第32卷 第22期 计 算 机 工 程 2006年11月
Vol.32 No.22 Computer Engineering November 2006
·博士论文· 文章编号：1000—3428(2006)22—0004—03 文献标识码：A 中图分类号：TP391

基于聚类的区间数时间序列的索引方法

翁小清[1,2]，沈钧毅[1]
(1. 西安交通大学软件所，西安 710049；2. 河北经贸大学计算机中心，石家庄 050061)

摘 要：在时间序列数据库中，大多数现有的相似性搜索方法都集中在如何提高算法的效率，而对于由不精确数据组成的时间序列如何进行相似性搜索，则研究比较少，不精确数据经常用区间数据来表示；通过识别区间数时间序列中的重要区间数，使得区间数时间序列的维数大幅度降低，该文针对由区间数组成的时间序列，提出了一种基于低分率聚类的索引方法。实验表明，该方法加快了区间数时间序列的查找过程，不会出现漏报现象。

关键词：区间数时间序列；相似性搜索；聚类；索引

Time Series of Intervals Index Based on Clustering

WENG Xiaoqing[1,2], SHEN Junyi[1]
(1. Institute of Computer Software, Xi'an Jiaotong University, Xi'an 710049;
2. Computer Center, Hebei University of Economics and Trade, Shijiazhuang 050061)

【Abstract】Most existing approaches of similarity search in time series databases focus on the efficiency of algorithms but seldom provide a means to handle imprecise data. The imprecise data are normally presented in the interval. By identifying the important interval values from the time series of intervals, the dimensionality of the time series of intervals can be greatly reduced. This paper proposes an indexing approach of time series of intervals, based on clustering the time series of intervals in low resolution. As demonstrated by the experiments, the proposed approach speeds up the time series of intervals query process while it also guarantees no false dismissals.

【Key words】 Time series of intervals; Similarity search; Clustering; Index

时间序列数据库是数据序列的集合，序列中的元素按照时间先后的顺序排列；对给定的需要查询的时间序列，在时间序列数据库中查找与之变化模式相似的时间序列的过程，称为相似性搜索(Similarity search)。大多数现有的相似性搜索方法都集中在如何提高算法的效率，但是对于由不精确(imprecise data)、不确定(uncertainty data)数据组成的时间序列，如何建立索引，如何进行相似性搜索，则研究比较少。然而在现实世界中，不精确、不确定、不完善的数据到处存在，如我们无法用精确的数据来描述某天的气温情况，只能用区间数据来描述某天气温的变化范围在[6°C,15°C]之间，又如我们无法用精确的数据来描述IBM公司某天的股票价格，只能用区间数据来描述该公司某天的股价在[15$,22$]之间波动。文献[1]给出了两个区间数时间序列之间的相似度定义，然而由于计算较繁杂，该文没有给出区间数时间序列的索引方法。

本文针对区间数时间序列数据库，提出了一种基于低分辨率聚类的索引方法，对“最优”聚类个数K的值进行了估计，用多元方差分析对聚类的效果进行了检验。

1 相关定义

定义1 区间数时间序列[1]:

区间数时间序列X为有限个区间数所组成的序列，$X=(x_1,x_2,...,x_n)$，其中 $x_i=[\underline{x}_i, \overline{x}_i]$ 是一个区间数。

令 $\underline{X}=(\underline{x}_1,\underline{x}_2,...,\underline{x}_n)$ 是区间数时间序 X 的下限序列，$\overline{X}=(\overline{x}_1,\overline{x}_2,...,\overline{x}_n)$是区间数序列 X 的上限序列，则区间数时间序列 X 也可以表示为 $(\underline{X},\overline{X})$，区间数时间序列在下文中简称为区间数序列。

例1 荷兰阿姆斯特丹某年前6个月份气温变化情况组成一个区间数时间序列[2]：

X=([-4，4]，[-5，3]，[2，12]，[5，15]，[7，17]，[10，20])

例2 IBM公司某周5个交易日股票价格波动情况组成一个区间数序列[5]：

[92,93.09],[92.14,93.19],[92.47,93.38],[92.03,93.21],[91,92.35],)

定义2 设X和Y是两个长度都为n的区间数时间序列，X与Y之间的距离d(X,Y)为[2]

$d(X,Y)=\max(|\overline{x}_i-\overline{y}_i|,|\underline{x}_i-\underline{y}_i|,1\le i\le n)$

Rangarajan等[2]证明了d(X,Y)满足作为距离度量的3个公理(非负、对称和三角不等式)。

例3 阿姆斯特丹以及苏黎世某年一季度气温变化情况分别为

X=([-4，4]，[-5，3]，[2，12])，Y=([-11,9],[-8,15],[-7,18])

按定义容易计算出这两个区间数序列之间的距离为

d(X,Y)=12

2 重要区间数的识别

2.1 识别时间序列中的重要点

这里所说的重要点是时间序列的一些极值点；从时间序列中提取重要点的方法如下[3]：

图1给出了从时间序列T=(0.1,0.4,0.3,0.7,0.9,0.6,0.7,0.4,0.3,0.5)识别出前5个重要点的操作过程。

基金项目：国家自然科学基金资助项目(60173058)
作者简介：翁小清(1965－)，男，博士生、副教授，主研方向：数据挖掘；沈钧毅，教授、博导
收稿日期：2005-12-08 **E-mail：**xqweng@stu.xjtu.edu.cn

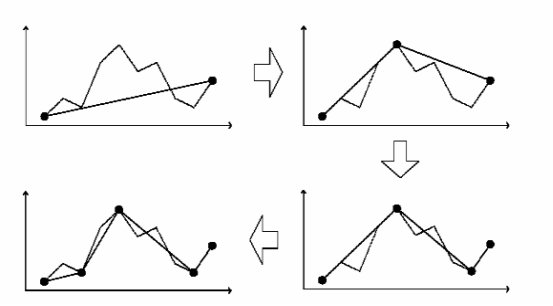

图 1 从时间序列中识别前 5 个重要点的过程

时间序列 T 的第 1 个数据 0.1 和最后一个数据 0.5 分别是 T 的第 1 个和第 2 个重要点，T 的第 3 个重要点是到第 1 个重要点与第 2 个重要点之间连线距离(纵向距离)最远的点，通过计算是时间序列 T 中的第 5 个数据 0.9，第 4 个重要点是到两个相邻重要点连线距离最远的点，第 4 个重要点可能在第 1 个重要点与第 2 个重要点之间，也可能在第 2 个重要点与第 3 个重要点之间，通过计算第 4 个重要点是 T 的第 9 个值 0.3，T 的其它重要点可以类似求出。

2.2 从区间数序列中识别重要区间数

对区间数时间序列的上限序列与下限序列求平均值，计算出该区间数时间序列的中值序列，采用 2.1 节所述方法从中值序列中识别其重要点，在区间数时间序列中相应位置的区间数，就为重要区间数。

例 4 IBM 公司 10 个交易日股票价格的波动情况组成了一个区间数序列

X=[92,93.09],[92.14,93.19],[92.47,93.38],[92.03,93.21],[91,92.35],[91.21,92.14],[89.01,91.51],[89.75,90.46],[93.55,95.65],[94.71,95.35])

对每个区间数的上限和下限取平均值就得到了的中值序列

X_{middle}=(92.54，92.66，92.92，92.62，91.67，91.67，90.26，90.10，94.60，95.03)

采用 2.1 节方法从时间序列 X_{middle} 中识别出它的前 5 个重要点，在 X_{middle} 中的位置按照重要性排列依次为 1，10，8，9，4。在区间数序列 X 中相应位置上的区间数即为 X 的重要区间数。将这 5 个重要区间数按照它们在 X 中时间的先后次序(即 1,4,8,9,10)排列，组成一个长度为 5 的重要区间数序列

([92,93.09], [92.03,93.21], [89.75,90.46], [93.55,95.65], [94.71,95.35])

该重要区间数序列既保留了原区间数序列 X 的主要特征，又对其进行了压缩，数据压缩比为 10/5 = 2。

3 区间数序列索引的建立与查询

3.1 区间数序列的维数约减

对于长度为 m 的区间数序列 P，采用 2.2 节所述方法，抽取出前 n 个重要区间数(n << m)，再将这 n 个重要区间数按照它们在 P 中时间的先后次序排列，组成一个长度为 n 的重要区间数序列，由于 n 远远小于 m，因此由这 n 个重要区间数组成的序列，既保留了原区间数序列 P 的主要特征，又对其进行了大幅度的压缩，压缩比例为 m/n。

3.2 索引的建立

对数据库中每一个区间数序列，都抽取出它的前 n 个重要区间数，将这前 n 个重要区间数再按照其在原区间数序列中的时间的先后次序组成重要区间数序列，将这些重要区间数序列分成若干个类(组)，每一个类中的区间数序列都具有相似的主要特征，用这些类(组)作为区间数序列数据库的索引。采用最常用的 K-均值聚类方法，对重要区间数序列进行聚类，将它们分为 K 类。

3.3 索引的更新

对 3.2 节建立的索引很容易进行更新，当数据库中新增加一个区间数序列时，首先从这个区间数序列中抽取前 n 个重要区间数，这 n 个重要区间数(按照其在原区间数序列中时间的先后次序)组成一个长度为 n 的重要区间数序列，计算该重要区间数序列到 K 个类中心的距离，将其分配到与之距离最短的那个类中心所在的类。

3.4 索引查询

用户输入需要查询的区间数序列 Q，算法从 Q 中抽取前 n 个重要区间数并组成重要区间数序列，然后计算该重要区间数序列到 K 个类中心的距离，选取与之距离最短的类；然后，计算这个类中每一个区间数序列与 Q 的前 n 个重要区间数序列的距离，与之距离最短的那个区间数序列，即为查询结果，查询结果具有与 Q 相似的主要特征。

3.5 “最优”聚类个数K的估计

估计“最优”聚类个数K值的方法[4]如下：让K的取值从 1 增加到N(N为数据库中含有区间数序列的个数)，重复执行K-均值聚类，观察J(K)的变化情况来估计“最优”的聚类个数K。J(K)定义为

$$J(K)=\sum_{i=1}^{K}\sum_{X\in M_i} d_{c_i,X} \tag{1}$$

其中：M_i表示第i类，C_i表示M_i类的类中心，$i=1,2,...,K$，X为区间数序列，用$d_{c_i,X}$表示区间数序列X与第i类的类中心C_i的距离。

一般情况下，J(K)的取值随着聚类个数 K 值的增加而减小，dJ(K)描述的是 J(K)变化的情况。

$$dJ(K)=\frac{|J(K+1)-J(K)|}{J(K)}\times 100\% \tag{2}$$

$$\psi(K)=Sign[dJ(K+1)-dJ(K)] \tag{3}$$

K=1,2,…,N

第 1 个使符号函数$\psi(K)$由负变为正的 K 值即为“最优”K 值，如果$\psi(K)$的符号一直不变化，选取使得$dJ(K)$非常接近 0 的 K 值。

4 实验

选取标准普尔 500 指数[5]中 266 家公司每天股票交易的最高和最低价格组成的区间数序列作为测试数据集，区间数序列长度在 37~253 之间。从这 266 家公司中随机抽取 10 家公司的区间数序列作为查询序列，以下显示的实验结果均为这 10 个区间数序列实验结果的平均值。用Matlab6.1 编写了所有的程序，并在清华同方笔记本(CPU 1.5GHz，内存 112MB，硬盘 40GB，Windows XP操作系统)上实现。

4.1 建立索引所花费的 CPU 时间

我们让重要区间数的个数从 5~30 之间变化，聚类的个数从 2~10 之间变化，测试了 K-均值聚类的叠代次数以及所花费的 CPU 时间，叠代次数都没有超过 50 次，说明在建立索引时，每一个重要区间数序列都被分配到了一个确定的类中，从而保证了在索引查询过程中，不会产生漏报(false dismissals)现象。

由于篇幅限制，以下只给出重要区间数的个数 n 为 10 或 5 的实验结果。图 2 给出了重要点个数固定为 10 的情况，当聚类的个数增加时，建立索引所花费的 CPU 时间也逐渐增加。

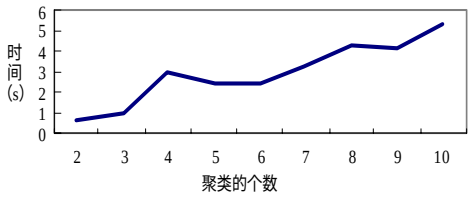

图 2 聚类个数与建立索引花费的 CPU 时间

4.2 索引查询剪切率(pruning power)测试

剪切率[3]的计算公式如下：

$$P=\frac{\text{索引查询中需检查的区间数序列的个数}}{\text{数据库中存放的区间数序列的总个数}}$$

从图 3 可以看到，随着聚类个数的增加，剪切率快速下降，从而索引查询时间也快速下降，说明剪切率受聚类个数的影响很大。

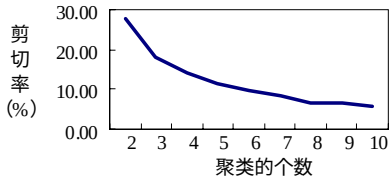

图 3 聚类个数与剪切率

4.3 索引查询与顺序查询之间的比较

重要区间数的个数 n 固定为 10，聚类个数为 5，从图 4 中可以看出，本文提出的索引查询方法远远好于顺序查询，当数据库中区间数序列个数增加时，索引查询的时间也能保持稳定。

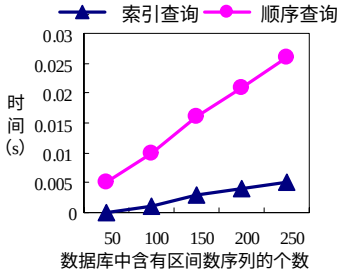

图 4 索引查询与顺序查询的比较

从以上实验可以看出，对于区间数序列数据库，虽然建立索引要花费一些时间，但只需建一次索引即可，而索引查询能够大幅度减少查询时间，索引更新也比较容易，在索引查询过程中也不会出现漏报现象。

4.4 “ 最优 ”聚类个数 K 的选择

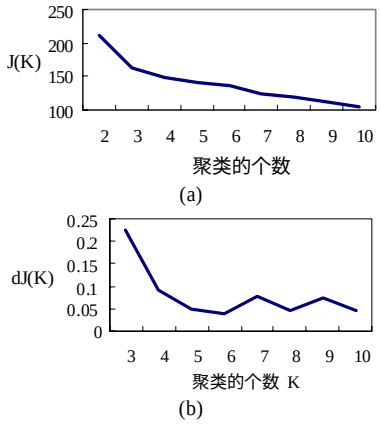

图 5 J(K)与 dJ(K)随聚类个数的变化情况

取重要区间数个数为 5，聚类的个数 K 从 2 变化到 10，从图 5 中可以看到J(K)随着聚类个数的增加而逐渐减小；当聚类的个数K大于 5 以后，dJ(K)的取值趋于稳定；通过计算得知第 1 个使符号函数$\psi(K)$由负变为正的K值是 7，所以当取 5 个重要区间数，进行K-均值聚类建立索引时，“ 最优 ”的聚类个数是 7 个，这 7 个类，类与类之间的差别是否显著，如何去检验？通常采用类的内聚性、类与类之间的相异性[2]来评价聚类的效果或聚类的质量，本文采用多元方差分析[5]来评价聚类的效果。多元方差分析计算的统计量为

$$\Lambda=\frac{|E|}{|E+B|}$$

其中 B 是组间离差矩阵，E 是组内离差矩阵，统计量 Λ 服从 Wilks 分布，记作

$$\Lambda \sim \Lambda_{p,N-k,k-1}$$

其中 p 是变量的个数，N 是总样本含量，k 是类(组)的个数。对于取重要区间数为 5 个，将 266 个重要区间数序列分为 7 个类，建立索引，即 p=5, N=266, k=7，所以统计量 $\Lambda\sim\Lambda_{5,259,6}$，对重要区间数序列的上限(下限亦可)进行多元方差分析：

$$\Lambda=\frac{|E|}{|E+B|}=\frac{7.8994\times10^{6}}{4.1618\times10^{8}}=0.019$$

界值$\Lambda_{0.01}(5,259,6)\approx0.808\ 906$，Λ统计量值为 0.019 小于界值 0.808 906，所以显著性水平(概率 P 值)小于 0.01，说明这 7 个类，类与类之间差异显著。

5 结论

针对区间数序列数据库，提出了一种基于低分辨率聚类的索引方法，对“ 最优 ”聚类个数 K 的值进行了估计，用多元方差分析对聚类的效果进行了检验。实验表明，本文提出的方法是有效的，当选取的重要区间数的个数比较少时，K-均值聚类经过有限的叠代次数就能收敛，从而在索引查询时不会产生漏报现象。本文的方法对于金融证券领域的区间数序列数据库比较有效，对于不同长度的区间数序列也能进行比较。

如何用离散傅里叶变换(DFT)、离散小波变换(DWT)、分段多项式(PPR)等表示区间数序列，并建立索引，值得今后作进一步的研究。

参考文献

1 Liao S S, Tang T H, Liu W Y. Finding Relevant Sequences in Time Series Containing Crisp, Interval, and Fuzzy Interval Data[J]. IEEE Transactions on Systems, Man, and Cybernetics — — Part B: Cybernetics, 2004, 34(5): 2071-2079.

2 Rangarajan L, Nagabhushan P. Dimensionality Reduction of Multidimensional Temporal Data Through Regression[J]. Pattern Recognition Letters, 2004, 25(8): 899-910.

3 Fu T C, Chung F L, Luk R, et al. Financial Time Series Indexing Based on Low Resolution Clustering[C]. Proceedings of Temporal Data Mining: Algorithms, Theory and Applications Held in Conjunction with ICDM’04, 2004: 1-10.

4 Singhal A, Seborg D E. Clustering of Multivariate Time-series Data[C]. Proceedings of the American Control Conference, Anchorage, AK, USA, 2002: 3931-3936.

5 张尧庭，方开泰. 多元统计分析引论[M]. 北京：科学出版社，1982.