一种基于马尔可夫模型的稀疏轨迹终点预测算法

徐广根[1] 杨 璐[1,2] 严建峰[1,2]
(苏州大学计算机科学与技术学院 苏州 215006)[1] (香港城市大学创意媒体学院 香港 999077)[2]

摘 要 随着移动设备的普及与定位技术的成熟,涌现出了各种基于地理位置的应用软件不断涌现。为了使这类应用软件给用户提供精准的基于地理位置的服务,实时、准确、可靠地预测移动对象的不确定性轨迹显得尤为重要。目前大多数传统的轨迹终点预测方法都是通过计算轨迹之间的相似度来预测给定轨迹的终点,这种算法的弊端是没有充分考虑轨迹数据时间序列之间的前后联系,导致预测结果偏差较大。理论证明,马尔可夫模型对处理时间序列数据具有较好的效果。因此,针对轨迹终点预测的问题,提出了一种基于马尔可夫模型的预测算法。同时,针对样本运动空间提出一种新的划分网格策略——K-d tree 网格划分。实验结果表明,相比于传统方法,运用马尔可夫模型预测轨迹终点的算法的精度有明显提高,预测时间会大大缩短。

关键词 轨迹挖掘,终点预测,马尔可夫模型
中图法分类号 TP391 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2017.08.034

Sparse Trajectory Destination Prediction Algorithm Based on Markov Model

XU Guang-gen[1] YANG Lu[1,2] YAN Jian-feng[1,2]
(School of Computer Science and Technology, Soochow University, Suzhou 215006, China)[1]
(School of Creative Media, City University of Hong Kong, Hong Kong 999077, China)[2]

Abstract With the popularity of mobile devices and the maturity of location technologies, a variety of location-based applications emerge. In order to make such applications provide users with accurate location-based services, timely, accurately, reliably forecast uncertainty track of moving objects is particularly important. Currently, most traditional methods predict the destination of a given trajectory by calculating the similarity between two trajectories, and this algorithm does not fully consider the drawbacks of backward and forward linkages between trajectory time series, resulting in a larger prediction error. Theory demonstrates that the Markov model to deal with time series data have very good effect. Therefore, we proposed a sparse trajectory destination prediction algorithm based on Markov model. Meanwhile we investigated a new partitioning strategies for the sample motion space——grid partitioning based on K-d tree. The experimental results show that compared with traditional methods, using Markov model to predict the destination of the trajectory will significantly improve the accuracy of the algorithm, and the predicted time will be shortened.

Keywords Trajectory mining, Destination prediction, Markov model

1 引言

随着各种智能手机以及车载导航系统在人们日常生活中的逐渐普及,获取移动对象时空位置的手段变得越来越丰富。目前在各个现实应用领域中,如智能交通系统、智能导航、电子商务等,用户都需要实时获取和分析移动对象的位置信息。同时,也涌现出了很多基于地理位置预测轨迹终点的手机应用软件,这些手机应用软件的功能主要包括:基于地理位置给用户推荐附近的风景区;基于历史轨迹预测得到用户终点,从而有针对性地给用户推送广告。因此,准确、实时地预测用户行驶轨迹的终点位置,为用户提供更加人性化的服务,具有极高的应用价值。移动对象轨迹位置预测引起了学术界的广泛关注。现在很多研究都已经成功证实人类的移动轨迹具有极高的预测性[1-2]。绝大多数的轨迹预测方法使用的轨迹数据来源于轨迹共享网站的历史时空轨迹[3]和大量城市出租车轨迹集合。针对如何精准、实时地预测轨迹终点位置这一问题,目前已有一些研究。一方面主要从挖掘轨迹频繁模式[4]出发,例如 Morzy 等人[5]提出一种结合前缀树 PrefixSpan 和频繁挖掘模式 FP-tree 算法来挖掘移动对象的动态规则,这一方案的劣势在于构建前缀树和 FP-tree 的时间复杂度较高;

到稿日期:2016-06-23 返修日期:2016-09-03 本文受国家自然科学基金(61373092,61033013,61272449,61202029),江苏省教育厅重大项目(12KJA520004),江苏省科技支撑计划重点项目(BE2014005)资助。
徐广根(1992—),男,硕士生,主要研究方向为移动对象数据挖掘、机器学习和深度学习;杨 璐(1982—),女,博士,副教授,CCF 会员,主要研究方向为软件可靠性和机器学习;严建峰(1978—),男,博士,副教授,CCF 会员,主要研究方向为并行计算和机器学习,E-mail:yanjf@suda.edu.cn(通信作者)。

Jeung 等人[6]提出了一种综合考虑移动对象运动模式和运动函数的混合预测方法，不足之处是提供的查询预测仅仅支持较短时间内的查询结果。上述两种轨迹预测方法都是基于地理位置的特性，而 Ying 等人[7]利用用户历史轨迹的语义特征和空间位置预测移动对象的下一个落点位置信息，但该方法在计算每条候选路径的 Semantic Score 时，代价相对较高。郑宇等人[8]充分利用用户的行为习惯，包括旅行经历和兴趣爱好，提出了一种 Hypertext Induced Topic Search 模型来推测用户感兴趣的运动路线，但是该方法通常是用于给用户推荐潜在感兴趣的地点，存在一定的局限性。上述所有方案都为本文的研究提供了指导，但是都忽略了历史轨迹数据稀疏的问题：给定的历史轨迹空间中没有能够与要预测轨迹相匹配的轨迹。在本文之前，微软亚洲研究院郑宇提出了一种基于 SubSyn(Sub-Trajectory Synthesis) 算法[9-10]预测轨迹终点的方法，其主要思想是首先将历史轨迹数据序列化，再将经过序列化处理的轨迹分割为只包含相邻两个点的子轨迹，并最终把这些子轨迹拼接成一条完整的轨迹。但是这种采用均匀划分城市地图的方案没有考虑到拼接形成的轨迹与真实轨迹存在极大的偏差，导致轨迹数据的真实性下降，最终影响预测结果。针对上述方案的不足，本文在郑宇所提方法的基础上，充分考虑轨迹稀疏问题，将城市地图基于 K-d tree 划分网格，然后运用马尔可夫模型[11]预测稀疏轨迹终点。

本文第 2 节介绍马尔可夫模型的相关概念和定义；第 3 节介绍轨迹终点预测的过程；第 4 节给出实验结果和分析；最后总结全文。

2 马尔可夫模型

2.1 马尔可夫链

定义 1 马尔可夫链[12-14](Markov Chain)是状态空间中从一个状态转移到另一个状态的随机过程。马尔可夫链满足马尔可夫性质的随机变量 $X_1, X_2, X_3, \cdots, X_n$，即给出当前状态，将来状态和过去状态是相互独立的。从形式上看，如果两边都有条件分布的定义(即如果 $\Pr(X_1=x_1, \cdots, X_n=x_n)>0$)，则：

$$\Pr(X_{n+1}=x \mid X_1=x_1, X_2=x_2, \cdots, X_n=x_n)=\Pr(X_{n+1}=x \mid X_n=x_n) \quad (1)$$

其中，x_i 的可能值构成的可数集 S 叫做该链的“状态空间”。

式(1)中假设 X_n 表示当前状态，$X_1, X_2, X_3, \cdots, X_{n-1}$ 分别表示过去 $n-1$ 种状态，X_{n+1} 表示将来状态，那么式(1)的含义表明在将来时刻 $n+1$ 处于状态 x 仅仅与当前时刻 n 所处状态 x_n 有关，与过去的 $n-1$ 个时刻的状态无关。

2.2 k 步转移概率

用 k 步将从状态 i 到状态 j 的概率表示为 $p_{ij}^{(k)}=\Pr(X_k=j \mid X_0=i)$，而单步转移概率是 $p_{ij}^{(1)}=\Pr(X_1=j \mid X_0=i)$。转移概率表示已知 n 时刻处于状态 i，经过 k 个单位时间后处于状态 j 的概率，记作：$P_{ij}(n, n+k)$。k 步转移概率满足 Chapman-Kolmogorov[13]等式：对任意 $k \in \{1, 2, 3, \cdots, n-1\}$，$p_{ij}^{(n)}=\sum_{r \in S} p_{ir}^{(k)} p_{rj}^{(n-k)}$，$p_{ij}^{(n)}=p_{ij} * p_{ij}^{(n-1)}$。

3 轨迹终点预测过程

在对移动对象运动空间进行划分的过程中，通常最简单且最直接的方法是采用固定尺度的均匀网格去划分覆盖样本的空间。在这里，所有移动对象覆盖的运动空间大小是固定的，设置不同大小的网格粒度，最终由子轨迹拼接形成的序列化轨迹与真实轨迹之间的差异也是互不相同的。类似于用户在浏览器中查看地图时，用户选择不同的比例尺，网页地图所呈现出的效果也是不一样的：比例尺设置得越小，所看到的元素数量越少，精度越高；比例尺设置得越大，所看到的元素数量越多，精度越低。与此类似，将整个样本运动空间划分得越细，得到的序列轨迹与真实轨迹之间的差异就越小。在使用均匀网格划分样本运动空间的过程中，存在如下缺点：极易将原本联系紧密的数据点划分到不同的网格中，从而丢失轨迹数据点本身分布的信息，最终降低终点预测的精度。

3.1 K-d tree 网格划分

为了解决上述均匀网格划分样本运动空间导致轨迹数据分布信息丢失的问题，本文提出一种基于 K-d tree 划分样本运动空间的算法。K-d tree 是每个节点都为 k 维点的二叉树。所有非叶子节点都可以被看作用一个超平面把空间区分成两个半空间(Half-space)，节点左边的子树代表超平面左边的点，节点右边的子树代表超平面右边的点。选择超平面的方法如下：每个节点都与 k 维中垂直于超平面的那一维有关。因此，若选择按照 x 轴划分，所有 x 值小于指定值的节点都会出现在左子树，所有 x 值大于指定值的节点都会出现在右子树。超平面可以用该 x 值确定，其法向量为 x 轴的单位向量。与均匀网格划分不同的是，K-d tree 网格划分使得轨迹数据点密集的区域会被更多的网格划分，以尽可能保留轨迹数据点的分布信息，从而提升预测给定轨迹终点的精度。图 1 和图 2 分别给出了均匀网格划分和 K-d tree 网格划分的示意图。

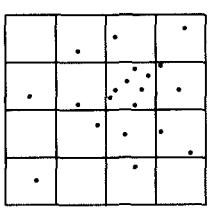

图 1 均匀网格划分

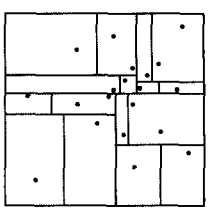

图 2 K-d tree 网格划分

K-d tree 网格划分样本运动空间递归算法如算法 1 所示。

算法 1 K-d tree 网格划分样本运动空间递归算法

输入：历史轨迹数据点集合 D，每个网格轨迹数据点个数阈值 n

输出：K-d tree 划分方法网格 G

1. 对于历史轨迹数据点集合 D 中所有轨迹点，只保留经度 Lon 和纬度 Lat 数值小数点前 3 位，去重后得到新的数据点集合 D′。
2. 对 D′中的所有数据点，比较 Lon 和 Lat 不同方向上的方差，在方差较大的维度上选取一条中线，中线选取的规则是：若轨迹数据点的数量为偶数，则中线使得两侧的数据点的数量相等；若为奇数，中线使得两侧的数据点的数量相差一个即可。
3. 计算中线两侧轨迹数据点的数量，若网格 g 中轨迹数据点的数量小于阈值 n，则将该网格 g 加入到 G，并停止对网格 g 的划分；反之，若上一次在 Lon 方向划分，则对网格 g 在 Lat 方向上继续划分，否则对网格 g 在 Lon 上继续划分，直到所有网格中的轨迹数据点的数量小于阈值 n 时停止划分。

算法 1 中第 1 步保留原始轨迹点数值的小数位点前 3 位

的原因是：*Lon* 和 *Lat* 数值小数点第三位在实际中是百米数量级，选取数值小数点前 3 位可以避免原始轨迹数据集合 D 中数据量庞大而导致最后通过基于 K-d tree 生成的网格集合 G 过于密集，同时也可以缩短该网格划分算法的运行时间。阈值 n 是当前网格是否继续划分的条件，若当前网格 g 中包含的数据点数量大于 n，则该网格 g 继续被划分，否则表示该网格 g 是稀疏的，不需要继续划分。阈值 n 体现的实际效果可以理解成：在城市交通中心地带，会用较多的网格划分该区域；在城市偏远的“空闲”地带，只需要用相对较少的网格进行划分。

本文从数学理论角度出发，证明 K-d tree 网格划分方法确实优于最简单的均匀网格划分方法。首先，设定 i 表示划分完成之后的第 i 个网格，p_i 表示数据点落在第 i 个网格的概率。然后，定义“网格熵”，记作 H，具体的计算公式如下：

$$H=-\sum_{i=1}^{m}p_i\ln p_i=\sum_{i=1}^{m}p_i\ln\frac{1}{p_i} \tag{2}$$

其中，m 表示最终划分的网格个数。通过计算得到“网格熵”的最大值 $H_{\max}$ 和最小值 $H_{\min}$，比较两种不同网格划分方法。

$H_{\max}$ 的计算过程：根据数学理论，$f(x)=\ln x$ 是一个凸函数，由 Jensen 不等式[15]可得：

$$H=\sum_{i=1}^{m}p_i\ln\frac{1}{p_i}\leqslant\ln\sum_{i=1}^{m}p_i\ln\frac{1}{p_i}=\ln m \tag{3}$$

因此，$H_{\max}=\ln m$，等式成立的条件是：$\forall i\in[1,m]$，$p_i=\tilde{p}$，其中 $\tilde{p}=\frac{1}{m}\sum_{i=1}^{m}p_i=\frac{1}{m}$ 是所有 p_i 的均值。

$H_{\min}$ 计算过程：使用一阶泰勒公式展开函数 $f(x)=\ln x$，如下式：

$$\begin{aligned}f(x)&=f(x_0)+f'(x_0)(x-x_0)+\frac{1}{2!}f''(\xi)(x-x_0)^2\\&=\ln x_0+\frac{1}{x_0}(x-x_0)+\frac{1}{2!}\left(-\frac{1}{\xi^2}\right)(x-x_0)^2\end{aligned} \tag{4}$$

其中，$\frac{1}{2!}\left(-\frac{1}{\xi^2}\right)(x-x_0)^2$ 表示拉格朗日余项，ξ 表示在 x 和 x_0 之间的一个实数。由于拉格朗日余项小于 0，因此 $f(x)\leqslant\ln x_0+\frac{1}{x_0}(x-x_0)$。将 x 替换为 p_i，x_0 替换为 $\tilde{p}$，得到：

$$\begin{aligned}\sum_{i=1}^{m}p_i\ln p_i&\leqslant\sum_{i=1}^{m}p_i\left(\ln\tilde{p}+\frac{1}{\tilde{p}}(p_i-\tilde{p})\right)\\&=\sum_{i=1}^{m}-p_i\ln m+mp_i^2-p_i\\&=-\ln m\sum_{i=0}^{m}p_i+m\sum_{i=1}^{m}p_i^2-\sum_{i=1}^{m}p_i\\&=-\ln m+m\sum_{i=1}^{m}p_i^2-1\end{aligned} \tag{5}$$

通过方差计算公式得到数据点落在网格中概率分布的方差为：

$$\sigma^2=\frac{1}{m}\sum_{i=1}^{m}(p_i-\tilde{p})^2 \tag{6}$$

方差反映了轨迹数据点在所有网格中的分布情况。与均匀网格划分相比，K-d tree 网格划分方案所对应的方差更小，因为使用 K-d tree 网格划分得到的每个网格中所拥有的数据点个数相近，从而每个网格的概率值也是相近的。

扩展式(6)得到 $H_{\min}$，具体过程如下：

$$\begin{aligned}\sigma^2&=\frac{1}{m}\sum_{i=1}^{m}(p_i-\tilde{p})^2\\&=\frac{1}{m}\sum_{i=1}^{m}(p_i^2-2p_i\tilde{p}+\tilde{p}^2)\\&=\frac{1}{m}\sum_{i=1}^{m}p_i^2-\frac{1}{m}\sum_{i=1}^{m}2p_i\tilde{p}+\frac{1}{m}\sum_{i=1}^{m}\tilde{p}^2\\&=\frac{1}{m}\sum_{i=1}^{m}p_i^2-\frac{2}{m^2}+\frac{1}{m^2}\\&=\frac{1}{m}\sum_{i=1}^{m}p_i^2-\frac{1}{m^2}\end{aligned} \tag{7}$$

由此得到：$\sum_{i=1}^{m}p_i^2=m\sigma^2+\frac{1}{m}$，代入式(5)得到：

$$\begin{aligned}\sum_{i=1}^{m}p_i\ln p_i&\leqslant-\ln m+m\left(m\sigma^2+\frac{1}{m}\right)-1\\&=-\ln m+m^2\sigma^2\end{aligned} \tag{8}$$

最终得到：$H_{\min}=-\ln m+m^2\sigma^2$。

通过计算得到 $H_{\max}$ 和 $H_{\min}$，可以发现当 $\sigma^2\to0$ 时，H 会逼近 $H_{\max}$。K-d tree 网格划分方法会使得每个网格得到数据点的概率相近，σ^2 越小，“网格熵”越大，从而丢失的数据点分布信息越少。

3.2　轨迹序列生成方法

大多数通过 GPS 移动定位设备采集到的原始轨迹序列 $Trj=\{(lon_1,lat_1),(lon_2,lat_2),\cdots,(lon_n,lat_n)\}$，将 Trj 中所有 GPS 数据点全部映射到对应的网格，生成轨迹序列 $Trj=\{g_1,g_2,g_3,\cdots,g_n\}$，其中 g_i 表示在第 i 个时刻轨迹所在的网格。对于任意相邻的 g_i 和 g_{i+1}，若 $g_i=g_{i+1}$，则合并 g_i 和 g_{i+1} 为同一个网格；以此类推，合并轨迹序列中所有相邻且相同的网格，从而得到最终的轨迹序列为 $Trj=\{g_1,g_2,g_3,\cdots,g_m\}$，其中任意相邻两个网格 $g_i\neq g_{i+1}$。

3.3　基于马尔可夫模型的稀疏轨迹终点预测方法

基于马尔可夫模型的预测主要是利用转移状态矩阵来评估状态直接转移的可能性。3.1 节和 3.2 节通过将原始轨迹映射到划分好的网格中来得到轨迹序列。针对数据点稀疏问题，郑宇提出子轨迹综合轨迹预测方法，大致思路为：将所有轨迹全部划分为只包含相邻两个节点的子轨迹；然后连接子轨迹合成“综合”轨迹，该过程可以适当地扩展历史轨迹以包含更多的轨迹；最后使用马尔可夫模型离线训练转移概率，并计算给定轨迹终点的后验概率。

首先本文基于轨迹终点预测的问题构建马尔可夫模型，将所有轨迹序列中的每一个网格节点 g 映射到模型中的状态。模型中任意两个相邻状态之间的转移对应着轨迹序列中相邻网格节点 g_i 和 g_j 之间的移动。轨迹序列中相邻两个网格节点 g_i 和 g_j 表示从 g_i 移动到 g_j，转移概率用 p_{ij} 表示，如图 3 所示。

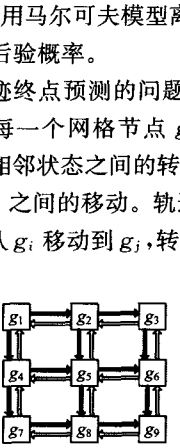

图 3　3×3 马尔可夫模型的状态转移图

p_{ij} 是一个条件概率，计算方式为包含序列网格节点对 $\{g_i,g_j\}$ 的轨迹数量除以包含网格节点 g_i 的轨迹数量：

$$p_{ij}=P(g_j\mid g_i)=\frac{|T_{i,j}|}{|T_i|} \tag{9}$$

设定划分完成后网格的数量为 N。对于所有轨迹序列

中的相邻网格节点，采用式(9)离线计算转移概率，并将这些概率值保存在一个 $N\times N$ 的二维矩阵 M 中(见图 4)。

$$M=\begin{pmatrix} 0 & p_{12} & 0 & p_{14} & 0 & 0 & 0 & 0 & 0 \\ p_{21} & 0 & p_{23} & 0 & p_{25} & 0 & 0 & 0 & 0 \\ 0 & p_{32} & 0 & 0 & 0 & p_{36} & 0 & 0 & 0 \\ p_{41} & 0 & 0 & 0 & p_{45} & 0 & p_{47} & 0 & 0 \\ 0 & p_{52} & 0 & p_{54} & 0 & p_{56} & 0 & p_{58} & 0 \\ 0 & 0 & p_{63} & 0 & p_{65} & 0 & 0 & 0 & p_{69} \\ 0 & 0 & 0 & p_{74} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & p_{85} & 0 & p_{87} & 0 & p_{89} \\ 0 & 0 & 0 & 0 & 0 & p_{96} & 0 & 0 & 0 \end{pmatrix}$$

图 4 一阶转移概率矩阵 M

利用 M 和 Chapman-Kolmogorov 等式可以得到 $k(2\leqslant k\leqslant len_{\max})$阶转移矩阵 M^k，其中 $len_{\max}$表示网格图中两个网格节点的最长路径，即网格中任意两节点经过 k 步的转移概率。

根据上述方式求解 k 阶概率转移矩阵，得到从一个网格节点 g_i 到另一个网格节点 g_k 的所有可能路径(包括绕道行驶路径 $L_{de,i\to k}$)的转移概率之和，用 $p_{i\to k}$表示。$p_{i\to k}$的计算如下：

$$\begin{aligned} p_{i\to k}&=\sum_{r=L_{i\to k}}^{L_{i\to k}+L_{de,i\to k}} M_{ik}^{r} \\ &=M_{ik}^{L_{i\to k}}+M_{ik}^{L_{i\to k}+1}+\cdots+M_{ik}^{L_{i\to k}+L_{de,i\to k}} \end{aligned} \tag{10}$$

其中，$L_{i\to k}$表示从 g_i 到 g_k 的最短距离。根据 2.2 节中的性质，将式(10)化简为：

$$\begin{aligned} p_{i\to k}&=\sum_{r=L_{i\to k}}^{L_{i\to k}+L_{de,i\to k}} M_{ik}^{r} \\ &=M_{ik}^{L_{i\to k}}\sum_{r=L_{i\to k}}^{L_{de,i\to k}} M_{ik}^{r} \\ &=M_{ik}^{L_{i\to k}}(M_{ik}^{0}+M_{ik}^{1}+\cdots+M_{ik}^{L_{de,i\to k}}) \end{aligned} \tag{11}$$

在 M 的基础上，定义给定的用于查询预测终点的轨迹 T 的先验概率 $P(T)$：

$$P(T)=P(T_{1,2,\cdots,k})=\prod_{i=1}^{k} p_{i(i+1)} \tag{12}$$

将给定的轨迹 T 扩展表示为$\{g_s,\cdots,g_c\}$，其中 g_s 表示起始网格节点，g_c 表示当前节点。针对每个网格，根据 Bayes Rule[16-17]计算 $P(g_d|T)$，从而得到轨迹终点预测结果。计算公式如下：

$$\begin{aligned} P(g_d|T)&=P(g_d|g_s,\cdots,g_c) \\ &=P(g_s,g_d|g_s,\cdots,g_c) \\ &=P(g_s,g_d|T) \\ &=\frac{P(T|g_s,g_d)\cdot P(g_s,g_d)}{P(T)} \\ &=P(T|g_s,g_d)\cdot\frac{P(g_d|g_s)\cdot P(g_s)}{P(T)} \end{aligned} \tag{13}$$

对于各个不同的网格，由于轨迹起始网格节点的先验概率 $P(g_s)$不会影响不同网格之间的相对大小，因此式(13)中可以不考虑因子项 $P(g_s)$，从而得到：

$$P(g_d|T)\propto P(T|g_s,g_d)\cdot\frac{P(g_d|g_s)}{P(T)} \tag{14}$$

然后计算两个条件概率 $P(T|g_s,g_d)$和 $P(g_d|g_s)$。

$$\begin{aligned} P(T|g_s,g_d)&=\frac{P(T,g_s,g_d)}{P(g_s,g_d)}=\frac{P(g_s,\cdots,g_c,g_d)}{P(g_s,g_d)} \\ &=\frac{P(g_s,\cdots,g_c)\cdot P(g_c,g_d)}{P(g_s,g_d)} \\ &=\frac{P(T)\cdot P(g_c\to g_d)}{P(g_s\to g_d)} \\ &=\frac{P(T)\cdot p_{c\to d}}{p_{s\to d}} \end{aligned} \tag{15}$$

其中，$P(T)$表示给定用于预测终点的轨迹先验概率，根据式(12)计算即可；$p_{c\to d}$表示从给定轨迹的当前网格转移到预测的可能终点网格的概率；$p_{s\to d}$表示从给定轨迹的起始网格转移到预测的可能终点网格的概率。对于 $p_{c\to d}$和 $p_{s\to d}$，根据马尔可夫模型推导出的式(11)计算即可。

式(14)中，影响各个网格的后验概率相对大小的另一个因子项 $P(g_d|g_s)$的计算公式如下：

$$P(g_d=g_k|g_s=g_i)=\frac{|T_{g_s=g_i,g_d=g_k}|}{|T_{g_s=g_i}|} \tag{16}$$

式(16)的具体含义可以解释为：计算出所有轨迹中起始网格为 g_i、终止网格为 g_k 的条件概率。结合式(14)和式(15)，得到预测终点网格的后验概率如下：

$$\begin{aligned} P(g_d|T)&\propto P(T|g_s,g_d)\cdot\frac{P(g_d|g_s)}{P(T)} \\ &=\frac{P(T)\cdot p_{c\to d}}{p_{s\to d}}\cdot\frac{P(g_d|g_s)}{P(T)} \\ &=\frac{p_{c\to d}}{p_{s\to d}}\cdot P(g_d|g_s) \end{aligned}$$

$$P(g_d|T)\propto\frac{p_{c\to d}}{p_{s\to d}}\cdot P(g_d|g_s) \tag{17}$$

对于一条给定预测终点的轨迹，通过对所有的网格计算后验概率，并且对所有网格的后验概率进行降序排序，得到最有可能包含轨迹终点的若干个网格。

基于马尔可夫模型的稀疏轨迹终点预测算法如算法 2 所示。

算法 2 基于马尔可夫模型的稀疏轨迹终点预测算法

输入：通过 3.2 节处理得到的轨迹序列数据集 Trj′，给定待预测终点轨迹 q，设置选取网格个数参数 K

输出：轨迹终点预测结果 prediction_result

1. 将 Trj′中轨迹序列分割为只包含相邻两个网格节点的子轨迹，计算得到马尔可夫模型一阶转移概率矩阵 M，并利用 Chapman-Kolmogoro公式得到 M_{total}。
2. 将待预测终点的轨迹 q 序列化为 q′。
3. 对于划分完成的所有网格，根据式(17)计算轨迹终点后验概率，并且按照降序排序得到概率序列：ordered_problist＝$\{p_1,p_2,p_3,\cdots,p_N\}$

   根据排序完成后的概率序列，选取对应的前 K 个网格，并分别计算轨迹预测终点的经度 Lon 和纬度 Lat：$Lon_{predict}=\sum_{i=1}^{K} p_i\cdot lon_i$，$Lat_{predict}=\sum_{i=1}^{K} p_i\cdot lat_i$。其中，$lon_i$，$lat_i$ 分别是降序排序后第 i 个网格中心的经度和纬度。

4 实验结果和分析

本节通过实验来分析基于马尔可夫模型的稀疏轨迹终点预测方法的准确性和时间效率。实验硬件环境配置：CPU 为 Intel Xeon CPU X5690 @ 3.468GHz，GPU 为 NVIDIA GeForce GTX TITAN X，内存为 140GB。实验数据采用 ECML/PKDD 15：Taxi Trajectory Prediction(I)比赛提供的数据

训练集 train. csv 和测试集 test. csv 中的轨迹数据，train. csv 中的轨迹数据主要包括了葡萄牙波尔图市 442 辆出租车从 2013 年 7 月 1 日到 2014 年 6 月 30 日的 1710671 条轨迹数据，test. csv 包含 320 条轨迹数据。train. csv 和 test. csv 中轨迹数据的具体格式为：$[[lon_1, lat_1], [lon_2, lat_2], \cdots, [lon_n, lat_n]]$，其中 $[lon_n, lat_n]$ 表示轨迹终点，相邻两个轨迹数据点之间的时间间隔为 1 分钟。统计得到：train. csv 中包含约 1. 7 千万个不同的轨迹数据点，经算法 1 第 1 步去重后得到约 9 万个数据点，去重比例为 5‰。

4.1　评价标准

本文的评价指标是通过模型计算预测所有轨迹得出的终点与轨迹实际终点之间的距离的均值 *error*。*error* 越小，表示预测的终点距离实际终点越近，准确性越高。*error* 的具体计算公式如下：

$$error=\frac{\sum_{i=1}^{n_{test}} distance(point^i_{true}, point^i_{predict})}{n_{test}} \tag{18}$$

其中，$distance(point^i_{true}, point^i_{predict})$ 表示预测终点与实际终点间的距离，计算如下：

$$distance(point_{true}, point_{predict}) = 2R \cdot \arcsin(\sqrt{a(point_{true}, point_{predict})}) \tag{19}$$

其中：

$$a(point_{true}, point_{predict}) = sin^2(\frac{\text{lat}_{\text{true}} - \text{lat}_{\text{predict}}}{2}) + \cos(lat_{true}) \cdot \cos(lat_{predict}) \cdot \sin^2(\frac{lon_{true} - lon_{predict}}{2})$$

4.2　不同网格划分方法的比较

本节对比了基于马尔可夫模型的两种不同网格划分策略对预测轨迹终点结果的影响。两种划分方法分别是均匀划分网格和基于 K-d tree 划分网格。均匀划分网格的参数如下：划分密度为 g，即将整个样本运动空间划分为 $g \times g$ 个网格；基于 K-d tree 划分网格的参数如下：每个网格容纳轨迹数据点的最大个数阈值为 n。具体实验结果如图 5 和图 6 所示。

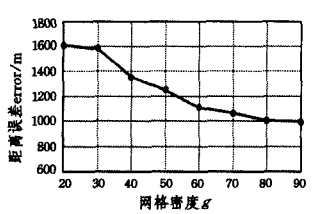

图 5　均匀网格划分

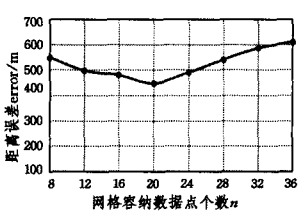

图 6　基于 K-d tree 网格划分

从图 5 可以看出，采用均匀网格划分样本运动空间的方法随着网格密度 g 的增大，*error* 逐渐减小并最终趋于稳定。从图 6 看出，K-d tree 网格划分样本运动空间的方法随着阈值 n 的逐渐增大，*error* 先增加后减小，当 $n=20$ 时，*error* 达到最小值。对比图 5 和图 6 可以得出结论：实验中选择 K-d tree 网格划分方法计算得到的 *error* 要明显小于均匀网格划分方法所得，这充分表明基于 K-d tree 网格划分样本运动空间之后，序列化得到的轨迹更接近原始轨迹，从而进一步证实了 K-d tree 网格划分样本空间比均匀网格划分更有优势。

4.3　与传统方法的比较

谈及迹终点预测，目前最常用的方法是在训练集中搜索与给定轨迹相似的轨迹，并找出前 K 条最“匹配”的轨迹。根据搜索的 K 条轨迹，计算 K 条轨迹的终点经度 *Lon* 和纬度 *Lat* 的均值作为预测终点的经、纬度。这种通过轨迹相似度预测终点的方法被称为 Baseline 算法。除此以外，深度学习[18]（Deep Learning）中的长短期记忆模型[19-21]（Long Short-Term Memory，LSTM）处理时间序列问题时也有不错的效果。因此，本节从预测精度、预测时间、轨迹覆盖率 3 方面对 Baseline、LSTM、郑宇提出的 SubSyn 算法及马尔可夫模型（Markov Model）4 种算法进行实验结果比较分析。图 7 示出了这 4 种算法的预测精度；图 8 示出了这 4 种算法的预测时间；图 9 示出了这 4 种算法在不同轨迹长度下搜索轨迹的覆盖率。

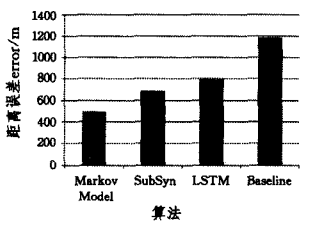

图 7　各种算法预测距离误差的对比

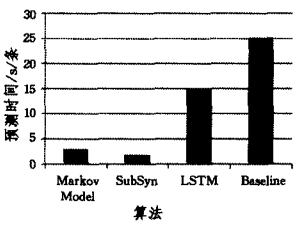

图 8　各种算法预测时间的对比

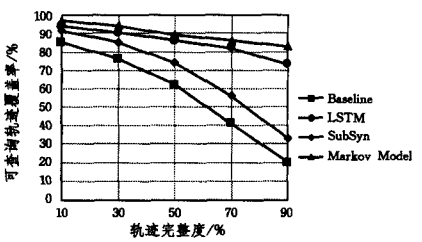

图 9　各种算法查询轨迹覆盖率的对比

从图 7 中可以看出，Baseline，LSTM 和 SubSyn 算法预测终点产生的距离误差都要大于 Markov Model 的误差。从图 8 中可以看出，Markov Model 预测一条轨迹终点的平均时间要远远少于 Baseline 算法和 LSTM 的时间，主要原因是：Baseline 算法在搜索最“匹配”轨迹时必须遍历整个训练数据集；LSTM 通过神经网络进行训练预测的过程要完成大量的矩阵计算，同样十分耗时；而 Markov Model 算法只需要从训练好的概率转移矩阵中直接提取出对应的转移概率即可，时间几乎可以忽略。Markov Model 算法的预测时间略多于 SubSyn 算法的原因是：Markov Model 算法预处理过程中基于 K-d tree 划分样本运动空间的算法的时间复杂度大于 SubSyn 算法的均匀划分算法的复杂度。综合考虑预测精度和时间复杂度，Markov Model 更能满足准确、实时地预测轨迹终点的要求。从图 9 中可以看出，随着轨迹完整度即轨迹长度的逐渐增加，Baseline 算法和 SubSyn 算法可以预测的轨迹数量迅速下降。出现这一现象的原因是：当给定轨迹长度不断增加时，训练数据集中能够与给定轨迹“匹配”的条数越来越少；而 Markov Model 和 LSTM 受轨迹完整度的影响很小。因此，可以得出结论：Baseline 算法在解决稀疏轨迹终点预测问题方面存在明显缺陷，而 Markov Model 可以正常预测稀疏轨迹终点。

（下转第 224 页）

Journal of Operational Research, 2012, 217(1): 204-213.

[12] KIM B, KIM S, PARK J. A school bus scheduling problem[J]. European Journal of Operational Research, 2012, 218(2): 577-585.

[13] CHEN X, KONG Y, DANG L, et al. Exact and Metaheuristic Approaches for a Bi-objective School Bus Scheduling Problem [J]. Plos One, 2015, 10(7): e0132600.

[14] NANRY W P, BARNES J W. Solving the pickup and delivery problem with time windows using reactive tabu search[J]. Transportation Research Part B Methodological, 2000, 34(2): 107-121.

[15] WU B. Particle Swarm Optimization for Vehicle Routing Problem and its Application[D]. Hangzhou: Zhejiang University of Technology, 2008. (in Chinese)
吴斌. 车辆路径问题的粒子群算法研究与应用[D]. 杭州: 浙江工业大学, 2008.

[16] DANG L X, HOU Y E, KONG Y F. Spatiotemporal Neighborhood Search for Solving Mixed-load School Bus Routing Problem[J]. Computer Science, 2015, 42(4): 221-225. (in Chinese)
党兰学, 侯彦娥, 孔云峰. 时空相关的混载校车路径问题邻域搜索[J]. 计算机科学, 2015, 42(4): 221-225.

[17] PENNA P H V, SUBRAMANIAN A, OCHI L S. An iterated local search heuristic for the heterogeneous fleet vehicle routing problem[J]. Journal of Heuristics, 2013, 19(2): 201-232.

(上接第 197 页)

结束语 本文提出了一种利用马尔可夫模型对城市移动对象稀疏轨迹终点预测的方法。在此过程中，提出并证明了比均匀划分网格更有优势的基于 K-d tree 网格划分样本运动空间的算法。然后，本文详细地阐述了预测方法和步骤。最后，本文通过实验对比了 Baseline，LSTM，SubSyn 和 Markov Model 4 种不同预测算法的精度、时间效率和轨迹覆盖率，综合得出马尔可夫模型可以有效解决稀疏轨迹终点预测问题。

本文中研究的不足之处是仅仅使用了轨迹数据信息，并没有考虑路况、天气、移动对象的运行速度等上下文信息，因此轨迹终点预测还有待进一步研究。在未来工作中，可以对上下文信息进行特征抽取并将其加入到预测模型中，以进一步提高预测的精度。

参 考 文 献

[1] GONZALEZ M C, HIDALGO C A, BARABASI A L. Understanding individual human mobility patterns[J]. Nature, 2008, 453(7196): 779-782.

[2] SONG C, QU Z, BLUMM N, et al. Limits of predictability in human mobility[J]. Science, 2010, 327(5968): 1018-1021.

[3] ZHENG Y, ZHOU X F, et al. Computing with spatial trajectories[M]. Springer Science & Business Media, 2011.

[4] MAMOULIS N, CAO H, KOLLIOS G, et al. Mining, indexing, and querying historical spatiotemporal data[C]// Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2004: 236-245.

[5] MORZY M. Mining frequent trajectories of moving objects for location prediction[M]// Machine Learning and Data Mining in Pattern Recognition. Springer Berlin Heidelberg, 2007: 667-680.

[6] JEUNG H, LIU Q, SHEN H T, et al. A hybrid prediction model for moving objects[C]// IEEE 24th International Conference on Data Engineering, 2008(ICDE 2008). IEEE, 2008: 70-79.

[7] YING J J C, LEE W C, WENG T C, et al. Semantic trajectory mining for location prediction[C]// Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems. ACM, 2011: 34-43.

[8] ZHENG Y, ZHANG L, XIE X, et al. Mining interesting locations and travel sequences from GPS trajectories[C]// Proceedings of the 18th International Conference on World Wide Web. ACM, 2009: 791-800.

[9] XUE A Y, ZHANG R, ZHENG Y, et al. Destination prediction by sub-trajectory synthesis and privacy protection against such prediction[C]// 2013 IEEE 29th International Conference on Data Engineering (ICDE). IEEE, 2013: 254-265.

[10] XUE A Y, QI J, XIE X, et al. Solving the data sparsity problem in destination prediction[J]. The VLDB Journal, 2015, 24(2): 219-243.

[11] GHAHRAMANI Z. An introduction to hidden Markov models and Bayesian networks[J]. International Journal of Pattern Recognition and Artificial Intelligence, 2001, 15(1): 9-42.

[12] Handbook of Markov Chain Monte Carlo[M]. CRC Press, 2011.

[13] 汪荣鑫. 随机过程[M]. 西安: 西安交通大学出版社, 2006.

[14] KARLIN S. A first course in stochastic processes[M]. Academic Press, 2014.

[15] BAKULA M K, NIKODEM K. On the converse Jensen inequality for strongly convex functions[J]. Journal of Mathematical Analysis and Applications, 2016, 434(1): 516-522.

[16] STONE J V. Bayes' rule: a tutorial introduction to Bayesian analysis[M]. Sebtel Press, 2013.

[17] BERNARDO J M, SMITH A F M. Bayesian theory[J]. Journal of the Royal Statistical Society, 2000, 15(19): 13-23.

[18] LIU J W, LIU Y, LUO X L. Research and development on deep learning[J]. Application Research of Computers, 2014, 31(7): 1921-1930. (in Chinese)
刘建伟, 刘媛, 罗雄麟. 深度学习研究进展[J]. 计算机应用研究, 2014, 31(7): 1921-1930.

[19] GREFF K, SRIVASTAVA R K, KOUTNÍK J, et al. LSTM: A search space odyssey[J]. IEEE Transactions on Neural Networks & Learning Systems, 2015, pp(99): 1-11.

[20] WÖLLMER M, KAISER M, EYBEN F, et al. LSTM-Modeling of continuous emotions in an audiovisual affect recognition framework[J]. Image and Vision Computing, 2013, 31(2): 153-163.

[21] MONNER D, REGGIA J A. A generalized LSTM-like training algorithm for second-order recurrent neural networks[J]. Neural Networks, 2012, 25(1): 70-83.