ISSN 1009-3044
Computer Knowledge and Technology 电脑知识与技术
Vol.16, No.33, November. 2020
E-mail: wltx@dnzs.net.cn
http://www.dnzs.net.cn
Tel:+86-551-65690963 65690964

面向轨迹大数据的压缩空间算法研究

王伟

(河南工业贸易职业学院 信息工程系,河南 郑州 451191)

摘要:大量定位技术和移动技术的成熟为多源、多模态数据的增加奠定了基础,其中有关运动物体轨迹的GPS轨迹数据增长尤为突出。因此,提出了一种路网轨道压缩算法。首先,在预处理步骤中,将轨迹分解为空间路径和时间序列。其次,在压缩步骤中,设计对空间路径执行空间压缩的算法,以及对时间序列并行执行时间压缩的算法。

关键词:路网;空间压缩算法;时间压缩算法

中图分类号:TP311.13 文献标识码:A

文章编号:1009-3044(2020)33-0039-02

DOI:10.14004/j.cnki.ckt.2020.3367

开放科学(资源服务)标识码(OSID):

存储或传输大量由位置采集技术产生的轨迹数据是非常具有挑战性的。在一些研究中通过减少轨迹数据中的位置点的信息冗余大幅度减少存储需求和通信负载[1]。压缩方法通常需要在压缩比和最大误差之间进行了权衡,一般来说,压缩比越高,压缩轨迹数据的质量就越差,反之亦然。如果相应的轨迹应用或者相应用户能够容忍一定范围内的轨迹误差,trajic算法是一种很合适的算法,其在一定误差范围内可以获得相对良好的压缩比。它具有预测下一位置点数据的预测器,以及一个产生小残差的残差编码方案,该方案可用于补偿预测值和实际值之间差异。

1 问题模型与符号定义

传统的方法通过n个三元组的序列表示一个轨迹T,其形式类似($<x_1,y_1,t_1>,<x_2,y_2,t_2>,...,<x_n,y_n,t_n>$),其中$<x_i,y_i>$记录移动物体的经度和纬度,而$t_i$则表示移动物体处于当前位置的时间。

轨迹分解作为轨迹压缩的第一步,从而为轨迹压缩奠定基础。COMPRESS轨迹压缩算法是基于道路网络实现的,为此首先将路网建模为有向图$G=(V,E)$,其中V是顶点集合,E是边集合。边e上的权重被表示为$w(e)$,可以是物理距离、旅行时间或其他成本,具体取决于不同的应用环境。轨迹是运动物体随时间在空间中的运动痕迹。因此,它包含空间信息和时间信息[2]。

原始轨迹数据三元组$<x_i,y_i,t_i>$说明了移动对象在时间t_i时,位于$<x_i,y_i>$的位置。假设$<x_i,y_i>$为轨迹的起点,如果已知第i个点的位置$<x_i,y_i>$,那么从起始点到第i个点的距离d_i必是确定的,但如果我们知道移动物体从时间t_1到t_i的距离d_i,对应于时间戳t_i的位置$<x_i,y_i>$却是无法确定的。因此,轨迹分解记录轨迹T所经过的空间路径$<e_1,e_2,...,e_m>$,并根据d_i计算转换回$<x_i,y_i>$,其中$<e_1,e_2,...,e_m>$表示的移动物体在轨迹T中所经过的m条边的序列。如此一来算法只需要压缩$<e_1,e_2,...,e_m>$边的序列即可,这更意义实现,压缩效率也会更高。

2 基于最短路径的空间压缩算法

假设$SP<e_i,e_j>$表示从边e_i到边e_j的最短路径,而$SP_{end}<e_i,e_j>$则表示$SP<e_i,e_j>$路径的最后一条边。其中每条边的数字表示该路径的距离,则$SP<e_{15},e_7>=<e_{15},e_{12},e_9,e_{10},e_7>$,$SP_{end}<e_{15},e_7>=e_{10}$,$SP_{end}<e_{15},e_{10}>=e_9$。

路径的距离并不是空间距离,COMPRESS定义距离为$d_i=d_{pi}\times v_i$,其中d_{pi}表示当前路径的空间距离,v_i表示经过该路径的平均时间差。如此一来,压缩算法的核心思想在于跳过那些符合最短路径子轨迹序列中的边。原始轨迹$T=<e_{15},e_{12},e_9,e_6,e_3>$,在边$e_{15}$与$e_9$之间满足最短路径可以用$e_{15},e_9$表示,同理可继续进行压缩,最后得到压缩后轨迹$T'=<e_{15},e_3>$。

3 时间序列压缩算法

传统方法中,时间同步欧氏距离(Time Synchronized Euclidean Distance,TSED)是用于评估误差上限的重要度量。与TSED不同,本文提出两种新的误差评估度量-时间同步网络距离(Time Synchronized Network Distance,TSND)和网络同步时间差异(Network Synchronized Time Difference,NSTD),其定义如下:

定义1:TSND是给定一个时间序列TS_T及其对应的压缩序列TS_T^c,TNSD用于度量移动物体在两条轨迹上相同时间戳下距离的最大差异。

$$TSND=\max_{t_x\in[0,t_{max}]}(\left|d(t_x,TS_T)-d(t_x,TS_T^c)\right|)$$

定义2:NSTD是给定一个时间序列TS_T及其对应的压缩序列TS_T^c,NSTD用于度量移动物体在两条轨迹上相同距离下的最大时间差。

$$NSTD=\max_{d_x\in[0,d_{max}]}(\left|t^{min}(d_x,TS_T)-t^{min}(d_x,TS_T^S)\right|,\left|t^{max}(d_x,TS_T)-t^{max}(d_x,TS_T^S)\right|)$$

用于压缩时间序列的有损压缩算法需要满足以下两个核心目标:首先,压缩轨迹与原始轨迹之间的TSND和NSTD必须

收稿日期:2020-05-28
基金项目:河南省科技攻关项目:城市计算范畴下基于GPU集群的时空大数据并行计算平台(182102210021);河南省高等学校重点科研项目,厅级(18A520014);河南省高等学校重点科研项目计划支持(项目编号:21B520004)
作者简介:王伟(1982—),女,河南林州人,本科,讲师,研究方向:大数据分析、人工智能系统。

被限定到一定的误差值范围内，该阈值根据具体用户或者应用而决定。其次，压缩率越高越好。

1)严格贯穿折线压缩算法

轨迹分解之后，时间序列是一系列由距离和时间构成的二维元组。因此，在一个距离-时间图中可以绘制出原始的时间序列图，而任何一条轨迹的时间序列都是一条贯穿该轨迹时间序列所有点的折线。对于一个给定值为τ误差TSND，那么可以围绕轨迹中n个点，并每个点为中心绘制长度为2τ的垂直线段(如图1所示)。如果将这n条垂直线当成对象，然后时间序列压缩问题可以简化为一个计算几何问题，称为贯穿折线问题。该问题的本质是找到一条折线，该折线包含按顺序通过多个对象的最少定点数。如图1，从四个点压缩为3个点，其压缩率为1.33。与传统贯穿折线定义不同，本文提出新的多限制贯穿折线，其中既考虑了TSND问题也考虑了NSTD问题[3,4]。

定义3：贯穿折线是指横穿原始时间-距离折线中所有n个垂直线段和以垂直线段为中心的n个水平线段构成的矩形区域所形成的一条折线。

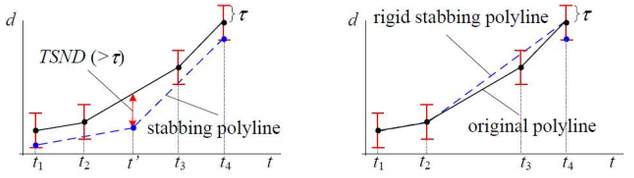

图1　贯穿折线和严格贯穿折线示例图

如图1左图所示，符合定义的贯穿折线只能保证在原始折线取样时间戳点才能保证TSND限制在τ，在其他时间节点则无法保证如时间点t'。为此，给出了严格贯穿折线的定义(如图1右图所示)：

定义4：严格贯穿折线是一条所有顶点都属于原始折线顶点子集的贯穿折线。

理论研究证明搜索符合条件的严格贯穿折线并用于时间序列压缩其计算复杂度可以限制在$O(n)$，但相应的算法无法保证获取最优解，而最优算法的计算复杂度达到了$O(n^2)$。

2)管道贯穿折线压缩算法

如上所述，贯穿阵线压缩算法缺乏严格的限制，从而无法将压缩时间序列的TSND和NSTD误差限制在某个范围内，而严格贯穿折线压缩算法则给出了过多的限制，在保证误差范围的同时也将大量不必要的时间点引入到了压缩数据中，这降低了压缩的比率，增加了算法的计算量。基于此，管道贯穿折线压缩算法得以提出。

与贯穿折线相似，原始时间序列被映射为时间-距离空间中的一条折线，以原始折线的每一个顶点作为中心绘制长度为2τ的垂直线段。然后以连续垂直线段的上顶点和下顶点作为多边形顶点，构建一个多边形称为TSND管道P_d。原始时间序列这些位于该管道的中心，因此，只要管道贯穿折线完全位于TSND管道P_d内部，那么就可以确保压缩时间序列和原始时间序列之间的TSND误差不超过τ。相较于严格贯穿折线压缩算法(3顶点)，管道贯穿折线仅需要2个顶点。

根据同样的算法逻辑，在给定一个逻辑时间误差上限η情况下可以计算获取NSTD管道P_t。因此在给定距离和时间误差τ和η的情况下，综合TSND和NSTD度量可以获得一个综合管道$P = P_t \cup P_d$。因此管道贯穿折线压缩算法的核心可以表述为找到一条折线该折线完全位于管道P内且具有最少的顶点。

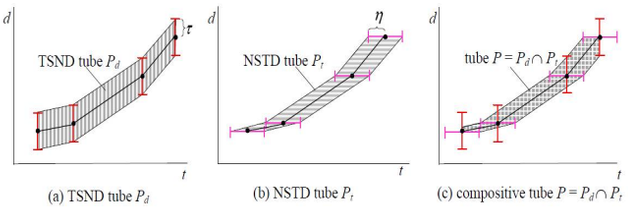

图2　管道贯穿折线算法示例

基于上述论述，TSND管道、NSTD管道和综合管道的定义介绍如下：

定义5：TSND管道是给定一个时间序列以及相应的TSND误差限制τ，则相应的TSND管道P_d是包含2n个顶点的多边形。

定义6：NSTD管道是给定一个时间序列以及相应的NSTD误差限制η，则相应的NSTD管道P_t是一个包含2n个顶点的多边形。

定义7：综合管道是给定一个时间序列以及TSND和NSTD误差分别为τ和η，而P_d和P_t分别表示TSND管道和NSTD管道，那么P_d与P_t的交叉形成的归为称为综合管道$P(P = P_t \cup P_d)$。

定义8：综合管道贯穿折线是给定一个时间序列及其相应的综合管道P，则综合管道贯穿折线就是完全位于该管道内的一条折线，该折线连接了(d_1,t_1)与(d_n,t_n)。

4 对比实验

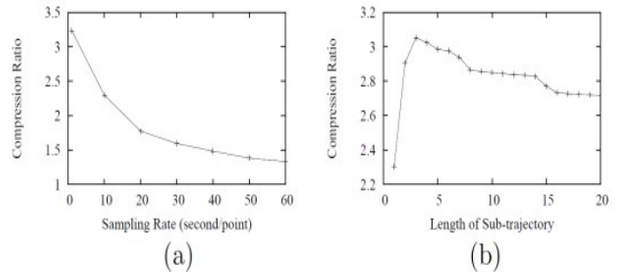

图3　空间无损压缩算法的压缩比率

图3(a)给出了在不同取样率下的压缩率，结果显示取样率对于压缩率影响相对要少得多，在取样率从1秒/点到60秒/点的变化过程中，本文提出的空间压缩算法的平均压缩率是1.52，这与最高压缩率比较接近。在图3(b)中展示了在不同子序列长度阈值下对空间路径的压缩比率。这里压缩率是指经过第二阶段压缩后得到的轨迹与第一阶段空间压缩后的轨迹之间的比率。

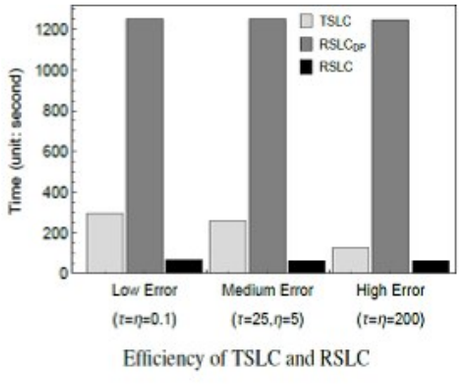

图4　时间压缩的效率

(下转第43页)

识库、数据库的技术上，使搜索更加的智能化，并通过对信息进行提取和分析，精准地实现了智能化搜索这一特点。

5 结论与展望

时代进步科技发展，搜索引擎技术也一步步的从最初分类目录导航进化到海量网页关联再更新到了细粒度的知识实体抽取，从第一代进化到第三代，搜索引擎技术变得愈来愈成熟也更加庞大，囊括的辅助性的知识内容也越来越多，简简单单的一行搜索框背后隐藏着极其复杂的机制。人们想要的是找准唯一的、正确的答案，所以搜索引擎与大数据分析技术相结合是未来必不可少的、更加智能的搜索技术。若想要了解清楚在这个特殊的技术背后的原理，我们仍需要在未来投入大量的工作并展开更加深入的研究，仅仅做几组实验通过数据浅显的证明是不够的。我期望这一先驱性的工作可以激励本领域更多的同行研究人员在此方向上开展更为相近深入的研究。让搜索引擎将在我们以后的生活中发挥更加出色的作用，将它与我们的生活变得更加的密不可分。

参考文献：

[1] 姜恩波，覃琳．基于结构化数据的搜索引擎[J]．现代情报，2019,39(2):66-72.
[2] 李海莹．百度公司搜索引擎技术的专利分析[J]．中国发明与专利,2019(4):99-106.
[3] 周永红，吴芳．大数据时代搜索引擎用户的信息安全问题研究[J]．图书馆,2017(5):32-35,57.
[4] 刘波．计算机搜索引擎智能化技术探析[J]．现代信息科技，2019(5):102-104.
[5] 龙佳．论搜索引擎的特点与发展态势[J]．电脑知识与技术，2019,15(1):200-201.
[6] 方师师．搜索引擎中的新闻呈现:从新闻等级到千人千搜[J]．新闻记者,2018(12):45-57.

【通联编辑：代影】

（上接第36页）

能提升信息化管理水平，不断加强信息化管理系统的完善、统一规划和管理，加强对校内各种新闻信息资源的整合和优化，提高高校信息利用率，避免各种信息的随意推送。另外，高校还需要注重对信息化管理人才的培养，聘请专业人士对信息化管理人员进行信息化素养和技能培训，让他们借助人工智能[4]技术高效、精准地进行数据分析、计算、做出更加科学的决策，促进相关管理工作的顺利开展。

参考文献：

[1] 王媛．企业人力资源管理信息化建设探讨[J]．大众标准化，2019(16):176-177.
[2] 高昱．医院信息化管理中智能信息处理技术的应用研究[J]．中国管理信息化,2017,20(5):157-158.
[3] 姜启源,谢金星,叶俊．数学模型[M].4版．北京:高等教育出版社,2011:249-260.
[4] 董云川,韦玲．人工智能促进高等教育发展的伦理纠偏[J]．重庆高教研究,2020(8):1-5.

【通联编辑：闻翔军】

（上接第40页）

贪婪RSLC和TSLC都是线性算法，但RSLC(DP)的时间复杂度为O(n2)。如图4所示，RSLC(DP)压缩包含108个时态元组的训练集比TSLC和RSLC贪婪实现花费的时间要长得多。与贪婪(greedy)实现的RSLC相比，TSLC更加耗时，因为当误差范围很小时，多边形包含更多的边。但是，多边形的边缘不会超过4n+4，因此可以保证TSLC的效率。另一方面，贪婪的RSLC总是运行得很快，因为它只需扫描时间序列中的n个点。

5 结论

本文设计了一种实用的轨迹分解方法，将空间数据与时间数据分开，实现了轨迹数据的降维处理。在此基础之上，提出了充分考虑时空特征的轨迹大数据压缩算法。最后，实验结果表明了算法在压缩比和压缩质量指标上的优势。

参考文献：

[1] 梁明,陈文静,段平,等．轨迹压缩的典型方法评价[J]．测绘通报,2019(4):60-64,70.
[2] 王伟谭松荣．基于轨迹大数据离线挖掘与在线实时监测的出租车异常轨迹检测算法[J]．数字技术与应用，2018(12):118-119.
[3] 张玺君,袁占亭,张红,等．交通轨迹大数据预处理方法研究[J]．计算机工程,2019,45(6):26-31.
[4] 冯慧芳,柏凤山,徐有基．基于轨迹大数据的城市交通感知和路网关键节点识别[J]．交通运输系统工程与信息,2018,18(3):42-47,54.

【通联编辑：张薇】