历史轨迹数据的学习型索引

摘要

近年来，随着无线通信技术的飞速发展和嵌入式设备的广泛应用，车辆轨迹数据呈爆炸式增长。传统索引结构在处理大规模数据时面临显著的存储和查询性能挑战。随着机器学习技术的不断进步，研究者开始探索通过构建学习型索引模型来拟合数据分布，替代部分传统数据结构，提升查询效率并降低存储开销。学习型索引技术通过对底层数据分布和查询负载特征进行建模和优化，能够动态调整索引结构，减少访存空间需求，从而为大规模轨迹数据的快速查询提供新的解决方案。

在学习和总结前人学习型索引方法的基础上，本文实现了面向历史轨迹数据的学习型索引。采用图结构与深度优先搜索（DFS）对道路进行排序，确保道路拓扑结构的有序性。创新之处在于结合 Hilbert 曲线与勒贝格测度设计映射函数，将二维轨迹点映射到一维空间，提升数据在空间中的局部性表达能力。对于降维后的轨迹数据，采用分段线性回归模型进行分片预测，并利用局部模型精确预测每个轨迹点的存储位置。实验结果表明，该方法减少存储空间需求并提高查询性能。

关键词: 历史轨迹；学习型索引；Hilbert 曲线

Learning-based Indexing for Historical Trajectory Data

ABSTRACT

In recent years, with the rapid development of wireless communication technology and the widespread application of embedded devices, vehicle trajectory data has been growing explosively. Traditional indexing structures face significant challenges in storage and query performance when dealing with large-scale data. With the continuous advancement of machine learning technology, researchers have begun to explore the construction of learning-based indexing models to fit data distribution, replacing some traditional data structures to improve query efficiency and reduce storage overhead. Learning-based indexing technology, by modeling and optimizing the underlying data distribution and query load characteristics, can dynamically adjust the indexing structure to reduce memory access requirements, thereby providing a new solution for the fast querying of large-scale trajectory data.

Based on the study and summary of previous learning-based indexing methods, this paper has implemented a learning-based indexing system for historical trajectory data. A graph structure and depth-first search (DFS) are used to sort the roads, ensuring the orderliness of the road topology. The innovation lies in the combination of Hilbert curves and Lebesgue measure to design a mapping function that projects two-dimensional trajectory points into a one-dimensional space, enhancing the spatial locality expression of the data. For the reduced-dimension trajectory data, a piecewise linear regression model is employed for segment-based prediction, and local models are used to accurately predict the storage location of each trajectory point. Experimental results show that this method reduces storage space requirements and improves query performance.

Key words: Historical trajectories; learned index; Hilbert curves

目 录

1 绪论

1.1 课题背景及目的

随着车载传感网、智能设备和移动计算技术的不断发展，车辆轨迹数据的规模呈现爆炸式增长[1]。这些数据包含了移动对象丰富的时空信息，成为智能交通、智慧城市等应用的重要信息来源。为了支持高效的轨迹查询和分析，构建合理的时空数据索引结构至关重要。

轨迹查询通常根据目标时间分为三类：未来查询、当前查询和历史查询。其中，历史轨迹数据的查询是其他两类查询以及更复杂轨迹分析的基础。通过历史数据，可以挖掘出移动对象的行为模式、热点区域的分布特点以及长期趋势。这些信息不仅对轨迹预测有重要意义，还为实时查询提供了必要的上下文支持。然而，面对轨迹数据的复杂性和分布不均衡问题，如何在保证查询效率的同时压缩索引体积，已成为当前亟须解决的核心问题。

在进行数据查询时，通常要构建以读为主的索引来支持各种查询。早期研究主要聚焦于传统索引结构。典型的传统索引结构有用于范围查询的 B 树索引、用于点查询的哈希索引，以及用于存在性查询的布隆过滤器等。然而随着数据规模爆发性增长，传统索引处理海量数据时面临随着数据量增大结构变大的问题，这使得查询时待搜索的节点变多而需要较多查询时间。

为了应对这一挑战，近年来学习型索引成为研究热点。这类方法将索引问题转化为函数拟合问题，通过机器学习模型对数据分布进行建模，以预测记录的位置或区间，从而代替传统的索引遍历过程。典型的一维学习型索引代表性工作如 Kraska 提出的 RMI[2](Recursive Model Index)，该索引通过训练模型逼近数据的累积分布函数，在查询效率和存储开销方面均优于 B 树等传统结构。历史轨迹数据属于时空数据，往往使用多维学习型索引进行查询，多维学习型索引分为降维方式的学习型索引和非降维方式的学习型索引。前者将多维数据映射到一维空间中，再针对降维后的数据构建一维学习型索引，如基于 Z-order 曲线降维的 RSMI[3](Recursive Spatial Model Index)；后者则直接在高维空间中建立模型，避免了降维过程中可能造成的空间信息损失，例如 Flood[4] 索引。与传统索引相比，学习型索引不仅能大幅减少索引大小，还能在数据分布良好建模的场景下实现近似常数时间查询，极大地提升了轨迹数据管理效率。因此，围绕历史轨迹数据构建高效、轻量的学习型索引结构具有显著的现实应用价值。

1.2 国内外研究现状

学习型索引是一种基于机器学习的高效索引方法，主要通过拟合数据的累积分布函数（CDF）来预测键的位置，实现对有序数据的高效访问。相较传统索引结构，其更扁平、空间利用率高、查找速度快，广泛用于一维和多维数据场景。

在一维学习型索引中，RMI[2]（Recursive Model Index）通过递归模型层次结构逼近 CDF，实现键值映射，但不支持动态更新且分层复杂。ALEX[5]（Adaptive Learned Index）对此改进，引入叶节点大小限制和两种数据结构（GA 和 PMA）以支持更新，但牺牲了查询效率，且无误差控制。

多维学习型索引分为降维和非降维两类。降维方法如 RSMI[3]、LISA[6]、ML[7] 先将多维数据映射为一维数据，再使用 RMI 建模。RSMI 基于 Z-order 曲线进行空间映射，但维护成本高；LISA 利用分段线性回归建模，查询快但更新慢；ML 借鉴 iDistance 思想，仅支持只读场景。非降维方法如 FloodFlood[4]、Tsunami[8]、SPRIG[9] 等直接在多维空间建模。Flood 适用于分布均匀数据，性能在数据倾斜时下降；Tsunami 通过增强网格和 CDF 提高性能，但仍不支持动态更新；SPRIG 结合自适应网格、空间填充曲线与插值函数，支持高性能查询，能学习空间分布，但仅限二维数据，存储开销较大。

2 相关技术介绍

2.1 降维方式的多维学习型索引

学习型索引可以通过学习数据分布预测键存储的估计位置，从而与基于树遍历查询的传统索引相比，学习型索引减少了内存开销和查询时间。在多维数据管理中，高维特征容易导致学习型索引的结构复杂，会降低查询性能，所以许多学习型索引会采用降维方式，即将多维数据映射到一维空间，再基于降维后的空间建立学习型索引，从而实现更加紧凑、更高效的索引结构。典型的降维方式的多维学习型索引有 RSMI、LISA、ML 等。

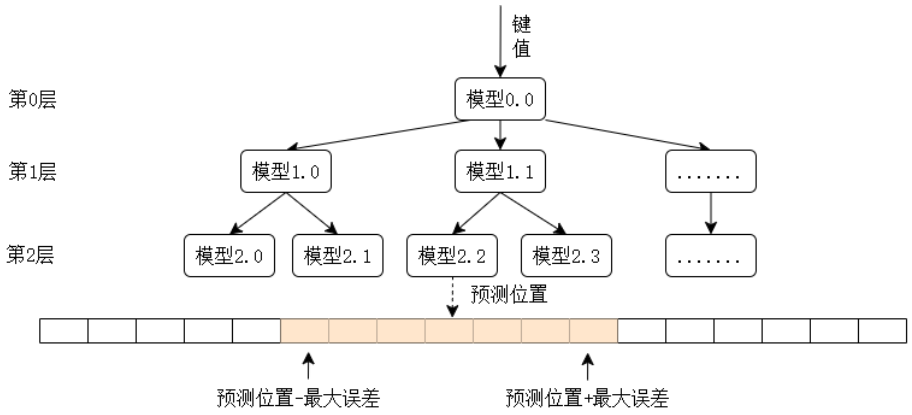

图 2.1 RSMI 结构图

RSMI 是基于 RMI 将多维数据映射到一维空间的多维学习型索引，图 2-1 是 RSMI 的索引结构，他通过学习将 p.key 映射到存储块 ID，即，但是由于模型会有误差，所以要在误差范围里面查询。他采用 Z-order 空间填充曲线将多维数据转化为一维映射值，但是在降维之前，他先将原始数据映射到了一个排序空间，映射是在排序空间中进行的。

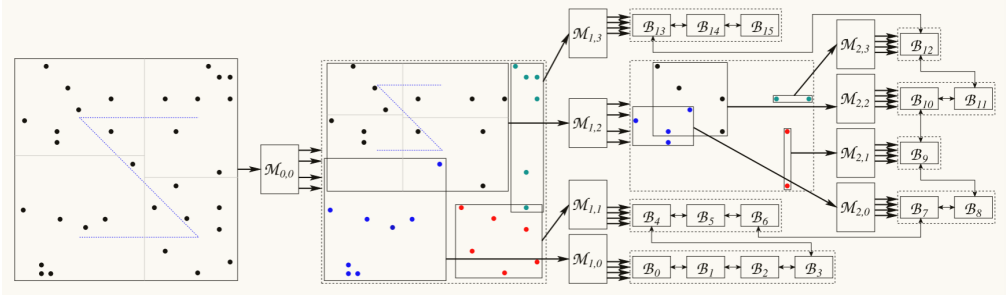

图 2.2 RSMI 索引结构构建过程 (N=8,B=2)

在数据规模较大的情况下，使用单个函数很难将所有数据点映射到他们的等级，RSMI采用了递归划分分区的方式，首先将数据集进行分区，对每个分区训练模型，如果分区的数据规模仍然超过规定阈值，则继续分区，一直执行该递归过程，直到所有分区的数量可以满足规定阈值，对每个符合阈值的分区训练模型 M 预测点所在的逻辑块。图 2-2 展示了在规定每个叶模型最多预测 $N=8$ 个数据点，共有 $n=32$ 个数据。每个逻辑块存储两个数据的例子的递归划分分区的过程。首先将数据空间切分成 n/N 个网格，并且保证横坐标分割的列数都是 $2^{\lfloor \log_4 n/N \rfloor}$，每一列平分所有数据点，对每一列切成 $2^{\lfloor \log_4 n/N \rfloor}$ 个方格。对于分割后的方格，用 Z-order 空间填充曲线为每一个方格赋予一个曲线值，训练一个映射函数将点映射到曲线上，训练完毕后，再用它来预测每个点的曲线值，但是并不准确。RSMI 将预测到的映射值对数据进行重新分组，对于每个组如果数据点超过 N，则重复前面分割过程，否则学习一个函数来预测 *p.blk*，该函数就是图 2-1 中的叶子模型。

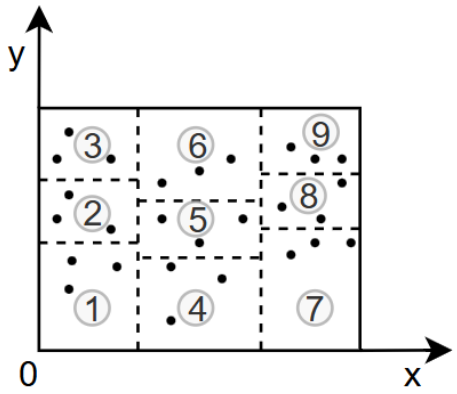

图 2.3　网格划分策略

LISA 是一种基于磁盘的多维学习型索引，该索引对数据进行降维后，先学习数据的分布特征，最后才根据分布特征把数据写入磁盘。如图 2-3 所示，对于空间数据划分为多个部分，使得每个部分的数据量近似相等，并把每个维度的分割点保存下来。每个方格的编号方式如图所示，映射函数满足以下条件

$$M(x_i \in V) < M(x_j \in V) \text{ when } i < j, \\ \text{where } x_i \in C_i \text{ and } x_j \in C_j. \tag{2-1}$$

网格内会有多个点，两个点的映射值应该不同，因此利用勒贝格测度 μ 定义了映射函

数 M。假设 $x=(x_0,x_1,\ldots,x_{d-1})$，并且 $x\in C_i=[\theta_{j_0}^{(0)},\theta_{j_0+1}^{(0)})\times\cdots\times[\theta_{j_{d-1}}^{(d-1)},\theta_{j_{d-1}+1}^{(d-1)})]$，则

$$M(x)=i+\frac{\mu(H_i)}{\mu(C_i)} \tag{2-2}$$

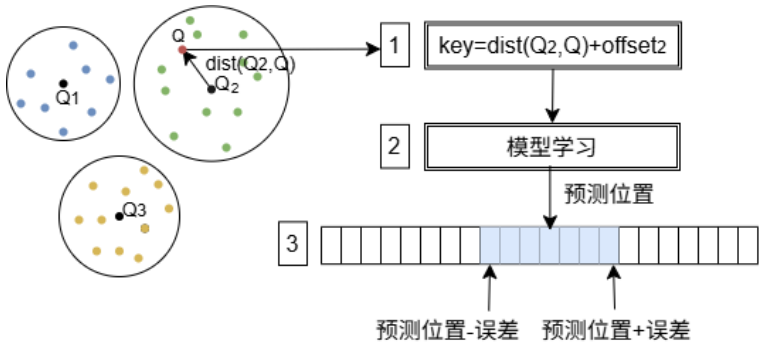

图 2.4 ML 索引结构

ML 一种内存只读多维学习型索引，与 RSMI 索引采用空间填充曲线将多维数据转化成一维数值的方法不同，ML 索引采用类似于 iDistance[23] 的方法进行降维。ML 索引的结构如图 2-4 所示，索引主要由两部分构成：第一部分是一组参考点；第二部分由一个学习模型和一个有序数组构成。更具体地，选定选择 m 个有序参考点 Q_i，每个参考点 Q_i 可以被认为是数据分区 P_i 的质心。ML 索引首先计算各个参考点与数据之间的欧式距离，与参考点距离相近的数据被划分到参考点对应的分区中；然后，根据分区偏移量以及每个数据 p 与所在分区参考点之间的距离来计算每个数据的一维映射值，计算公式如下

$$key=offset_i+dist(Q_i,p) \tag{2-3}$$

其中 $offset_i$ 的值为分区的半径之和，公式为

$$offset_i=\sum_{j<i}r(j) \tag{2-4}$$

其中 r 是分区中离参考点最近的点到参考点的距离，该累加计算方法可以使不同分区 P_i 的 $offset_i$ 设置的值不同。最后采用 RMI 模型来学习一维映射值的分布，所有模型都是线性模型。

2.2 Hilbert 曲线

Hilbert 曲线（Hilbert Curve）是一种将高维空间连续地映射到一维空间的数学构造方法，由德国数学家 David Hilbert 于 1891 年提出。Hilbert 曲线是一种分形曲线，能填充满一个平面正方形，形成的方式就是将正方形逐层四等分，从其中一个正方形的中心开始，依次穿线，穿过正方形的中心。这是一个递归过程，每一次递归细分被称为“一阶数”，每次递归过后，曲线会变得更加复杂，阶数越高当然映射精度也会越复杂。这种曲线不仅呈现重复性和相似性，能在有限的区域内有无限长的空间。图 2-5 呈现了该曲线的一阶、二阶、三阶的递归编码过程。

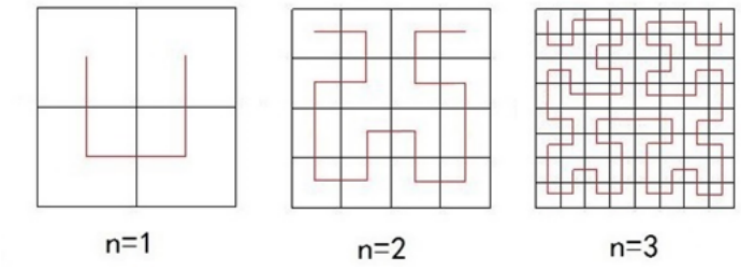

图 2.5 Hilbert 曲线

Hilbert 曲线有三个核心特性。第一个是空间填充性，该曲线可以遍历多维空间中的所有单元格，并且不留空隙。第二个是空间局部性，空间中相邻的点经过 Hilbert 曲线编码后，在映射为一维空间后会尽可能的邻近，不会突然跨越很远的空间区域。第三个是连续不可导的特性。这些特性使得它在计算机科学、图像处理、地理信息系统等多个领域中得到了广泛地应用。在计算机科学领域中，数据压缩、图像处理、计算机图形学以及加密和解密等领域都有应用 Hilbert 曲线。在图像处理领域，Hilbert 曲线能够执行压缩、色调处理和文本分析等多种任务，同时还能有效提升光线追踪等渲染算法的性能。在遥感与地图制图方面，Hilbert 曲线常用于生成高分辨率地图数据，优化地理信息系统的数据管理与检索效率。在生物信息学领域，希尔伯特曲线也被用于基因序列分析，辅助挖掘生物大数据中的潜在模式，为生命科学研究提供新的技术手段。

2.3 分段线性回归模型

分段线性回归是指当 y 对 x 的回归在 x 的某一范围服从某种线性关系，在其他范围内又服从斜率不同的线性关系时适用的一种回归估计方法。该方法通过引入指示变量，将不同区间的数据纳入同一个回归模型中，从而在统一建模框架下分别拟合各段的线性关系。具体来说分段线性回归模型可以表示为：

$$f(x)=\begin{cases}\alpha+\alpha_0(x-\beta_0), & \beta_0\le x<\beta_1\\ \alpha+\alpha_0(x-\beta_0)+\alpha_1(x-\beta_1), & \beta_1\le x<\beta_2\\ \vdots & \vdots\\ \alpha+\alpha_0(x-\beta_0)+\alpha_1(x-\beta_1)+\cdots+\alpha_\sigma(x-\beta_\sigma), & \beta_\sigma\le x\end{cases} \tag{2-5}$$

其中 σ 表示表示一共分割的段数，$\{\alpha_0,\alpha_1,\ldots,\alpha_\sigma\}$ 和 $\{\beta_0,\beta_1,\ldots,\beta_\sigma\}$ 分别表示斜率序列和分割点序列。为了更好拟合数据，定义一个损失函数：

$$L(\alpha,\beta)=\sum_{i=0}^{V}(f(x_i)-y_i)^2 \tag{2-6}$$

其中 $\alpha=(\alpha_0,\alpha_1,\ldots,\alpha_\sigma),\beta=(\beta_0,\beta_1,\ldots,\beta_\sigma)$。模型求解目标是找到使损失函数最小化的参数组合。该优化问题通过以下迭代步骤实现：

步骤一：设固定断点集合 $\hat{\beta}=(\hat{\beta}_0,\hat{\beta}_1,\ldots,\hat{\beta}_\sigma)$，此时斜率参数 α 可通过线性最小二乘求解：

$$\begin{bmatrix}1 & x_0-\hat{\beta}_0 & (x_0-\hat{\beta}_1)1_{x_0\ge\hat{\beta}_1} & \cdots & (x_0-\hat{\beta}_\sigma)1_{x_0\ge\hat{\beta}_\sigma}\\ 1 & x_1-\hat{\beta}_0 & (x_1-\hat{\beta}_1)1_{x_1\ge\hat{\beta}_1} & \cdots & (x_1-\hat{\beta}_\sigma)1_{x_1\ge\hat{\beta}_\sigma}\\ \vdots & \vdots & \ddots & \vdots & \vdots\\ 1 & x_V-\hat{\beta}_0 & (x_v-\hat{\beta}_1)1_{x_v\ge\hat{\beta}_1} & \cdots & (x_v-\hat{\beta}_\sigma)1_{x_v\ge\hat{\beta}_\sigma}\end{bmatrix} \tag{2-7}$$

其中指示函数 $1_{x_i\ge\hat{\beta}_j}$ 满足：

$$1_{x_i\ge\hat{\beta}_j}=\begin{cases}1, & x_i\ge\hat{\beta}_j\\ 0, & \text{其他}\end{cases} \tag{2-8}$$

通过计算可以求出 α 的求解方式为：

$$\alpha=(A^TA)^{-1}A^Ty \tag{2-9}$$

步骤二：下一步是选择一个最佳断点。设 $\tilde{\alpha}(\beta)$ 是特定断点 β 的最优 α，则问题转化为：

$$\min_{\beta} L(\tilde{\alpha}(\beta),\beta) \tag{2-10}$$

通过拟牛顿法更新，计算修正方向 $s(k)$：

$$s(k) = -Y^{-1}\nabla_{\beta} L \tag{2-11}$$

其中 Y 是 Hessian 矩阵的近似，$\nabla_{\beta} L$ 是梯度。断点更新公式为：

$$\beta^{(k+1)} = \beta^{(k)} + \eta(k)\cdot s(k) \tag{2-12}$$

其中 $\eta(k)$ 为自适应步长，迭代至损失函数收敛。

与传统线性回归模型相比，分段线性回归模型显著优势。一方面，它能更精确地拟合具有非线性趋势的数据，因为假设了每个分段内的数据存在线性关系，提高了对数据变化规律的拟合度；另一方面，由于每个分段的回归模型是独立的，分段线性回归模型能够更有效地应对数据的结构突变和趋势转折点，对变化的数据有更强的适应性。

参考文献

[1] 冯钧, 张立霞, 陆佳民, et al. 路网环境下的移动对象查询技术研究综述. 软件学报, 28(06):1606–1628, 2017.

[2] Tim Kraska, Alex Beutel, Ed H. Chi, et al. The case for learned index structures. In *Proceedings of the 2018 International Conference on Management of Data*, SIGMOD '18, pages 489–504, New York, NY, USA, 2018. ACM.

[3] Vikram Nathan, Jialin Ding, Mohammad Alizadeh, et al. Learning multi-dimensional indexes. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*, SIGMOD '20, pages 985–1000, New York, NY, USA, 2020. ACM.

[4] Jianzhong Qi, Guoliang Liu, Christian S. Jensen, et al. Effectively learning spatial indices. *Proceedings of the VLDB Endowment*, 13(12):2341–2354, 2020.

[5] Jialin Ding, Umar Farooq Minhas, Jia Yu, et al. Alex: An updatable adaptive learned index. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*, SIGMOD '20, pages 969–984. ACM, 2020.

[6] Pengfei Li, Hua Lu, Qian Zheng, et al. Lisa: A learned index structure for spatial data. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*, SIGMOD '20, pages 2119–2133. ACM, 2020.

[7] Angjela Davitkova, Evica Milchevski, and Sebastian Michel. The ml-index: A multidimensional, learned index for point, range, and nearest-neighbor queries. In *EDBT*, pages 407–410, 2020.

[8] Jialin Ding, Vikram Nathan, Mohammad Alizadeh, et al. Tsunami: A learned multi-dimensional index for correlated data and skewed workloads. *arXiv preprint arXiv:2006.13282*, 2020.

[9] Sheng Zhang, Subrata Ray, Ruihong Lu, et al. Sprig: A learned spatial index for range and knn queries. In *Proceedings of the 17th International Symposium on Spatial and Temporal Databases*, pages 96–105. ACM, 2021.