第32卷 第15期 计 算 机 工 程 2006年8月
Vol.32 № 15 Computer Engineering August 2006
·软件技术与数据库· 文章编号：1000—3428(2006)15—0061—03 文献标识码：A 中图分类号：TP311

IC-索引：一种支持时间序列反向查询的索引方法

曲吉林[1,2]，寇纪淞[1]，李敏强[1]
(1. 天津大学管理学院，天津 300072；2. 山东财政学院计算机系，济南 250014)

摘 要：为解决时间序列的反向查询问题，提出了一种新的时间序列动态索引方法——IC-索引。采用单调链表示时间序列的状态变化，利用AVL树对时间序列的变化区间进行索引。实验结果表明，算法的运行时间比现有的IP-索引减少了50%。
关键词：时间序列；反向查询；IC-索引

IC-Index: A New Index Structure for Inverse Queries on Time Sequence

QU Jilin[1,2], KOU Jisong[1], LI Minqiang[1]
(1. School of Management, Tianjin University, Tianjin 300072;
2. Department of Computer Science and Engineering, Shandong University of Finance, Jinan 250014)

【Abstract】 This paper presents a new dynamic indexing structure, the IC-index, for inverse queries on time sequences. Using the domain of time sequence monotone chains as the index entry and AVL-tree as the index structure, the IC-index dramatically improves the processing time of inverse queries compared to the IP-index, which is reduced to 50% according to the experimental results.
【Key words】 Time sequence; Inverse query; IC-Index

时间序列是随时间变化的一系列数据，在商业、金融、医疗、气象、工程技术和科学实验等领域都具有广泛的应用，例如气象分析、股市分析、用水量分析等。目前，时间序列的相似性搜索、趋势分析等问题得到了比较深入的研究，其反向查询技术近年来也已经引起了研究者的注意[1~4]。

时间序列的反向查询是指在时间序列中查找值等于 v'的时间点，或值在某一范围内的时间段，如某一地区什么时间气温超过37℃等。时间序列的反向查询具有重要的理论和实践价值。

反向查询最直接的方法是采用传统的顺序查找，这种方法虽然简单，但需要扫描整个时序数据库，同时由于时间序列状态是不断增加的，每次查询都需要重新对整个数据库进行扫描。针对这种问题，Lin提出了一种IP-索引方法[2]，用线段连接连续的两个状态，将时间序列的数值变化范围划分为区间，利用数组记录每个区间内的线段，同顺序查找相比在查询效率上有较大的提高[2,3]，但由于时间序列数据量大，区间数量和数组内线段的数量过多，影响了查询效率[3]；文献[3, 4]对IP-索引方法进行了改进，采用R*树表示数值变化区间，但这些改进对数据处理和查询的时间影响不明显[3]。

1 问题描述

1.1 时间序列的特点

时间序列可以描述为连续的一系列状态$S_i=(t_i,v_i)$的集合，其中t_i为时间，$v_i=F(t_i)$为时间序列值。其主要特点如下：

(1)顺序性。时间序列各个状态的时间是有序的，对任意的i,j，若$i>j$，则$t_i>t_j$。

(2)唯一性。对任意的t_i，$v_i=F(t_i)$是唯一的，但反之不成立。

(3)连续性。时间序列的状态是连续的。

(4)动态性。时间序列的状态是不断增加的。

1.2 时间序列的查询

时间序列的查询分为正向查询和反向查询两种方式[1]：正向查询是根据时间t查询时间序列的值v，如某一地区上午10:00的气温是多少，或某一地区2005年7月16日的最高气温是多少等；反向查询则是根据时间序列的值v查询时间t。主要有以下3种方式：

(1)点查询。时间t等于多少时，时间序列的值等于v'，即求$t'=F^{-1}(v')$。例如，在什么时间某一地区的气温达到37℃。

(2)范围查询。时间 t 在那些时间段内时，时间序列的值大于 v'，或小于 v'，或在 v'与 v''之间。例如，某一地区在哪些时间气温超过37℃。

(3)统计查询。时间序列何时达到最大值或最小值。例如，某一地区2005年的什么时间气温最高。

2 IC-索引方法

2.1 IC-索引的原理

时间序列的状态$S_i=(t_i,v_i)$可以看作二维t-v空间的一个点。对于任意一个点S_i，若其v值分别大于其相邻的两个点，即$v_i>v_{i-1}$且$v_i>v_{i+1}$，则S_i称为极大值点，如图1中的S_1, S_9；同理，若$v_i<v_{i-1}$且$v_i<v_{i+1}$，S_i称为极小值点，如图1中的 S_5, S_{13}。

如果用线段连接两个连续的状态S_i和S_{i+1}，则极值点将时间序列分割为许多单调链$[S_i, S_j](i>j)$，如图1，其中S_i为单调链的始点，S_j为单调链的尾点。对单调链$[S_i, S_j]$，若$v_i>v_j$，则$[S_i, S_j]$称为递增链，如图1中的$[S_5, S_9]$；否则$[S_i, S_j]$为递减链，

作者简介：曲吉林(1963—)，男，博士生、教授，主研方向：数据挖掘；寇纪淞、李敏强，教授、博导
收稿日期：2005-12-04 **E-mail：**qujl@sdfi.edu.cn

如图 1 中的[S_1, S_5]和[S_9, S_{13}]。

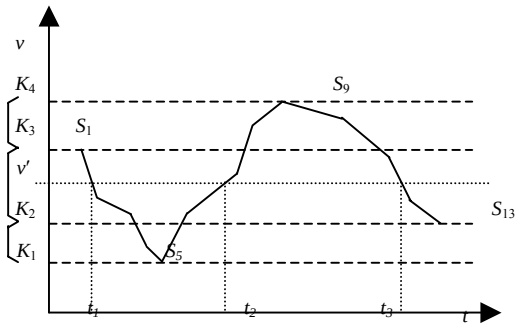

图 1 IC-索引的原理

如果将所有极值点向v轴投影，设投影点有M个，按其v值从小到大依次为$K_1,K_2,\ldots,K_M$，则这些极值点在v轴上的投影构成M-1 个连续、有序的区间$R_i=[K_i, K_{i+1})$，K_i为R_i的上界，K_{i+1}为R_i的下界，时间序列各个状态的v值分布就分布在这些区间内。

时间序列的一条单调链可能穿越一个区间或多个连续的区间，穿越区间R_i的单调链用anch(i)表示。如图 1 中有：

anch(1)={ [S_1, S_5], [S_5, S_9]}

anch(2)={ [S_1, S_5], [S_5, S_9], [S_9, S_{13}]}

anch(3)={ [S_5, S_9], [S_9, S_{13}]}

这样，对于一个查询$v= v'$，由于R_i区间是连续、有序的，因此首先利用折半查找找到v'所在的区间R_i，然后再求R_i内的单调链与直线$v= v'$的交点，就可以得到$v= v'$ 的时间点t。对于R_i内的一条单调链C_j，根据其单调性，采用折半查找就可以求出与C_j直线$v=v'$的交点。如图 1 中，$v= v'$的时间点为t_1, t_2, t_3。

2.2 IC-索引的插入

由于时间序列是动态的，随着时间的变化时间序列的状态也不断增加，因此 IC-索引必须支持单调链的插入。

不妨设新增的单调链$C_{insert}=[S_i, S_j]$为递增链，其始点和尾点分别为S_i 和S_j。根据时间序列的连续性，以S_i为尾点的单调链必是递减的，因此S_i是极小值点，如图 2。设S_i在v轴上的投影点的v值为K_i，K_i必是一个区间的下界，设这个区间为$R_i=[K_i, K_{i+1}]$。

根据S_j的v值v_j与区间R_i的关系，C_{insert}的插入分为以下 3 种情况：

(1)$v_j=K_{i+1}$。这时，S_j到达区间R_i的上界，如图 2(a)。将C_{insert}直接加入到R_i区间内，即

anch(i)= anch(i)+{C_{insert}}

(2)$v_j<K_{i+1}$。这时，S_j在区间R_i内，将R_i分为上、下两个区间，其中$R_i=[K_i, v_j)$，$R_i+1=[v_j, K_{i+1})$，如图 2(b)。R_{i+1}内的单调链为R_i内的单调链，R_i内的单调链加入C_{insert}，即：

anch(i+1)= anch(i)

anch(i)= anch(i)+{C_{insert}}

(3)$v_j>K_{i+1}$。这时，S_j超出R_i的上界。

1)若R_i之上没有其它区间，如图 2(c)。这时，需要在R_i之上增加一个新区间$R_{i+1}=[K_{i+1}, v_j)$，并将C_{insert}分别加入到R_i和R_{i+1}内，即：

anch(i)=anch(i)+{C_{insert}}

anch(i+1)={C_{insert}}

2)若R_i之上有区间R_{i+1}，如图 2(d)。这时，将C_{insert}加入到R_i后，再利用同样方法，根据v_j与区间R_{i+1}的关系，继续将C_{insert}插入到R_{i+1}内。

对于$C_{insert}=[S_i, S_j]$为递减链的情形，C_{insert}的插入方法与递增链原理类似，不再重述。

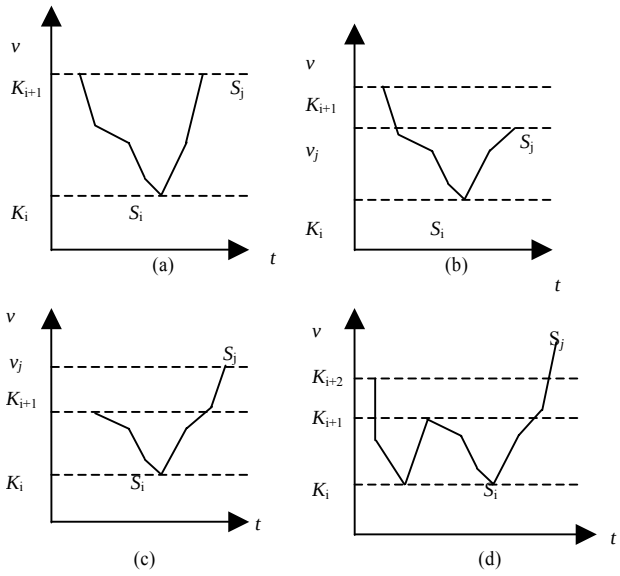

图 2 IC-索引的插入

2.3 IC-索引的数据结构

根据IC-索引的原理，IC-索引中主要有区间和单调链两种数据结构。如果将时间序列状态$S_i=(t_i,v_i)$存储在数组ts内，根据操作要求，IC-索引采用的主要数据结构如下：

(1)区间操作主要包括查找和动态插入。插入过程中可能引起节点的分裂，同时要保持较高的查找效率。支持这种操作的数据结构包括二叉树、B+树和 R 树等，由于 AVL 树是一种平衡的二叉树，支持折半查找，其节点插入、删除和查找时间均为 $logM$(M 为节点数)，因此可以采用 AVL 树存储时间序列的区间。AVL 树中每个节点的信息包括区间的上界和下界、子节点指针和指向该区间内单调链的指针。

(2)区间内的单调链的操作主要包括查找、插入。采用数组或链表结构都可以实现，其节点的主要信息包括单调链的始点、尾点在数组 ts 内的编号，以及单调链的增减标志，如 1 表示递增，-1 表示递减等。

(3)时间序列单调链的主要操作是求与水平线的交点。在一条单调链内，由于v_i与t_i一一对应，采用折半查找就可以求出其与水平线$v=v'$的交点。

根据上述讨论，以图 1 为例，IC-索引采用的主要数据结构如图 3 所示。

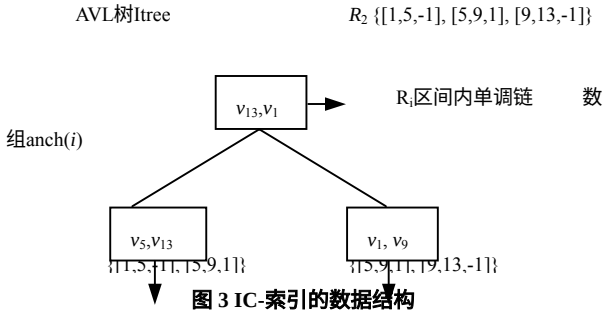

图 3 IC-索引的数据结构

3 反向查询的实现

利用 IC-索引可以实现时间序列的各种反向查询，下面介绍几种主要查询的实现方法：

(1)点查询

查询时间 t 等于多少时，时间序列的值等于 v'，例如在什么时间某一地区的气温达到 37℃。算法步骤如下：

步骤 1 利用折半查找在AVL树Itree内找到包含v'的区间R_i；

步骤 2 求R_i单调链数组anch(i)内单调链与直线$v=v'$的交点。对于anch(i)内的一条单调链C_j，利用折半查找求出与C_j直线$v=v'$ 的交点，得到时间t。

(2)范围查询

查询时间 t 在哪些时间段内时，时间序列的值 v 大于 v'，或小于 v'，或在 v'与 v''之间。由于整个时间序列是连续的链，只要求出时间序列与直线 $v=v'$的交点，就可以输出满足条件的时间段。下面给出查询 $v>v'$ 时步骤，其它查询类似。

步骤 1 利用点查询求时间序列与直线$v=v'$的交点，求得时间点t为$t'_1, t'_2, \ldots, t'_k$；

步骤 2 当k为偶数时。若$t= t'_1$时时间序列递增，则$v>v'$的时间段为$[t'_1, t'_2], [t'_3, t'_4], \ldots, [t'_{k-1}, t'_k]$；若$t= t'_1$时时间序列递减，$v>v'$的时间段为$[t'_{min}, t'_1]$ $[t'_2, t'_3], [t'_4, t'_5], \ldots, [t'_{k-2}, t'_{k-1}]$, $[t'_k, t'_{max}]$，其中t'_{min}为时间序列起始状态的时间，t'_{max}为最后状态的时间。当k为奇数时方法类似。

(3)统计查询

统计查询主要包括 t 在何时时间序列达到最大值或最小值等，例如某一地区 2005 年什么时间的气温最高。下面以求时间序列达到最大值的时间 t 为例，说明算法的实现步骤，其它查询步骤类似。

步骤 1 在AVL树Itree内找到v值最大的区间R_M，R_M的上界v_{max}就是时间序列的最大值；

步骤 2 利用点查询的方法，求R_M内的单调链与直线$v=v_{max}$的交点，这些交点的t值即为所求。

4 实验结果

时间序列的各种反向查询一般都需要通过点查询实现，即求时间序列与水平线 $v= v'$的交点。为测试算法的效率，采用 C++语言编程，对点查询算法进行了测试，其中时间序列的状态点采用随机函数自动生成。实验环境为 PIII CPU 1 000MHz，512MB 内存，实验结果如图 4 所示，其中时间为IC-索引建立和点查询的总时间。

从实验结果看出，当时间序列状态点达到 2 000 时，IC-索引查询的时间为 0.39s，算法效率比较高。同文献[2]的 IP-索引相比，时间减少 50%。实际应用中，时间序列通常是渐变的，IC-索引的优势将更加明显。

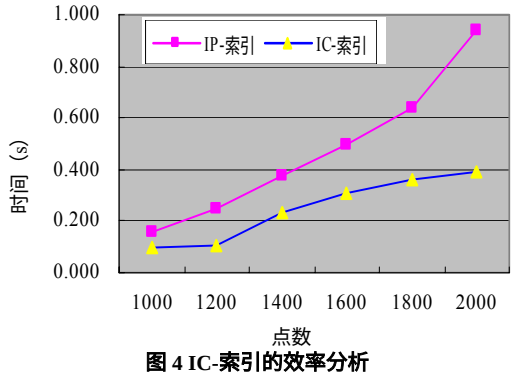

图 4 IC-索引的效率分析

5 结束语

本文提出了一种新的时间序列IC-索引方法，充分利用了时间序列连续性的特点，采用单调链表示时间序列的状态变化，数据处理时间比目前最好的IP-索引减少了 50%，算法的效率比较高，而且各种查询更加易于实现。实际应用中，对于时间序列中连续的两个状态S_i和S_{i+1}，可以用线段连接，也可以采用其它单调插值函数。

对于时间序列的反向查询还有许多问题需要进一步研究。例如：(1)高维时间序列反向查询问题，时间序列的状态点是高维的，例如分析空间中物体的位置随着时间变化等；(2)反向查询语言的实现问题，设计一种类似 SQL 的数据库查询语言，对时间序列进行反向查询；(3)在采用二级存储的情况下，IC-索引的优化问题。

参考文献

1 Hetland M L. A Survey of Recent Methods for Efficient Retrieval of Similar Time Sequences[M]. Data Mining in Time Series Databases. World Scientific Publisher, 2004.

2 Lin Ling, Tore R, Martin S. Indexing Values of Time Sequences[C]. Proc. of the 5th International Conference on Information and Knowledge Management, 1996.

3 Nanopoulos A. Indexing Time-series Databases for Inverse Queries[C]. International Conference on Database and Expert System Applications, Austria, 1998.

4 杜国明, 龚健雅, 陈晓翔. IP-索引在时间序列反向查询中的应用[J]. 计算机工程, 2003, 29 (7): 7-8, 58.

（上接第 60 页）

参考文献

1 Yang J, Malek K A. Approximate Swept Volumes of NURBS Surfaces or Solids[J]. Computer Aided Geometric Design, 2005, 22(1): 1-26.

2 Malek K A, Yeh H J. Geometric Representation of the Swept Volume Using Jacobian Rank-deficiency Conditions[J]. Computer Aided Design, 1997, 29(6): 457-468.

3 Blackmore D, Leu M C, Wang L P. Sweep-envelope Differential Equation Algorithm and Its Application to NC Machining Verification[J]. Computer Aided Design, 1997, 29(9): 629-637.

4 Weld J, Len M. Geometric Representation of Swept Volume with Application to Polyhedral Objects[J]. International Journal of Robotics Research, 1990, 9(5): 105-117.

5 Malek K A, Yang J, Blackmore D. On Swept Volume Formulations: Implicit Surfaces[J]. Computer Aided Design, 2001, 33(1): 113-121.

6 Abrams S, Allen P. Swept Volumes and Their Use in Viewpoint Computation in Robot Work Cells[C]. Proceedings of the International Symposium on Assembly and Task Planning, 1995: 188-193.

7 汪国平, 孙家广, 吴学礼. SWEEP 曲面的 NURBS 逼近[J]. 计算机学报, 1998, 21(9): 44-849.

8 Xia J, Ge J. Kinematic Approximation of Ruled Surfaces Using NURBS Motions of A Cylindrical Cutter[C]. Proceedings of the ASME Design Automation Conference, Baltimore, MD, 2000.

9 Litke N, Levin A, Schroder P. Trimming for Subdivision Surfaces[J]. Computer Aided Geometric Design, 2001, 18(5): 463-481.