨论第一种最基本的情况，即轨迹段的对应轨迹段是一条直线段。

在得到断点的对应点之后，就可以由轨迹段样本点和这些断点集合计算对应点的距离。如果按照图中的例子，我们可以得到距离、、、、，这是一条轨迹段中的点到对应轨迹段的空间距离。根据每一个端点和断点的位置为得到的距离赋予权值，最后相加，得到轨迹段之间的空间距离。即。即样本点的权值为，中间的断点的权值为，最后一个断点的权值为，样本点的权值为。更一般性的公式如公式所示。

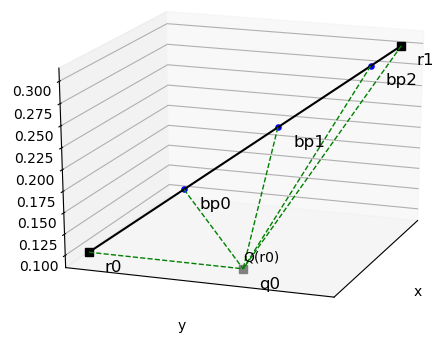
当n>=2时，

当n=1时，

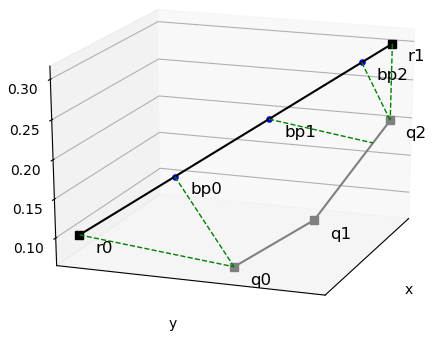
当n=0时，

其中，为断点的个数。

上面讨论了的对应轨迹段为直线段的情况。当出现第二种空间分布情况，即与为同一个点，如图二所示，依然可以使用上述方法，只不过这里每个端点和断点的对应点都是，省了一步求对应点的步骤。当出现对应轨迹段出现第三种空间分布情况，即与中间间隔着样本点，如图三所示，与第一种情况的区别就是每个断点需要在与中间间隔的折线段中求出对应点。当求出对应点之后，三种空间分布的对应轨迹段间距离的计算方法都相同，都是求对应样本点的距离乘上对应权值，最后再求和便是轨迹段到轨迹段的空间距离。



二



三

基于时间的断点和基于距离的断点的区别在于断点划分的维度是时间维度还是空间维度。在标准空间下，x轴和y轴代表的

对应轨迹段间的标准空间距离是轨迹距离的重要组成部分，对应轨迹段之间的距离越远，那么对应的轨迹段相似程度越小，越近相似程度越大。因此我们需要设计一个计算标准空间下对应轨迹段间距离的算法。

## // fixme:重写！因为此处空间距离太过简单。

我们记和到其对应点距离的平均值作为轨迹段之间的距离，计算方法如公式所示。

### 3.3.3对应轨迹段相似性及代价

对应轨迹段之间的相似性由轨迹段形状相似性和轨迹段的空间距离决定，对应轨迹段的相似性和轨迹段形状相似性呈正相关的关系，和轨迹段之间的空间距离呈负相关的关系，并且由于在标准空间下，的取值范围为。因此我们可以使用下面的公式描述对应轨迹段相似性。

### 【有问题】当形状距离为负数时，对于空间距离小的没有起到激励作用。

为了找出数据轨迹R中和查询轨迹Q最相似的部分轨迹段，就需要使用算法删除掉R中和Q中不相似的部分轨迹段，而轨迹段相似性只是作为两条轨迹段是否相似的一个衡量标准，虽然其中涉及了到对不相似的轨迹段的惩罚，比如改进余弦距离对形状不相似的惩罚，但是不能完全依据改进轨迹段相似性去对一条轨迹段做出过滤和保留操作。所以我们需要定义一个轨迹段相似性代价的概念，表示如果保留该轨迹段，我们在获得它的相似性的同时需要付出的代价。代价是我们在寻找最相似轨迹段的时候需要尽可能少地去付出的，同时，我们在找最相似轨迹段的时候，希望保留下来的数据轨迹段都是与查询轨迹最相似的部分，发现了代价和数据轨迹段长度的共性，我们可以使用数据轨迹段的长度做为代价，每当想保留下来一个相似性较大的数据轨迹段时，我们都需要用数据轨迹段的长度作为代价，可以用来评估保留下来轨迹段的相似程度值不值。保留轨迹段的代价公式如公式所示。

## // TODO: 此处应该在加一个计算两条轨迹段为输入，轨迹段相似程度为输出的算法。

## 3.4本章小结

本章首先提出了数值不均衡问题，为了解决该问题，引入了标准空间的概念，对时间和空间维度做了一个标准化处理，使得在高时效性的问题下，时间距离和空间距离可以进行统一考虑。

然后基于标准空间，定义了轨迹段的相似性和保留轨迹段的代价，其中轨迹段的相似性由轨迹段形状相似性和轨迹段标准空间距离构成。后文将基于轨迹段的相似性和代价，对数据轨迹段做过滤操作，过滤掉那些相似性太低，而代价又很大的轨迹段，从而找出最相似的局部轨迹。

# 第四章 基于标准空间的局部轨迹相似性算法

## 4.1轨迹对应点查找算法

## 4.2基于最相似子轨迹段的轨迹相似性计算函数

## 4.3三维网格索引加速对应点查找

4.3.1三维网格索引方法

4.3.2基于上下界的剪枝

# 第五章 实验设计与分析

# 第六章 总结与展望