DOI :10.13203/j.whugis2004.01.011
第29卷 第1期 武汉大学学报·信息科学版 Vol. 29 No. 1
2004年1月 Geomatics and Information Science of Wuhan University Jan. 2004

文章编号: 1671-8860(2004)01-0052-03 文献标识码: A

时间序列中反向查询算法的研究

杜国明[1] 龚健雅[2] 朱家松[2]

(1 中山大学地理科学与规划学院, 广州市新港西路 135号, 510275)
(2 武汉大学测绘遥感信息工程国家重点实验室, 武汉市珞喻路 129号, 430079)

摘 要: 针对时间序列中的反向查询, 提出了一种新的索引方法——ES-索引, 介绍了ES-索引的建立、查询过程, 并对数据动态更新时的ES-索引作了详细的阐述, 简要说明了ES-索引在点、范围及近似查询中的应用。

关键词: 时间序列; 反向查询; ES-索引

中图法分类号: P208

时间序列是指按时间顺序排列的一系列观测数据(v_1, v_2, …, v_n), 它广泛地应用于科学研究、社会经济、工程建设等领域, 因而对它的研究非常重要。

由于构成时间序列的数据量通常很大, 造成查询所占用的时间较长, 所以时间序列数据的查询成为一个研究热点。时间序列的两种基本查询为前向查询和反向查询。前向查询是指查找在 t 时刻或时间区间[t', t'']内 v 的取值, 用 $Q(t)$ 表示在时间点 t 的值; 反向查询是指在时间序列中查找等于值 C 的时间点或在某一值域范围内的时间区间, 也叫值查询, 如在什么时刻 t 值等于 v', 在什么时间范围内 t 值大于或小于 v', 如图1。其中范围查询可转化为点查询。本文针对连续型时间的反向查询进行讨论, $Q^{-1}(v)$ 表示反向查询。

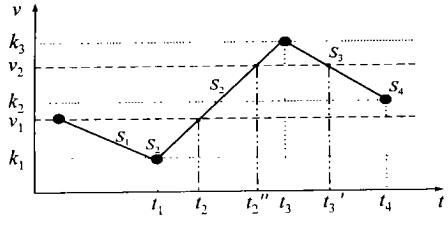

图1 时间序列示意图

Fig. 1 Illustration of Time Series

反向查询广泛应用于日常生活、生产及科研中。如在城市居民用水中, 在一段时间内, 对用水量超过某一规定值的用户实施变价收费; 在病人体温记录中, 什么时间体温超过38℃。传统的方法是在时间序列中对所有数据顺序扫描, 找到所需的数据。这种方法虽然简单, 但比较费时, 时间复杂度为 $O(n)$, 而且有时会得不到期望的结果。

为了解决上述问题, 笔者提出了ES-索引。

1 ES-索引建立及查询

对于时间序列 $T_s(s_1, s_2, \cdots, s_m)$, 其中 $s_i(t_i, v_i)$ 称为状态, 两个连续状态 s_i 和 s_{i+1} 之间的线段构成时间片断 S_i(如图1)。当 m 无限大时, 它的数值分布将稳定在一定区间[R_{min}, R_{max}]内。把区间[R_{min}, R_{max}]等分成 $n-1$ 份, 于是得到 n 个索引项($K_1, K_2, \cdots, K_n$)和 $n-1$ 个索引区间($I_1, I_2, \cdots, I_{n-1}$), 其中 $K_i = R_{min} + (i-1)(R_{max} - R_{min})/(n-1)$, 区间 $I_j = [K_j, K_{j+1})$。T_s 的每个 v_i 值对应于 j 个索引项, 其中, $j \in [1, n]$, 把属于区间 I_j 的所有时间片断 S_i 归为一组, 查询时只需找到查询值 v' 对应的区间 I_j 即可得到所有的时间片断。最后通过用户自定义内插方法计算出时间点, 这种索引方式称ES-索引。为方便起见, 用 K_j 表示区间[K_j, K_{j+1})。当数据量大时, 为减少索引项 K_j 所对应的时间片断, 可适当增大 n 值, 以减少查询时间。

1.1 索引的建立

对于任意时间片断 $S_i(i = 1, 2, \cdots, m-1)$ 上的一点 $s'(t', v')$, 如果 $\exists v'$: $(t', v') \in S_i \wedge v' \geqslant k_j$

收稿日期: 2003-07-20。
项目来源: 中山大学青年教师科研启动基金资助项目。

$\wedge v' < k_{j+1}$, 则索引项 K_j 的指针指向S_i, 用$F(K_i)$表示索引项 K_i 所指向的时间片断, 用$S_i(k_e)$表示时间片断 S_i 所对应的索引项, 那么$F(K_i) = F(K_i) + S_i$, $S_i(k_e) = S_i(k_e) + K_i$, 此处的"+"表示的意义为将该值加入到该集合, 例如$F(K_i) = < S_1, S_2 >$, $F(K_i) + S_3 = < S_1, S_2, S_3 >$。在计算机内部实现时, 为方便起见, 索引项指向的时间片断用下标表示, 如用 i 表示 S_i, $F(K_i) = < S_1, S_2, S_3 >$ 就表示为 $F(K_i) = < 1, 2, 3 >$, 以节省存储空间。

以图 1 为例, $R_{min} = v_2$, $R_{max} = v_3$, 将$[v_2, v_3]$等分成两份, 索引项分别为 K_1、K_2、K_3, $F(K_1) = < S_1, S_2 >$, $F(K_2) = < S_2, S_3 >$, $F(K_3) = < S_3 >$; $S_1(k_e) = < K_1 >$, $S_2(k_e) = < K_1, K_2 >$, $S_3(k_e) = < K_2, K_3 >$

对已有数据建立索引的算法如下。

1) 判断取值范围, 即最大值 R_{max} 与最小值 R_{min};

2) 设定索引项数 n;

3) 计算间距 L_s, $L_s = (R_{max} - R_{min})/(n-1)$;

4) 读取时间片断 S_i, 设其起、止点值分别为 v_i、v_{i+1}, 令 $v_s = \min(v_i, v_{i+1})$; $v_e = \max(v_i, v_{i+1})$;

5) 令 $h = [(v_s - R_{min})/L_s] + 1$, $j = [(v_e - R_{min})/L_s] + 1$(表示不大于该数的最大整数);

6) 初始化 $g = h$;

7) $F(K_g) = < F(K_g) > + i$; //将时间片断所在的序号加入到索引项所指向的队列中;

8) 如果 $g < j$, $g = g + 1$, 到步骤 7); 如果 $h * L_s ! = v_e \vee v_i > v_{i+1}$, 那么 $F(K_j) = < F(K_j) > + i$; 否则, 如果时间片断没读取结束, $i = i + 1$, 到步骤 4);

9) 结束。

特殊情况的考虑如下(见图 2)。

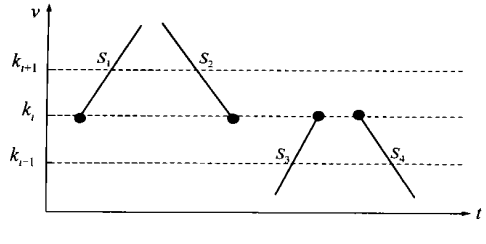

图 2　几种特殊情况的考虑示意图

Fig. 2　Illustration of Critical Cases

1) 当 $v_{min} = K_i$ 时, 设时间片断 S_1、S_2 的两个端点值 v_1、v_2, $v_{min} = \min(v_1, v_2)$。$v_{1min} = K_i$, $v_{2min} = K_i$, 这时 $S_1, S_2 \in F(K_i) \wedge S_1, S_2 \in F(K_{i+1}) \wedge S_1, S_2 \notin F(K_{i-1})$。

2) 当 $v_{max} = K_i$ 时, 设时间片断的起、止端点值分别为 v_1、v_2, $v_{max} = \max(v_1, v_2)$, 如图 2 中 S_3、S_4 所示, 分两种情况讨论。

1) $v_1 < v_2$ 时, 如图 2 中 S_3 所示, $S_3 \in F(K_{i-1}) \wedge S_3 \notin F(K_i)$。

2) $v_1 \geqslant v_2$ 时, 如图 2 中 S_4 所示, $S_4 \in F(K_{i-1}) \wedge S_4 \in F(K_i)$。

1.2　查询的实现

首先, 找到待查询值 v' 所在的索引项, 计算公式为 $j = [(n-1) \times (v' - v_{min})/(v_{max} - v_{min}) + 1]$, 索引项为 K_j。然后, 找到索引项 K_j 所指向的时间片断中符合条件的时间片断, 判断依据为: 对于 S_i, 有 $S_i \in F(K_j)$, S_i 的起、止两个端点在 v 轴上的投影分别为 v_i、v_{i+1}, 如果 $v_{i+1} \geqslant v_i$, 满足 $v_i \leqslant v' < v_{i+1}$的时间片断 S_i 即为所求; 如果 $v_i > v_{i+1}$, 满足 $v_{i+1} < v' \leqslant v_i$ 的时间片断S_i 即为所求。最后, 对时间点进行插值, 这里允许用户自定义插值, 本文采用线性内插方法, 公式为 $t = (t_{i+1} - t_i)(v' - v_i)/(v_{i+1} - v_i) + t_i$, 如果时间按等间隔排列, 即 $t_{i+1} - t_i$ 为常数, 对其单位化, 即 $t_{i+1} - t_i = 1$, 于是公式变为 $t = (v' - v_i)/(v_{i+1} - v_i) + t_i$。

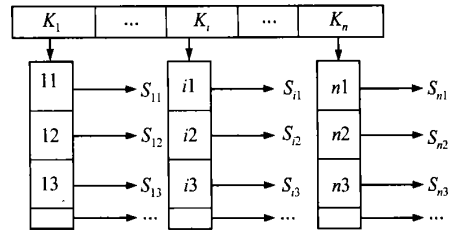

图 3　执行 ES-索引的结构示意图

Fig. 3　Structure of Implementing ES-Index

1.3　数据动态更新

本文讨论的动态更新是以追加方式进行的, 这种追加方式在现实中很常用, 如气象观测数据、用水量等只能添加数据, 而不能修改过去的数据。

上述提到的时间序列的 ES-索引是对已有数据进行的, 当 ES-索引建立好以后, 再进行数据追加时, 需要对索引项加以调整。假设已知时间序列 $T_s(s_1, s_2, s_3, \cdots, s_m)$, 其中, $s_i(t_i, v_i)$ $(i = 1, 2, \cdots, m)$已建立好 ES-索引, 其索引项序列为$(K_1, K_2, K_3, \cdots, K_n)$。当加入新值 $s_{m+1}(t_{m+1}, v_{m+1})$时, ① 如果 $R_{min} \leqslant s_{m+1} < R_{max}$, 按照前述建立索引步骤即可完成索引的建立, 索引项无需任何改变; ② 当 $R_{max} \leqslant v_{n+1} < R_{max} + (R_{max} - R_{min})/(n-1)$时, 也可应用前述索引建立方法, $F(K_n) = F(K_n) + m$, 索引项也无需改变; ③ 如果 $v_{m+1} \geqslant R_{max} + (R_{max} -$

$R_{min})/n$，令 $u=\left\lceil\frac{(n-1)\cdot(v_{m+1}-R_{max})}{R_{max}-R_{min}}\right\rceil+n$（$\lceil Q\rceil$表示不小于 Q 的最小整数），$K_u=\frac{R_{max}-R_{min}}{n-1}(u-1)+R_{min}$，$p=\left\lfloor\frac{(n-1)\cdot(v_m-R_{min})}{R_{max}-R_{min}}\right\rfloor+1$（$\lfloor Q\rfloor$表示不大于 Q 的最小整数）。于是，$F(K_p)=F(K_p)+m$，$F(K_{p+1})=F(K_{p+1})+m$，…，$F(K_{u-1})=F(K_{u-1})+m$，即$S_m(k_e)=<K_p,K_{p+1},\cdots,K_{u-1}>$，索引项新增 $K_{n+1},K_{n+2},\cdots,K_u$，此时的值域区间为$[R_{min},K_u]$。

可以看出，对方法②，当 $R_{max}\leqslant v_{n+1}<R_{max}+(R_{max}-R_{min})/(n-1)$时，方法③也适用，因此，当 $v_{n+1}\geqslant R_{max}$时即可采用方法③。

当 $v_{m+1}<R_{min}$时，在不改变以前已有索引的情况下，介绍一种双向索引的方法。

已知时间序列 $T_s(s_1,s_2,s_3,\cdots,s_m)$，加入数据 v_{i+1}，$v_{i+1}<R_{min}$时，$i+1>m$，在不改变已有的索引情况下，对小于 R_{min}的时间序列采用 Top-Down的方式建立索引，即索引项自顶向下逐渐增大。令索引项为 $L_1,L_2,L_3,\cdots$，则 $L_1<L_2<L_3<\cdots$，对大于 R_{min}的时间序列采用 Down-Top 的方式建立索引，即索引项自底向上逐渐增大。令索引项为 $K_1,K_2,K_3,\cdots$，则 $K_1<K_2<K_3<\cdots$。于是形成一条以过 R_{min}的直线为界的上下两区间$[R_{min},+\infty)$及$(-\infty,R_{min})$。当 $v_i\geqslant R_{min}$时，令 $p=\left\lfloor\frac{(n-1)\cdot(v_i-R_{min})}{R_{max}-R_{min}}\right\rfloor+1$，$u=\left\lceil\frac{(n-1)(R_{min}-v_{i+1})}{R_{max}-R_{min}}\right\rceil$，$L_u=R_{min}-u\frac{R_{max}-R_{min}}{n-1}$，可得 $F(K_1)=F(K_1)+i$；$F(K_p)=F(K_p)+i$；$F(L_1)=F(L_1)+i$；$F(L_u)=F(L_u)+i$。当 $v_i<R_{min}$时，令 $p=\left\lceil\frac{(n-1)(R_{min}-v_i)}{R_{max}-R_{min}}\right\rceil$，$u=\left\lceil\frac{(n-1)(R_{min}-v_{i+1})}{R_{max}-R_{min}}\right\rceil$，可得 $F(K_p)=F(K_p)+i$，$F(K_u)=F(K_u)+i$。

2 ES-索引在时间序列反向查询中的应用

应用 ES-索引计算各种形式的反向查询，这些查询包括点查询、范围查询和近似查询。

1）点查询

查询1：什么时候值等于 v_2？

这种查询可表示为 $Q^{-1}(v_2)$。从图1中可看到，没有一个存储的时间点等于 v_2，为了有效地处理这种查询，可以借助 ES-索引，直接通过公式 $i=\frac{(n-1)\cdot(v_2-R_{min})}{R_{max}-R_{min}}$（图1中 $n=2$）计算 v_2 所在的索引项为 K_2，K_2 指向时间片断 S_2、S_3，用公式 $(v_s-v_2)(v_2-v_e)\geqslant 0$ 判断是否符合 $Q^{-1}(v_2)$ 的查询条件，其中 v_s、v_e 分别为时间片断的起始端点在 v 轴上的投影值。S_2、S_3 都符合查询条件，通过与 S_2、S_3 相关联的时间 t_2、t_3，利用线性内插公式 $t=(v_2-v_s)/(v_e-v_s)+t_s$，计算可得 t_2''、t_3'。查询步骤如下。

① 输入查询值 C。

② 读取 T_s 时间序列。

③ 计算索引项。如果 $C\geqslant R_{min}$，则 $j=\left\lfloor\frac{(n-1)\cdot(C-R_{min})}{R_{max}-R_{min}}\right\rfloor+1$，索引项序列为 $K_1,\cdots,K_j,\cdots$；如果 $C<R_{min}$，则 $j=\left\lceil\frac{(n-1)\cdot(R_{min}-C)}{R_{max}-R_{min}}\right\rceil$，索引项序列为 $L_1,\cdots,L_j,\cdots$，for($j=0$；$i<t_{card}$；$i++$)，t_{card}表示索引项(K_j 或L_j)指针指向的时间片断数。如果满足$(S_{i+1}-C)(C-S_i)>=0$，通过 S_i、S_{i+1}进行线性插值计算，得 t_i'即为所求。

2）范围查询

范围查询返回一个时间区间序列，时间区间由 $Q^{-1}(C)$返回的时间点及开始时间 t_s 和终止时间 t_e 组成。因此，在范围查询中先执行点查询，得出相应的时间点，再由这些时间点及起始点共同构成时间区间序列即为所求。

判断时间 t 的增减性如下。如果 $S_i<C<S_{i+1}$，则在时间点 t 附近是局部递增的，方向符号为正，用 dire[t] =1 表示；如果 $S_i>C>S_{i+1}$，则在时间点 t 附近是局部递减的，方向符号为负，用 dire[t] =0 表示；其他令 dire[t] =−1。如果时间点 t 附近是递增的，则 $Q^{-1}(v>C)$；如果时间点 t 附近是递减的，则 $Q^{-1}(v<C)$。

3)近似查询

由于时间序列在数据采集时具有一定的误差，并且测量中还存在一些不确定性因素，因而在很多情况下并不需要查询某一确切值，而只是查询某一近似等于常数 C 的值。

3 结 语

本文提出的 ES-索引方法是为了有效地回答时间序列中的反向查询，它是在文献[1]的基础之上，并借鉴了文献[2]的某些思想。实验证明，该方法成功地应用于时间序列的反向查询中。当索引项 n 取值合适时，可减少空间存储量，同时达到提高查询速度的目的。 （下转第62页）

5 刘大杰, 史文中, 童小华, 等. GIS 空间数据的精度分析与质量控制. 上海: 上海科学技术文献出版社, 1999
6 汪孔政, 王解先. GIS 中曲线误差带模型的一种算法. 武汉测绘科技大学学报, 1999, 24(2): 142~144
7 易大义, 沈云宝, 李有法. 计算方法. 杭州: 浙江大学出版社, 1989
8 程正兴, 李水根. 数值逼近与常微分方程数值解. 上海: 交通大学出版社, 2000
9 武汉测绘科技大学测量平差教研室. 测量平差基础(第三版). 北京: 测绘出版社, 1996

第一作者简介: 王新洲, 教授, 博士生导师。主要从事测量数据处理理论与应用研究。代表成果: 非线性模型参数估计理论与应用; 模糊空间信息处理。出版专著和教材两本, 发表论文逾百篇。
E-mail: whwxz@163.com

ε_m-Band Based on Spline Fitting Function of Anomalous Curves in GIS

WANG Xinzhou[1] *TANG Zhong'an*[1] *CHEN Zhihui*[1]
(1 School of Geodesy and Geomatics, Wuhan University, 129 Luoyu Road, Wuhan 430079, China)

Abstract: This paper analyzes the numerical value modeling theory of anomalous curve ε_m-band based on the thrice spline function, puts forward an exactitude algorithm for the border enveloping of the anomalous curve ε_m-band, then analyzes the results of the ε_m-band.

Key words: GIS; anomalous curve; spline function; error-band; curve fitting; visualization

About the first author: WANG Xinzhou, professor, Ph. D supervisor, concentrated on the research and education in the theory and application of surveying data processing. His cardinal achievements include: the parameter estimation theory and its application of nonlinear model; fuzzy spatial information processing. He is the author of two books and over 100 papers.
E-mail: whwxz@163.com

(责任编辑: 平子)

(上接第 54 页)

参 考 文 献

1 Lin L. Indexing Values of Time Sequences. The 5th International Conference on Information and Knowledge Management, Maryland, 1996
2 Nanopoulos A. Indexing Time-Series Databases for Inverse Queries. International Conference on Database and Expert System Applications, Vienna, 1998

第一作者简介: 杜国明, 博士。现主要从事时态 GIS 等研究。
E-mail: gmdu@sina.com

Algorithm of Inverse Query on Time Series Data

DU Guoming[1] *GONG Jianya*[2] *ZHU Jiasong*[2]
(1 School of Geographical Science and Planning, Sun Yat-sen University, 135 West Xingang Road, Guangzhou 510275, China)
(2 State Key Laboratory for Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, 129 Luoyu Road, Wuhan 430079, China)

Abstract: This paper present a new method——EP-index which is used in time series data for inverse query. The procedures of establishing ES-index and querying are introduced. In addition, when data are appended, the dynamic procedure of establishing ES-index is expounded. Lastly, point interval and approximate search are introduced.

Key words: time series; inverse query; ES-index

About the first Author: DU Guoming, Ph. D, engaged in the research on temporal GIS.
E-mail: gmdu@sina.com

(责任编辑: 涓涓)