**移动对象轨迹相似性查询综述**

丁光伟,杨晓春

（东北大学 计算机科学与工程学院，辽宁 沈阳 110819）

**摘 要：**作为数据查询的一个分支，移动对象的轨迹相似性查询在日常的生活和科学研究中起着十分重要的作用。而随着大数据时代的到来和移动设备的普及，海量的轨迹数据也随之产生，研究轨迹数据中隐藏的内在信息显得尤为重要。本文的研究重点是轨迹数据的表示及其相似性查询，首先介绍了轨迹的一些基本概念，之后罗列了相似性查询中常见的函数，最后针对几个特殊场景，介绍了具体情况下的轨迹相似性查询方法。

**关键字：**轨迹数据；轨迹相似性查询；度量函数

**中图分类号：**TN0 **文献标志码：**A **国家标准学科分类代码：**510.4030

Survey of Trajectory Similarity Query for Moving Objects

*DING Guang-wei, Yang Xiao-chun*

(School of Computer Science and Engineering, Northeastern University, Shenyang 110819, China. Corresponding author: Ding Guang-wei, E-mail: 837478687@qq.com)

**Abstract:** As a branch of data query, the trajectory similarity query of moving object plays a very important role in daily life and scientific research. Massive trajectory data are produced with the arrival of large data age and the popularity of mobile devices, and the study of trajectory data hidden in the internal information is particularly important. In this paper, we focus on the trajectory data and its similarity. Firstly, some basic concepts of the trajectory are introduced. Then the common similarity query functions are listed. Finally, the trajectory similarity query method is introduced for several special scenes.

**Key words：**trajectory data; trajectory similarity query; measure functions

# 1 引言

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，比如用户当前所在位置的经纬度、时间、速度和移动方向等，对路网中的移动车辆和行人的追踪越来普遍，随之产生的是海量的轨迹数据。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[1]、隐私保护[2]和行为预测[3]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度量技术的发展。

本文组织结构如下：第2章介绍常见的移动对象轨迹的表示方法；第3章对使用最广泛的移动对象轨迹的相似性度量方法进行一个概述和总结；第4章结合不同的应用场景，对一些具体情况下使用的相似性度量方法做一个简单的介绍和分析。第5章总结全文。

# 2 移动对象轨迹的表示方法

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。理论上，轨迹是一个连续的函数，函数的自变量是时间，因变量是移动对象在该时间的位置坐标[4]。但在实际情况下，轨迹只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>[5]。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[6]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。

## 2.2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[7]。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。

## 2.3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[8]。下面这个例子可以很好地说明基于文本信息的轨迹的应用。

假如一个旅行者到一个陌生的城市度假，计划游览三个地点q1、q2和q3，然后通过地图软件上查找可以通过这三个地点的路线。图1将展示这个场景。用户发出查询请求Q，包括上述三个地点，然后软件系统根据与三个地点的距离，找到了离用户要求的三个地方最近的一条路线T1。同时系统提供另一条路线T2给用户，与T1相比，这条路线虽然与用户指定的三个地点相对较远，但是也能满足旅行者在q1、q2和q3这三个地点对应关键字的活动，比如购物、看电影等偏好，可以作为用户的参考方案。

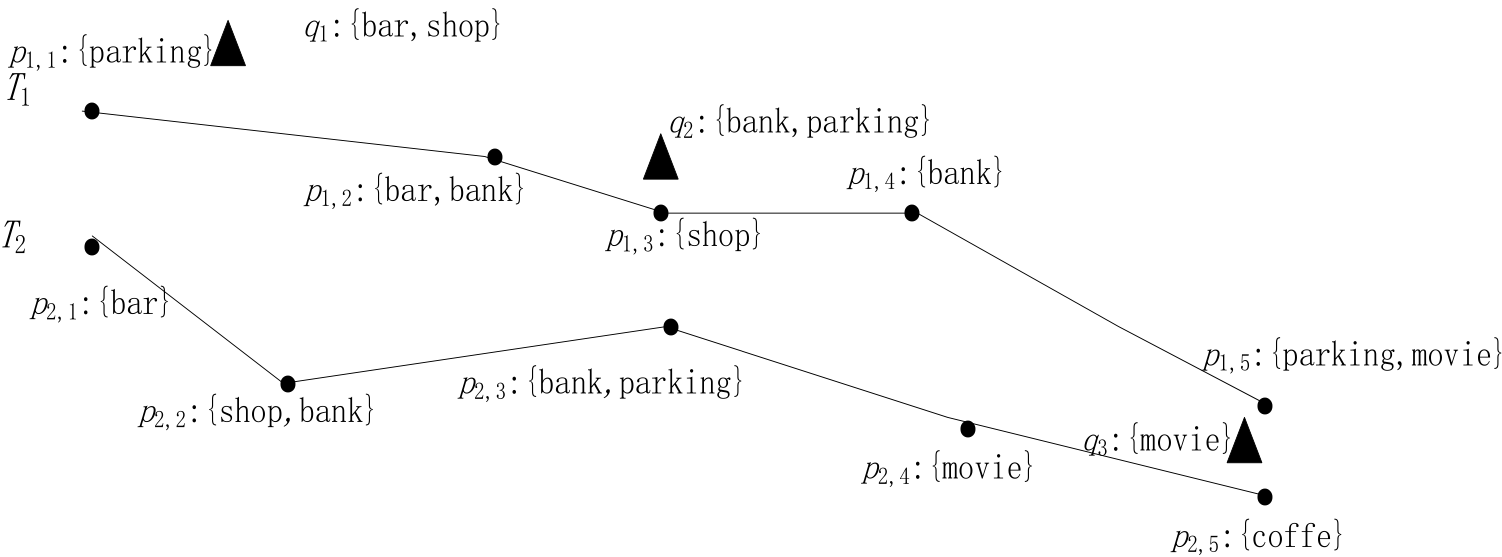


图1 基于文本的轨迹相似性查询

Fig.1 Text - based trajectory similarity query

也就是说，在实际情况中，历史轨迹数据并不完全通过用户指定的查询点，所以返回结果可能是几条最接近用户指定地点的轨迹，可以有空间上的接近，也可以有文本上的接近。返回的历史轨迹在空间上应当尽可能通过或者接近用户指定的位置，在文本上应当包括所有用户指定位置对应的文本关键字。

## 2.4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。

在图2中有TR(a)，TR(b)和TR(c)三条轨迹，可分别表示如下：

TR(a)={(c2,[0,3)),(c3,[3,6)),(c6,[6,8)),(c5,[8,9)),

(c8,[9,14)),(c9,[14,18))}；

TR(b)={(c2,[0,2)),(c3,[2,4)),(c6,[4,8)),(c9,[8,12)),

(c8,[12,18))}；

TR(c)={(c1,[0,2)),(c2,[2,6)),(c5,[6,8)),(c8,[8,11)),

(c5,[11,12)),(c8,[12,15)),(c9,[15,18))}。

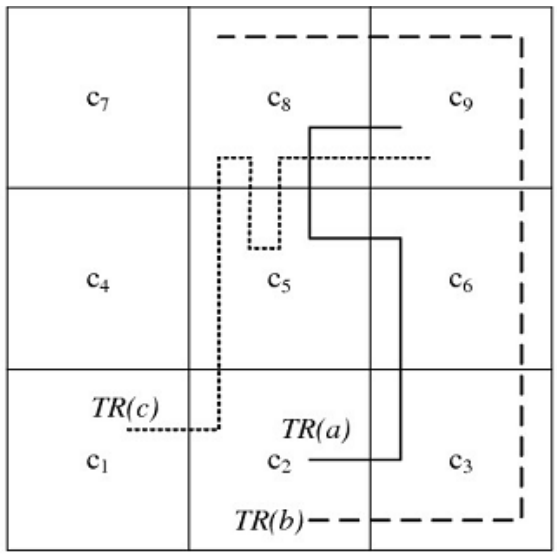


图2 网格表示法

Fig2. Grid notation

与前几种轨迹表示方法相比，网格表示法更关注的是移动对象进入和离开一个区域的时刻以及移动对象所处的大致区域，在计算轨迹相似性的时候，利用网格之间的位置关系可以有效地减少运算复杂度。与网格表示法类似的还有路段表示法。路段表示法的表示形式为T={(S1, L1),(S2, L2),…(Sn, Ln)}，其中Si代表路网中的路段，Li代表移动对象在该路段移动的长度。

## 2.5本章小结

本章介绍了移动对象轨迹的一些常见的表示方法，我们在不同的应用场景下会采取不同表示方法来表示移动对象产生的轨迹，用适合的表示方法来解决不同的问题，有助于简化问题，从而帮助我们更好地解决问题。

# 3 移动对象的轨迹相似性度量函数

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，其中就包括轨迹的相似性查询[9]，其定义如下。

**定义3.1**（轨迹相似性查询）给定一个轨迹T，轨迹数据库S和一个阈值，返回S的一个子集，使得

d(T,),

其中d(·)是轨迹数据之间的相似性度量函数。

在移动对象的轨迹相似性查询中，相似性度量函数是核心。我们给出相似性度量函数的形式化定义。

**定义3.2**（距离度量函数）给定一个定义在轨迹数据上的空间T，以及任意两条T中的轨迹x与y，则T上的距离度量函数d定义为：

其中R是实数空间。

选择一个合适的相似性度量函数和利用度量函数制定一个恰当的度量策略至关重要，这些因素同时决定了最终得出的查询结果的有效性。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度量函数进行相似性度量即可，这样就可以在一定程度上减少运算时间。因此，在面对不同场景时我们需要根据具体情况采用不同的度量方法。下面将介绍几种常用的相似性度量函数，并给出它们在二维空间下的表达形式。

## 3.1欧氏距离

轨迹的欧氏距离[10]是待测量的两条轨迹上所有对应点的欧氏距离进行综合处理得到的结果，我们在这里采取求和的方式，当然也可以求均值、取中值等。在二维空间上的表达式如式(1)和(2)所示：

(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，早期研究中使用十分广泛。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的时间偏移，局部时间偏移是指两条轨迹的对应点在时间上出现了偏移，需要将轨迹移动一段距离才能匹配上。第三，使用欧氏距离进行相似性度量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，我们现在一般用此度量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。

## 3.2动态时间弯曲距离

由于样本点采集设备的误差等原因，两条轨迹数据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度量。动态时间弯曲距离就是为了解决此类问题而提出的，它在二维空间上的公式如公式(3)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。

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

(3)

(4)

(5)

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。

## 3.3最长公共子序列距离

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(4)所示。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度量函数而言，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。

## 3.4编辑距离

编辑距离原本是用于求字符串相似度的一个函数，通过对一个字符串的增加、删除和修改操作，使之变成另外一个字符串所需要的最小操作次数。后来发现这个函数的思想也可以用于比较两条轨迹的相似程度。与动态时间弯曲距离和最长公共子序列距离一样，编辑距离也是通过递归的方式进行计算的，公式如公式(5)所示。如果两条轨迹起始点的距离小于阈值，那么判定两个点匹配，轨迹的编辑距离就是除初始节点以外的剩余轨迹的编辑距离，否则递归子轨迹，寻找最小编辑距离，知道某条子轨迹长度为0。和最长公共子序列距离相比，编辑距离可以区分具有相同子序列但是对应点距离不同的轨迹，提高了轨迹相似性度量的有效性和精度。同时编辑距离可以避免噪声干扰，但是时间复杂度仍然为。

# 4 移动对象轨迹的相似性查询

在第3章我们给出了移动对象轨迹相似性查询的定义，即给定一条待查询的轨迹和一个轨迹数据库，我们需要找到轨迹数据库的子集，使该子集中的所有轨迹在一定程度上与待查询的轨迹是相似的。但是由于应用场景和用户需求的不同，我们一般不会直接使用相似性计算公式，而是根据具体情况来指定合适的相似性查询方法。本章将介绍几个具体的应用场景下的轨迹相似性查询。

## 4.1轨迹校准

在路网环境下，尤其是城市路网中，车辆和人产生的轨迹将遵循一定的规则，比如车辆一般不会产生穿过建筑物的原始路径，人一般不会产生穿过一堵墙的原始路径。但是由于位置采集设备的参数和误差等原因，我们得到的采样轨迹和移动对象产生的原始路径有很大差异，这将导致我们后续的轨迹相似性查询结果不够准确。此外，我们在进行前期的轨迹采样时，会制定采样策略和设定采样参数，进行相似性查询的两条采样得到的轨迹可能具有不同的采样策略和不同的采样率，我们称之为轨迹异质现象。轨迹异质也会导致我们的相似性查询结果没有意义。轨迹校准技术[11]的产生就是为了解决采样误差和轨迹异质这两个问题。

轨迹校准依赖于由锚点组成的参考系统，锚点是指稳定且独立于轨迹数据而存在一个空间位置。锚点可以指物理存在的地理对象，比如兴趣点POI，或者可以被虚拟地定义为网格的质心。实际上，只要稳定且不受输入的轨迹数据影响，任何物理位置都可以当作锚点。而参考系统就是指包含锚点集的一个区域。比如包含以兴趣点和道路交叉点作为锚点的城市参考系统。

轨迹校准就是将轨迹中的样本点全部对应为参考系统中的锚点。并且在理想情况下，无论采用什么采样策略，只要有相同的原始路径，就会产生相同的校准后的轨迹。轨迹校准分为两步，第一是对齐，第二步是补充。对齐操作指的是找到距离采样点最近的锚点，如果二者距离大于阈值，则舍弃该采样点，否则将该采样点对齐到该锚点上，即在数据库中舍弃采样点数据，存储锚点数据。补充操作指的是对齐后的轨迹中，若在两个相邻锚点连线之间有距离小于阈值的锚点，则将锚点补充进轨迹中，如图3所示。

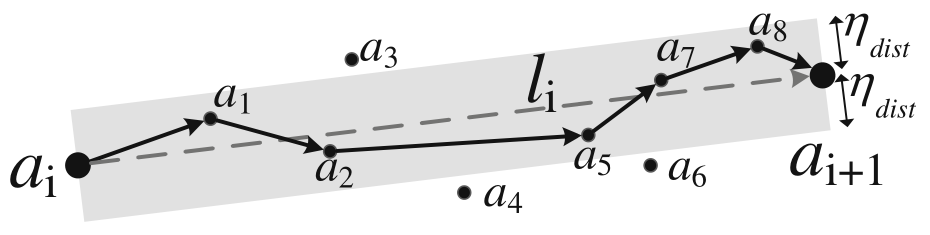


图3 补充操作的例子

Fig3. The example of complement

轨迹校准的效果如图4所示，轨迹a和轨迹b是两条高度相似的路径，但是采样后得到的轨迹相似度很差。经过轨迹校准后再进行相似度测量，此时得到的结果便近似于原始路径的相似度。所以为了减少采样误差和两条轨迹的采样策略不同给轨迹相似性度量带来的影响，我们可以采取轨迹校准技术来对轨迹进行预处理，然后再采取相似性度量函数来对轨迹进行相似性度量。轨迹校准技术可以应用于开发新的索引技术和编写更有效的查询处理算法，给未来的研究开辟了一个新的方向，

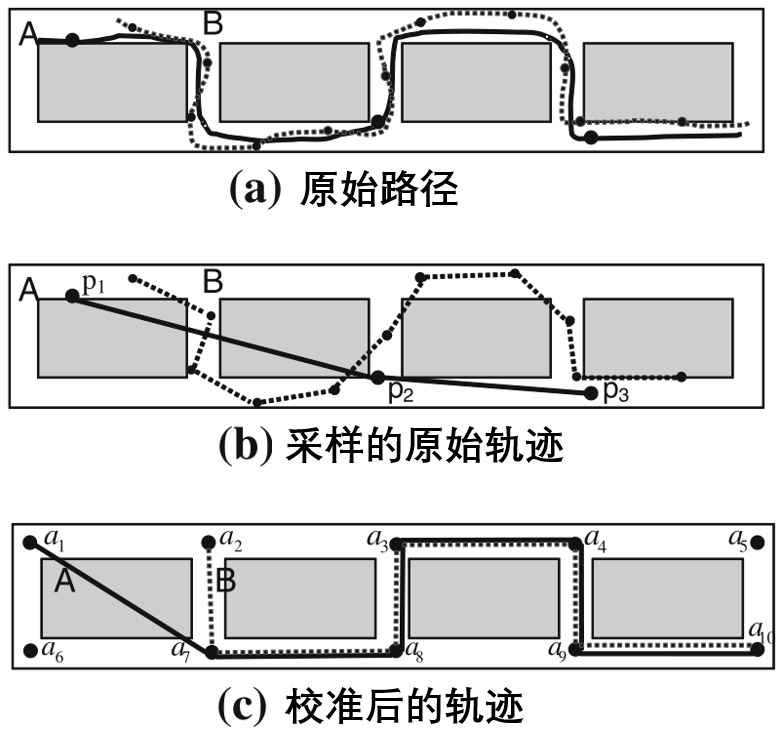


图4 轨迹校准的效果

Fig4. The effects of trajectory calibration

## 4.2加密轨迹的相似度计算

随着大数据时代的到来，我们可以采取将数据外包到云的方法来准确、高效地对海量数据进行存储、管理和查询处理。但是数据大多会涉及用户隐私，尤其是轨迹数据，其中包含大量可以推测出用户私密信息的时空关联，比如用户的家庭住址，工作单位和生活水平等。所以我们在对轨迹数据外包之前，需要进行加密处理。但是问题也随之而来，由于云中的轨迹数据库是加密的，所以不能直接和待测量的轨迹数据进行相似度测量。我们需要针对加密轨迹数据设计一个安全高效的相似度计算方法[11]。

首先我们需要了解一下加密轨迹数据库的查询过程，如图5所示。首先是数据拥有者将加密轨迹数据E(T)外包到云，然后用户上传一个加密轨迹数据E(Y)，然后直接在云中利用加密轨迹相似度计算方法进行相似度计算，最后将与轨迹Y相似的轨迹以加密形式返回给用户。在这里我们假定云属于半诚实的对手模型，就是云即使想盗取加密数据中的信息，但也会遵循我们制定的协议。

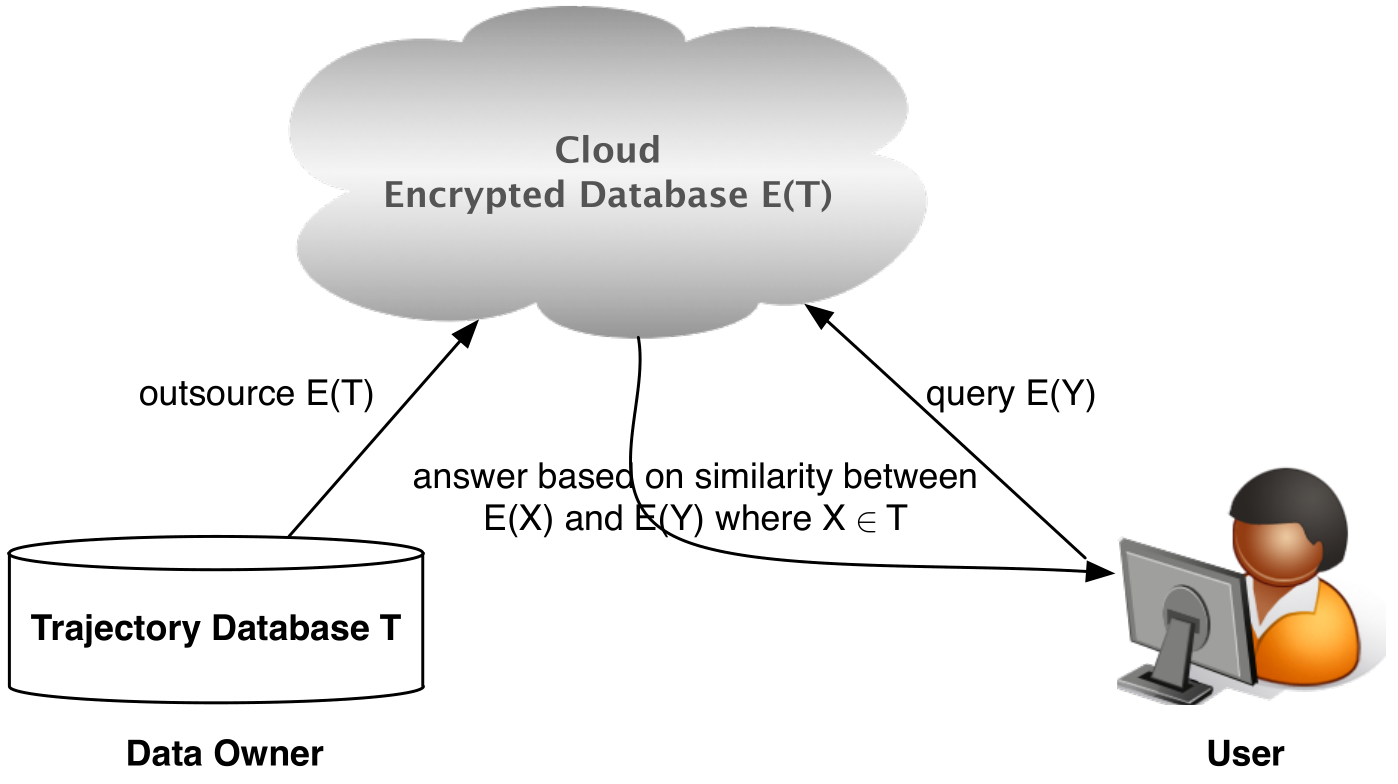


图5 加密轨迹数据库上的相似度查询

Fig5. The similarity query on the encrypted trajectory database

在加密部分，我们选取的是paillier加密系统来加密轨迹数据。加密函数E()的定义如公式6所示：

(6)

其中m是加密的消息，N是两个大素数p和q的乘积，g产生一个N阶子集，r是一个随机数。在paillier加密系统中包含两个重要的属性，一个是同态加属性，还有一个是语义安全性。同态加属性指的是两个密文的乘积将被解密为他们对应的明文的和，密文的k次幂将被解密为k及其相应明文的乘积。而语义安全性是指给定一组密文，攻击者无法导出敏文的任何信息。利用同态加属性，我们可以通过公式(7)来计算两个点的平方欧氏距离，从而计算出两条轨迹的相似性。最终将加密状态下的轨迹集返回给用户。

(7)

这种方法可以使用常见的度量函数来计算轨迹相似性，比如动态时间弯曲距离、最长公共子序列距离和编辑距离，并且可以确保了轨迹数据不仅在相似度计算期间保持私有，而且最终结果的返回不会暴露个人隐私。但是不足就是我们假定了对手是半诚实的，尚未解决恶意对手情况下的加密轨迹的相似度计算。

# 第5章 总结

本文中，我们介绍了移动对象轨迹相似性度量相关的概念、成熟的技术和一些新颖的研究。在文章开头我们说明轨迹数据的含义和一些常见的移动对象轨迹的表示方法。接着我们详细介绍了几个应用十分广泛的相似性度量函数和它们各自的优缺点以及适用范围。最后，我们假设了几个具体的应用场景并给出了轨迹相似性度量在这些场景下巧妙地运用。

随着大数据时代的到来，海量数据给相关方面的应用带来了前所未有的机遇，但同时在数据管理、计算和挖掘方面的难度也大大提升了。而轨迹数据的应用前景非常广泛，对轨迹数据相似性的研究在数据查询方面占有重要地位，扩展轨迹数据的应用场景意义十分重大。

**References：**

1. Chih-Chieh Hung, Wen-Chih Peng, Wang-Chien Lee. Clustering and aggregating clues of trajectories for mining trajectory patterns and routes [J]. VLDB, 2015, (24): 169-192.
2. Sheng Gao, Jianfeng Ma1, Weisong Shi, Guoxing Zhan. LTPPM: a location and trajectory privacy protection mechanism in participatory sensing [J]. Wireless communication and mobile computing, 2015, (15):155-169.
3. Zelei Liu, Liang Hu, Chunyi Wu, Yan Ding, Jia Zhao. A novel trajectory similarity–based approach for location prediction [J]. International Journal of Distributed Sensor Networks, 2016, (11):113-126.
4. 曹静. 基于GPS数据的用户轨迹相似性分析[D].北京:北京理工大学，2015:1-67.
5. 沙文强. 基于路网的移动对象轨迹相似性查询方法研究[D].天津:天津理工大学，2015:1-48.
6. Jin C, Qian W, Zhou A, Analysis and management of streaming data: a survey[J]. Journal of Software, 2014, 15(8): 1172-1181.
7. Florian Damerow, Stefan Klingelschmitt and Julian Eggert. Spatio-Temporal Trajectory Similarity and its Application to Predicting Lack of Interaction in Traffic Situations[C]. International Conference on Intelligent Transportation Systems, Windsor Oceanico Hotel, Rio de Janeiro, Brazil, November 1-4, 2016.
8. Bolong Zheng, Nicholas Jing Yuan, Kai Zheng, Xing Xie, Shazia Sadiq and Xiaofang Zhou. Approximate Keyword Search in Semantic Trajectory Database[C]. International Conferenc on Data Engineering, Seoul, South Korea, 2015.
9. 龚旭东. 轨迹数据相似性查询及其应用研究[D].合肥:中国科技大学，2015:1-114.
10. H. Wang, H. Su, K. Zheng, S. Sadiq, and X. Zhou. An effectiveness study on trajectory similarity measures[J]. In Proc. 24th Australasian Database Conference, volume 137, pages 13–22, 2013.
11. Han Su, Kai Zheng and Jiamin Huang. Calibrating trajectory data for spatio-temporal similarity analysis[J].VLDB, 2015,(24):93-116.
12. An Liu and Kai Zheng. Efficient Secure Similarity Computation on Encrypted Trajectory Data[J]. IEEE, 2015,(22):6-15.