第33卷 第17期 计 算 机 工 程 2007年9月
Vol.33 No.17 Computer Engineering September 2007
·软件技术与数据库· 文章编号：1000—3428(2007)17—0102—03 文献标识码：A 中图分类号：TP311

基于序变换的时间序列快速匹配搜索方法

袁晓峰[1]，许化龙[1]，陈淑红[2]
(1. 第二炮兵工程学院3系，西安 710025；2. 第二炮兵装备研究院第三研究所，北京 100085)

摘 要：分析了时间序列相似匹配搜索的研究现状，提出了基于序变换的时间序列相似匹配搜索方法。该方法能够实现趋势相似的不同长度子序列的快速匹配搜索，同时具有对匹配序列的平移、时间轴的伸缩不敏感、误警率低，以及易于建立索引等优点。实验结果证明了该方法的有效性。

关键词：时间序列；序模式；序变换；相似性搜索

Ordinal-transformation-based Method for Fast Similarity Search of Time Series

YUAN Xiao-feng[1], XU Hua-long[1], CHEN Shu-hong[2]
(1. No. 3 Dept., The Second Artillery Engineering Institute, Xi'an 710025;
2. No.3 Institute, The Second Artillery Armament Academy, Beijing 100085)

【Abstract】 This paper reviews the current research status of time series similarity search, and proposes ordinal-transformation-based similarity search method, which can achieve fast trend similarity search among sub-series of different length. In addition to insensitivity to horizontal shifting and time-axis scaling, the proposed method has a lower false alarm ratio and a higher indexing efficiency. Experimental results show the proposed method is quite competitive in terms of speed and robustness.

【Key words】 time series; ordinal pattern; ordinal transformation; similarity search

时间序列作为一种数据形式广泛存在于各种商业、医学、工程、自然科学和社会科学等数据库中。时间序列的搜索是整个时间序列数据挖掘的基础，是数据挖掘的一个重要研究方向。该领域的基础性工作最早是由Agrawal等在1993年提出的。该问题可描述为给定某个的时间序列，要求从一个大型时间序列数据库中找出与之最相似的序列。

近年来时间序列的相似性搜索问题正得到越来越多的重视，出现了许多面向相似性搜索的时间序列近似表示方法，如Agrawal采用的离散傅立叶变换DFT[1]，Chan等人提出的基于小波变换的方法[2]，Last等人提出的关键特征(如斜率和信噪比)法，Korn等人[3]提出的奇异值分解法SVD，Keogh等人[4]先后提出的分段累积近似法PAA、分段线性表示PLR和适应性分段常数近似法APCA[5]，Perng等人提出的界标模型[6]等，这些表示方法各有所长，针对了不同的应用背景，但仍有许多问题有待进一步的研究。例如：相似性度量进一步认识问题，降低对偏移、噪声的敏感性问题，算法的效率问题，用户与系统的交互性问题等。

时间序列的相似性搜索可分为整体匹配和子序列匹配2种。本文针对子序列匹配问题提出的基于序变换的相似匹配搜索方法，以时间序列特征点的序模式对相似性进行描述，并在此基础上以分段趋势相似进行相似度量，从而实现相似性匹配搜索的准确性和高效性。

1 基于序模式的时间序列相似匹配

1.1 序模式与序变换

序模式是对时间序列中距离均匀分布的时间序列值排序关系的描述[7]。

对于时间序列 $x(t)$，阶次 $d\in\mathrm{N}$，且延时 $\tau\in\mathrm{N}$，则在 t 时刻获得的唯一的排序为

$$\pi_d^\tau \equiv \begin{pmatrix} 0 & 1 & 2 & \dots & d \\ r_0 & r_1 & r_2 & \dots & r_d \end{pmatrix} \equiv (r_0, r_1, r_2, .., r_d) \tag{1}$$

其中，$\{r_0, r_1, \dots, r_d\}$ 满足：

(1) $x_{t+r_0\tau} \le x_{t+r_1\tau} \le \dots \le x_{t+r_{d-1}\tau} \le x_{t+r_d\tau}$；

(2)如果 $x_{t+r_{l-1}\tau} = x_{t+r_l\tau}$，取 $r_{l-1} \le r_l$。

对于式(1)中的排序，当 l=1, 2, …, d 时，其对应的序模式 P 的第 l 个元素表示如下：

$$i_l = i(\pi_d^\tau) = \#\{r \mid r \in \{0,1,\dots,l-1\}, \pi_d^\tau(r) < \pi_d^\tau(l)\} \tag{2}$$

其中，$\pi^\tau_d(r)$表示数值r对应的序值；运算符“#”表示取集合中满足不等式约束条件的元素r的个数。

如图1所示，参数设置为 t=1，d=4，τ=3，得出 $x(t+3\tau)<x(t+\tau)<x(t+4\tau)<x(t+2\tau)<x(t+0\tau)$，因此有

$$\pi_d^\tau \equiv \begin{pmatrix} 0 & 1 & 2 & 3 & 4 \\ 3 & 1 & 4 & 2 & 0 \end{pmatrix} \equiv (3,1,4,2,0) \tag{3}$$

π^τ_d=(3,1,4,2,0)对应的序模式P可通过如下分析求得：对于l=1，$\pi^\tau_d(0)$对应位置为4，而$\pi^\tau_d(1)$对应的位置为1，$\pi^\tau_d(0)$=4＞$\pi^\tau_d(1)$=1 不满足(2)中的条件，有i_1=0；对于l=2，有$\pi^\tau_d(0)$=4>$\pi^\tau_d(2)$=3，$\pi^\tau_d(1)$=1＜$\pi^\tau_d(2)$=3，r=1，一个元素满足条件，因此，i_2=1……依次类推，可得出对于π^τ_d，其序模式为

作者简介：袁晓峰(1975－)，男，工程师、博士研究生，主研方向：数据挖掘，故障诊断；许化龙，教授、博士生导师；陈淑红，工程师、硕士

收稿日期：2006-10-16 **E-mail：**epyxf@yahoo.com.cn

P={0,1,0,2}。

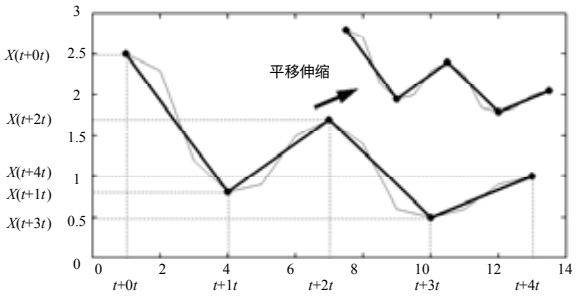

图 1　时间序列序模式示意图

在实际的序模式计算中，通常采用由式(1)和式(2)导出的式(4)进行计算：

$$i_l = i(\pi_d^\tau) = \#\{r \mid r \in \{0,1,...,l-1\}, x_{t+l\tau} \geq x_{t+r\tau}\} \tag{4}$$

序变换是在序模式定义的基础上，将给定的时间序列变换为处于[0, (d+1)!-1]区间的符号序列。

序模式$\{i_1,i_2,\ldots,i_3\}$与n_d之间形成的双射关系，即$\{0,1\}\times\{0,1,2\}\times\ldots\times\{0,1,\ldots,d\}$取值空间的元素排列与集合$\{0,1,\ldots,(d+1)!-1\}$中的整数一一对应，且该对应关系可通过式(5)描述[7]：

$$n_d(P) = \sum_{l=1}^{d} i_l \frac{(d+1)!}{(l+1)!} \tag{5}$$

对于前面的例子中序模式P={0,1,0,2}，按照式(5)可得$n_d(P)$=22。同时，通过分析可知当整个序列为升序时，$n_d(P)$达到最大值(d+1)!-1；当整个序列为降序时，$n_d(P)$达到最小值 0。

时间序列伸缩和平移序模式不变，序变换值不变，这一点可从图 1 中观察到。序模式和序变换的这种平移和伸缩不变性，以及序变换值与序模式的一一对应的特点，不仅为时间序列相似性提供了更加符合人类思维和视觉特点的描述方法，还为序模式的快速索引提供了一个有效途径。

1.2　相似性度量

序模式能够比较全面地描述原有序列中的信息，克服以点距离为基础的时间序列误匹配以及物理概念不明确等缺陷。显然，若直接采用序模式对原始时间序列进行描述，不但会引入繁重的计算量，而且对相似性匹配搜索没有实际意义。因此，本文采用时间序列分段线性表示，先将时间序列数据基于时间表示成多段相邻的直线，这样不但可以获得时间序列分段拐点作为特征点，而且实现了对原始时间序列的压缩、降噪和趋势特征的提取。子序列相似性度量，通过子序列的序模式分类和趋势分布距离匹配 2 个步骤实现。

(1)子序列的序模式分类

对数据库中的时间序列进行分段，线性表示获得特征点序列，根据需要设定特征点子序列的长度，并对各个子序列按照式(4)进行序变换，从而可以实现对特征点子序列的划分，将具有相同序变换数值的子序列归为同一类。

(2)趋势分布距离

设 2 个序列S_1和S_2对应的分段数为n，其对应各分段线段的斜率序列分别为$[k_{11}, k_{12}, k_{13}, \ldots, k_{1n}]$和$[k_{21}, k_{22}, k_{23}, \ldots, k_{2n}]$，定义 2 个序列的趋势分布距离为

$$D_k(S_1,S_2) = \sqrt{\sum_{i=1}^{n}\left(\frac{k_{1i}}{K_1} - \frac{k_{2i}}{K_2}\right)^2} \tag{6}$$

$$K_m = \sum_{i=1}^{n} |k_{mi}|, \quad m = 1,2 \tag{7}$$

按照式(6)和式(7)的距离定义，当S_1或S_2序列在水平方向进行缩放时，二者间的趋势分布距离不受影响。趋势分布距离是对 2 个序列内部相对趋势(或称趋势分布关系)相似性的度量，这就保证了趋势分布距离匹配的合理性。

通过对上述 2 点分析可知，如果某特征点子序列 P 与待查询特征点序列 Q 属于同一个序模式类，则二者趋势分布距离愈短，相似匹配程度愈高。

1.3　算法流程

基于序模式的时间序列相似匹配搜索，包括 2 大部分：

(1)索引生成

Step1 初始化，设定分段线性化的规则与相关参数，以及子序列的长度与匹配精度；

Step2 分段线性化，对原始时间序列分段线性化获得特征点序列；

Step3 基于滑动窗的特征子序列序模式提取及序变换；

Step4 建立索引，根据序变换值建立索引。

(2)匹配搜索

Step5 对待匹配子序列进行特征点提取、序模式计算及序变换；

Step6 根据变换的结果进行相似性搜索，得到同一序模式分类的匹配子序列；

Step7 在同一序模式分类内部，根据需要按照趋势分布距离进行二次匹配(或用户直接浏览选取)。

2　关键技术问题研究

2.1　基于滑动窗和递推算法的序模式提取算法

对于时间序列 S，取滑动窗口的长度为 d+1，设相邻 2 个滑动窗口获得的时间子序列为 $S(t)$和 $S(t+\tau)$，它们对应的序模式分别为 $P(t)$和 $P(t+\tau)$，如下所示：

$$S(t) = \{x_t, x_{t+\tau}, x_{t_{t+2\tau}}, x_{t_{t+3\tau}}, ..., x_{t_{t+d\tau-\tau}}, x_{t_{t+d\tau}}\}$$

$$S(t+\tau) = \{x_{t+\tau}, x_{t_{t+2\tau}}, x_{t_{t+3\tau}}, x_{t+4\tau}, ..., x_{t_{t+d\tau}}, x_{t_{t+d\tau+\tau}}\}$$

$$P(t) = \{i_1(t), i_2(t), i_3(t), ..., i_d(t)\}$$

$$P(t+\tau) = \{i_1(t+\tau), i_2(t+\tau), i_3(t+\tau), ..., i_d(t+\tau)\}$$

根据式(4)，对于序模式 $P(t)$和 $P(t+\tau)$，有

$$i_n(t) = \#\{r \mid 0 \leq r < n, x_{t+r\tau} \leq x_{t+n\tau}\}$$

$$i_n(t+\tau) = \#\{r \mid 1 \leq r < n+1, x_{t+r\tau} \leq x_{t+n\tau+\tau}\}$$

由此，对于 $2 \leq n \leq d$，有

$$\begin{aligned} i_n(t) &= \#\{r \mid 0 \leq r < n, x_{t+r\tau} \leq x_{t+n\tau}\} \\ &= \#\{r \mid 1 \leq r < n, x_{t+r\tau} \leq x_{t+n\tau}\} + \#\{r \mid r = 0, x_{t+r\tau} \leq x_{t+n\tau}\} \\ &= i_{n-1}(t+\tau) + c_n \\ &= \begin{cases} i_{n-1}(t+\tau)+1, x_{t+r\tau} \leq x_{t+n\tau+\tau} \\ \mathrm{i_{n\text{-}1}(t+\tau)}, \mathrm{else} \end{cases} \end{aligned}$$

根据上述推导，可得出下面的基于滑动窗和递推算法的序模式提取算法。对于T={t_1, t_2, t_3, t_4 ,..., t_N}，初始化窗口长度W=d+1，窗口序列T_n={t_n, t_{n+1}, t_{n+2}, t_{n+3}, t_{n+4} ,..., t_{n+d}}。按照定义直接计算T_1对应时间序列的序模式$P(T_1)$=[i_1 i_2 i_3 i_4 ... i_d]。基于滑动窗和递推算法的序模式提取算法的伪代码如下：

```
For n=2:1:(length(T)-W+1)
For i=n+1:1:n+d-1
If t_{n-1}≤t_i ; c_{i-1}=1; else; c_{i-1}=0; End
End
For k=1:1:d-1
i_k=i_{k+1}-c_{k+1};
End
按照定义直接计算i_d；
```

$P(T_n)=[i_1\ i_2\ i_3\ i_4\ \ldots\ i_d]$;

End

例如:d=4，$T1$={0.66, 0.55, 0.69, 0.06, 0.93}的序模式为{0 2 0 4}，$T2$={0.55, 0.69, 0.06, 0.93, 0.98}时，根据上面的算法：$[c_2, c_3, c_4]$=[1, 0, 1]，则$T2$ 的序模式为{(2-1), (0-0), (4-1), i_4}，由定义直接计算i_4=4，因此，$T2$ 的序模式为{1 0 3 4}。

2.2 序模式灵敏度

生成特征点子序列的序模式的过程中，可以根据需要设定序模式生成过程的灵敏度，以降低错误舍弃 (false dismissal)的出现概率。灵敏度的设置通过设定一个正数ε实现，在序模式生成过程中，同一子序列中的任意 2 点$x_{t+a\tau}$和$x_{t+b\tau}$，如果$|x_{t+a\tau}-x_{t+b\tau}|\le\varepsilon$，则认为$x_{t+a\tau}$和$x_{t+b\tau}$是相等的，即对小于$\varepsilon$的变化不敏感。

2.3 索引建立

长度为 d 的序模式，依照序变换值分为(d+1)!类，即序模式与序变换值一一对应。因此，可以以序变换值作为索引，并通过二分法实现快速查询。

3 实验结果与分析

实验在基于 Pentium4 主频 2GHz 计算机上进行，编程环境为 Matlab6.1。

3.1 递推算法的实验

实验目的是测试序模式计算的滑动窗递推算法与按照定义直接计算算法的效率。采用测试序列的长度为 10^4点，子序列的长度变化范围为[5, 25]，步长为 1。实验比较见图 2。

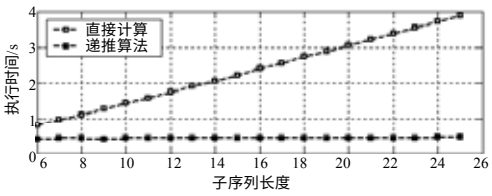

图 2 滑动窗递推算法与定义直接计算算法的实验比较

由图 2 可知递推算法对子序列长度的变化不敏感，算法时间保持在 0.5s 左右；而直接计算算法时间与子序列长度呈线性关系，随着子序列长度的增大，递推算法的优势愈加显著。由此可见，上述序模式的递推计算完全能够满足在线匹配搜索的需要。

3.2 相似匹配搜索实验

本实验目的是测试相似匹配搜索质量和速度。测试数据通过Wiener过程产生，模拟现实世界中的股票市场价格波动等随机现象。测试数据长度为 10^5个点，并进行规一化处理。线性分段阈值参数σ=0.02，获得的关键点个数为 1 927 个。设置灵敏度参数ε=0。

如图 3 所示，S1 为 216 点的待查询子序列，其分段拟合后得到趋势线段 L1，序模式为{1,2,2,4,5}，序变换值为 689。得到的匹配序列为多个不同长度的子序列，其中，趋势分布距离小于 0.6 的子序列为 7 个，长度从 173 到 334 不等。因为这里序模式长度为 5，其序变换值的区间为[0, 720]；查询时间只须从上述区间查询变换值 689 对应的子序列即可。将时间序列数据和索引加载在内存后，整个序模式匹配搜索算法的时间约为 70ms。

如果采用符号映射法(将趋势的升降用 0 和 1 表示)，上述待查询子序列 S1 对应的趋势符号表示为[1 1 0 1 1]，将产生无用候选子序列 76 个，其中典型的子序列如图 4 所示。虽然子序列 L1, L2, L3 和 L4 与待查询子序列的趋势分布距离小于 0.6，但它们显著区别于待查询子序列，其潜在原因是第 4 个关键点的序模式显著不同于对应待查询子序列的序模式。因此，这些子序列不能计入相似查询的结果。由此可见，序模式对相似性的描述更符合人们的思维和观察模式，而符号映射法对子序列的划分则过于粗糙，不仅产生了较大的无用候选集合，还难以构建有效的距离度量与之匹配。

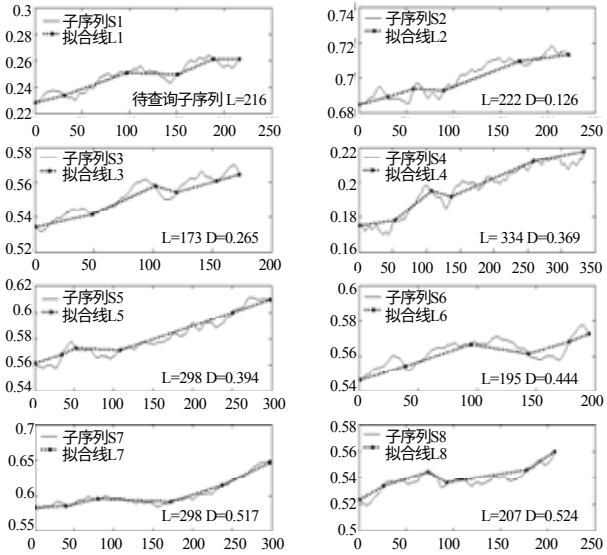

图 3 基于序模式的查询结果示例

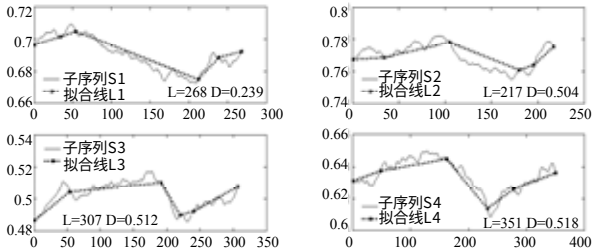

图 4 采用符号映射法得到部分候选子序列

4 结论

将序变换引入到时间序列的匹配搜索中是一项有益的尝试。本文采用的时间序列关键点的序变换是对时间序列相似性的合理抽象，它描述了子序列内部各关键点的相对关系；与之对应的趋势分布距离是对 2 个子序列内部趋势分布一致性的度量。二者的合理搭配实现了对相似性的有效描述。特征点的提取和序变换实现了原始时间序列的匹配搜索从高维空间到一维空间转化，且趋势分布距离对水平缩放不敏感，因此，文中的相似性查询能够实现不同长度子序列相似的高效匹配搜索。

参考文献

1 Agrawal C, Faloutsos A. An Efficient Similarity Search in Sequence Database[C]//Proceedings of the 4th International Conference of Foundations of Data Organization and Algorithms. 1993: 69-84.

2 Chan K P, Fu A W. Efficient Time Series Matching by Wavelet[C]// Proceedings of the 15th IEEE International Conference on Data Engineering. 1999: 126-133.

3 Korn P, Sidiropoulos N, Faloutsos C, et al. Fast Neares-neighbour Search in Medical Image Databases[C]//Proceedings of the 22nd International Conference on Very Large Data Bases, Bombay, India. 1996: 215-226.

（下转第 110 页）