数字趋势序列的子序列匹配算法

贾素玲 陈当阳

(北京航空航天大学 经济管理学院，北京 100083)

摘 要：针对时序数据挖掘中传统趋势序列分析的缺点，提出了数字趋势序列、趋势序列展开等概念.根据数字趋势序列的特点，使用片段斜率所对应的弧度值来度量片段的趋势.针对数字趋势序列的子序列匹配问题，设计了 DTW(Dynamic Time Warping)快速搜索算法.算法分为 3 个部分：DTW 顺序搜索、约束机制、冗余消除机制.并使用实际的股票数据对算法进行了验证.

关 键 词：时序数据挖掘；趋势序列；子序列匹配

中图分类号：TP 301.6

文献标识码：A **文 章 编 号**：1001-5965(2007)03-0350-04

Subsequence matching algorithm between number trend sequences

Jia Suling Chen Dangyang

(School of Economics and Management, Beijing University of Aeronautics and Astronautics, Beijing 100083, China)

Abstract: To overcome the demerits of traditional trend sequences' analysis in temporal data mining, two concepts that were number trend sequences and trend sequences unwrapping were put forward. According to features of number trend sequences, radians were used to represent the trends of line segments. DTW-QS(dynamic time warping quick searching) algorithm was designed to solve the problem of subsequence matching between number trend sequences. The algorithm included three parts: DTW sequential searching, the mechanism of restriction and the mechanism of redundancy control, and DTW-QS algorithm was evaluated via experiments.

Key words: temporal data mining; trend sequence; subsequence matching

时序数据挖掘 TDM(Temporal Data Mining)是数据挖掘的一个重要研究领域.TDM 通过把时间引入数据挖掘技术中，不仅为事件在时间上的先后顺序提供了一个具体的观察窗体，而且可以发现在其它数据挖掘技术中被忽略的因果关系[1].目前，对于 TDM 的研究主要集中在相似性搜索和时序模式挖掘两个领域.

相似性搜索的两个基本问题是全序列匹配 WSM(Whole Sequence Matching)和子序列匹配 SM(Subsequence Matching).在进行时序数据的相似性搜索时，现有的大多数方法都是基于点距离的度量方法[2-3]，这类方法有两个明显缺点：点距离是一种静态的度量方式；不能适应“解析粒度”的变化.把时序数据转化为趋势序列就可以很好地解决这两个问题，但现有的趋势序列分析本质上都属于字符串比较的范畴，用来描述趋势的都是定类尺度，用来衡量趋势间距离的都是编辑距离.本文把这种传统的趋势序列称为字符趋势序列 CTS(Character Trend Sequence).本文提出了数字趋势序列 NTS(Number Trend Sequence)的概念.相对于 WSM 而言，SM 是更为复杂的问题.较为有效的 SM 算法都采用 DTW(Dynamic Time Warping)距离作为距离度量标准，然而，这些算法的搜索效率都很低，搜索结果集中会出现大量的冗余[4-5].针对现有 SM 算法的缺点，结合 NTS 的特点，本文提出了 DTW 快速搜索 DTW-

收稿日期：2006-04-08

基金项目：国家自然科学基金资助项目(70371007)

作者简介：贾素玲(1954—)，女，河南洛阳人，教授，jiasuling@buaa.edu.cn.

QS(DTW Quick Searching)算法,这个算法包括 3 个部分:DTW 顺序搜索、约束机制、搜索冗余处理.

1　问题描述

首先给出几个概念.

定义 1　NTS:给定趋势序列 $M=((v_1,t_1),\cdots,(v_K,t_K))$,趋势指标集 $D=\{m_1,\cdots,m_n\}$,这里 (v_i,t_i)是一个二元组,其中 $v_i\in D$,t_i 是 v_i 的持续时间;如果描述 v_i 的是一种定序或定距尺度,则称 M 为 NTS.

定义 2　SM:给定序列 $Q=(q_1,\cdots,q_n)$,$S=(s_1,\cdots,s_m)$,设 $S_i=(s_t,s_{t+1},\cdots,s_p)=S[t:p]$($1\leqslant t\leqslant p\leqslant m$)是 S 的子序列,如果对于某种距离度量函数 F 和给定的阈值 ε,存在 $F(Q,S_i)\leqslant\varepsilon$,则认为 Q 和 S_i 是相似的.把从 S 中寻找和 Q 相似的子序列的过程称为 SM.

定义 3　趋势序列展开:给定趋势序列 $M=((v_1,t_1),\cdots,(v_n,t_n))$,定义趋势序列 $M_{\text{unwrap}}=(v_1,\cdots,v_1,\cdots,v_n,\cdots,v_n)$,在 M_{unwarp} 中,包含 t_1 个 v_1,……,t_n 个 v_n,任何一个 $v_i(i=1,2,\cdots,n)$的持续时间都是 1,把由 M 生成 M_{unwarp} 的过程称为趋势序列展开.

给定序列 $Q=(q_1,\cdots,q_n)$,$S=(s_1,\cdots,s_m)$.把 Q 转化为趋势序列,即 $M_Q=((v_{Q1},t_{Q1}),\cdots,(v_{Qn},t_{Qn}))$,对于 S,也把它转化为相应的趋势序列 $M_S=((v_{S1},t_{S1}),\cdots,(v_{Sn},t_{Sn}))$;然后按照 DTW 距离和给定的阈值在 M_S 中寻找和 M_Q 相似的子序列 M_{Si},最后得到相似集合 R.

2　DTW-QS 算法

2.1　DTW 顺序搜索

DTW 是一种递归计算距离的方式,它最大的优点在于它允许序列在时间轴上有弹性的移动以便能够在两个时间序列的不同的时间阶段发现相似的波形.虽然 DTW 距离支持在时间轴上有弹性的移动序列,但这种移动必须被控制在一个合理的范围之内,而不能无限制的移动,例如 q_1 可以和 s_1,s_2,s_3 或 s_4 相匹配,但不能和 s_m 相匹配.Sakoe-Chiba 边界是最为常见的 范围匹配条件[6].给定 Sakoe-Chiba 边界的常数 r,两个序列 Q 和 S,其中 $Q=(q_1,\cdots,q_n)$,$S=(s_1,\cdots,s_m)$,对于 S 的第 $i(i=1,2,\cdots,m-(n+2r-1))$个子序列 $s_i=(s_i,\cdots,s_{i+(n+2r-1)})$,可以得到如下的 DTW 距离矩阵 $\boldsymbol{d}$:

$$\boldsymbol{d}_{\text{DTW}}(Q,S_i)=\begin{pmatrix} d_{1i} & d_{1(i+1)} & \cdots & d_{1(i+n+2r-1)} \\ d_{2i} & d_{2(i+1)} & \cdots & d_{2(i+n+2r-1)} \\ \vdots & \vdots & \vdots & \vdots \\ d_{ni} & d_{n(i+1)} & \cdots & d_{n(i+n+2r-1)} \end{pmatrix} \tag{1}$$

在式(1)中,对于任意的 $d_{nj}<\varepsilon$,其中 $j=i+n-1-r,\cdots,i+n-1+2r$,就认为 $S[i:j]$是与 Q 相似的子序列.对于 S 中剩余的 $3r$ 个子序列 S_i,即 $i=m-(n+2r-2),\cdots,m-(n-r-1)$,此时

$$\boldsymbol{d}_{\text{DTW}}(Q,S_i)=\begin{pmatrix} d_{1i} & d_{1(i+1)} & \cdots & d_{1m} \\ d_{2i} & d_{2(i+1)} & \cdots & d_{2m} \\ \vdots & \vdots & \vdots & \vdots \\ d_{ni} & d_{n(i+1)} & \cdots & d_{nm} \end{pmatrix} \tag{2}$$

在式(2)中,对于任意的 $d_{nj}<\varepsilon$,其中 $j=i+n-1-r,\cdots,m$,就认为 $S[i:j]$是和 Q 相似的子序列.DTW 顺序搜索是整个 DTW-QS 算法的基本框架,但这种顺序搜索算法只能对普通的时序数据序列进行 DTW 距离的计算,为了使本文中研究的数字趋势序列也能够使用 DTW 顺序搜索算法,需要按照定义 3 把形如 $M=((v_1,t_1),\cdots,(v_n,t_n))$的趋势序列展开.这样,DTW 顺序搜索算法的时间复杂度也就会在原来的基础上大大增加.

2.2　约束机制

把两个趋势序列 M_Q,M_S 展开,得到 M'_Q和 M'_S.为了简化符号,用 M_Q,M_S 来代表 M'_Q,M'_S,即

$$\left.\begin{aligned} M_Q&=((m_{Q1},\cdots,m_{Q1}),\cdots,(m_{Qn},\cdots,m_{Qn})) \\ M_S&=((m_{S1},\cdots,m_{S1}),\cdots,(m_{Sm},\cdots,m_{Sm})) \end{aligned}\right\} \tag{3}$$

在 M_Q,M_S 的中趋势值总数是 T_Q,T_S.在本文中,把 DTW 中的基本距离函数定义为 Minkowski 距离,即 $D_{\text{base}}(x,y)=|x-y|$.

在 DTW 顺序搜索算法中,由于趋势序列已经被展开,因此 DTW 距离矩阵的计算是非常复杂的,而在实际的趋势序列 M_S 中,和给定的趋势序列 M_Q 相似的子序列肯定是少数,因此计算 DTW 距离矩阵的过程很可能是一种浪费.

定理 1　给定趋势序列 M_Q,M_S,定义变换 R 如下:

$$M_R=R(M)=\{m_i\in M\mid m_i\neq m_{i+1},\ i=1,\cdots,T-1\}+\begin{cases}\{m_T\}, m_T\in M\text{ 且 }m_T\neq m_{T-1}\\ \{\}, m_T=m_{T-1}\end{cases}$$

其中,M 是一个序列;T 是序列 M 的长度;M_R 是一个集合,把 M_R 中的元素按顺序排列,即得到序列 M_{RR}.对 M_Q 和 M_{Si} 进行上述变换,得到 M_{QRR} 和

M_{SiRR}，那么 $\boldsymbol{d}_{\mathrm{DTW}}(M_Q, M_{Si}) \geqslant \boldsymbol{d}_{\mathrm{DTW}}(M_{QRR}, M_{SiRR})$.

证明 要证明定理1，即证明如下命题：

序列 $Q=(q_1,\cdots,q_n)$，$S=(s_1,\cdots,s_m)$，设序列$\hat{Q}=(q_1,\cdots,q_{i-1},q_i,q_i,q_{i+1},\cdots,q_n)$，则 $\boldsymbol{d}_{\mathrm{DTW}}(\hat{Q},S) \geqslant \boldsymbol{d}_{\mathrm{DTW}}(Q,S)$

$$\boldsymbol{d}_{\mathrm{DTW}}(Q,S)=\begin{pmatrix} d_{11} & d_{12} & \cdots & d_{1m} \\ \vdots & \vdots & \vdots & \vdots \\ d_{i1} & d_{i2} & \cdots & d_{im} \\ \vdots & \vdots & \vdots & \vdots \\ d_{n1} & d_{n2} & \cdots & d_{nm} \end{pmatrix}$$

$$\boldsymbol{d}_{\mathrm{DTW}}(\hat{Q},S)=\begin{pmatrix} \hat{d}_{11} & \hat{d}_{12} & \cdots & \hat{d}_{1m} \\ \vdots & \vdots & \vdots & \vdots \\ \hat{d}_{i1} & \hat{d}_{i2} & \cdots & \hat{d}_{im} \\ \hat{d}_{(i+1)1} & \hat{d}_{(i+1)2} & \cdots & \hat{d}_{(i+1)m} \\ \vdots & \vdots & \vdots & \vdots \\ \hat{d}_{(n+1)1} & \hat{d}_{(n+1)2} & \cdots & \hat{d}_{(n+1)m} \end{pmatrix}$$

用数学归纳法证明，先证明 $\hat{d}_{(i+1)j} \geqslant d_{ij}$ $(j=1,2,\cdots,m)$. 当 $j=1$ 时，$\hat{d}_{(i+1)1}=D_{\mathrm{base}}(q_i,s_1)+\hat{d}_{i1} \geqslant d_{i1}$；当 $j=k$ 时，假设 $\hat{d}_{(i+1)k} \geqslant d_{ik}$；当 $j=k+1$ 时，$\hat{d}_{(i+1)(k+1)}=D_{\mathrm{base}}(q_i,s_{k+1})+\min(\hat{d}_{(i+1)k},\hat{d}_{i(k+1)},\hat{d}_{ik})$，$d_{i(k+1)}=D_{\mathrm{base}}(q_i,s_{k+1})+\min(d_{ik},d_{(i-1)(k+1)},d_{(i-1)k})$. 由假设，$\hat{d}_{(i+1)(k+1)}=D_{\mathrm{base}}(q_i,s_{k+1})+\min(d_{i(k+1)},d_{ik})$. 无论 $d_{i(k+1)} \leqslant d_{ik}$ 或 $d_{i(k+1)}>d_{ik}$，都有 $\hat{d}_{(i+1)(k+1)} \geqslant d_{i(k+1)}$. 所以，$\hat{d}_{(i+1)j} \geqslant d_{ij}$ $(j=1,2,\cdots,m)$. 设 $\hat{d}_{(k+1)j} \geqslant d_{kj}$ $(i<k<n, j=1,2,\cdots,m)$，对于 $\hat{d}_{(k+2)j} \geqslant d_{(k+1)j}$ $(j=1,2,\cdots,m)$，完全可以仿效 $\hat{d}_{(i+1)j} \geqslant d_{ij}$ $(j=1,2,\cdots,m)$的证明过程加以证明.

所以 $\boldsymbol{d}_{\mathrm{DTW}}(\hat{Q},S) \geqslant \boldsymbol{d}_{\mathrm{DTW}}(Q,S)$. **证毕**

通过定理1，可以快速排除不符合条件的DTW距离矩阵，加快搜索过程.

2.3 搜索冗余处理

在DTW顺序搜索算法中，需要对式(1)和式(2)两种形式的矩阵进行冗余控制.

对于 M_Q 和 M_{Si}，其中 $i=1,\cdots,m-(n+2r-1)$，DTW矩阵如式(1)，对于任意 d_{nq}，其中 $q \in \{i+n-1-r,\cdots,i+n-1+2r\}$，如果这些 d_{nq} 都小于给定的阈值 ε，则取最小的记为 d_{nx}.

对于 M_Q 和 M_{Si}，其中 $i=m-(n+2r-2),\cdots,m-(n-r-1)$，DTW矩阵如式(2)，对于任意 d_{nq}，其中 $q \in \{i+n-1-r,\cdots,m\}$，如果这些 d_{nq} 都小于给定的阈值 ε，则取最小的记为 d_{nx}.

2.4 算法描述

在本文中，用斜率对应的弧度来表示原始序列分段以后各个片断的趋势. 给定大于1的奇数 C，设 k_i，m_i 分别为原始序列中第 i 个片断的斜率和趋势，则

$$m_i=\begin{cases} m_{-\frac{C-1}{2}}, \arctan k_i \in \left(-\frac{\pi}{2}, -\frac{\pi}{2}+\frac{\pi}{C}\right] \\ \vdots \\ m_{\frac{C-1}{2}}, \arctan k_i \in \left(-\frac{\pi}{2}+\frac{(C-1)\pi}{C}, \frac{\pi}{2}\right) \end{cases} \tag{4}$$

即趋势指标集 $D=\{m_i\}$ $(i=-(C-1)/2,\cdots,(C-1)/2)$，其中 $m_i=i\pi/C$.

在相似性查询开始之前，还有一些预处理过程. 首先是对原始序列的分段，本文中使用标志点分段，这主要是因为本文在数据实验部分使用的是股票收盘价数据[3]；其次是根据当前的应用，选择恰当的误差阈值 ε 来控制误差，选择 Sakoe-Chiba 边界常数 r 来控制 DTW 距离匹配的范围；然后是选择合适的 C，恰当地划分 $(-\pi/2,\pi/2)$ 区间.

算法名称：DTW-QS算法

输入：$M_Q, M_S, r, \varepsilon, C$

输出：R_e

$R_e=\phi$，$M_{QRR}=R(M_Q)$

for each subsequence of M_S

{$M_{SiRR}=R(M_{is})$

if (getDistanceMatrix(M_{QRR}, M_{SiRR})$<\varepsilon$)

matrixd_{DTW} = getDistanceMatrix (M_Q, M_{Si})

}

$R_e=R_e+$getMinSubsequence(matrix$\boldsymbol{d}_{\mathrm{DTW}}$)

算法中，getDistanceMatrix 用来计算 DTW 矩阵，getMinSubsequence 用来去除冗余.

3 数据实验

3.1 实验条件和数据

硬件环境：Celeron® 2.66 GHz 的 CPU，512 M的 DDR400 内存，120 G 的 HDD 硬盘；软件环境：Windows2003 SP1 Server 操作系统. 编制算法的软件是 Matlab7.0.

用青岛海尔 2004 年度的股票收盘价格作为待比较的标准数据，从 2004 年 1 月 2 日开始取 24 个数据点，即数据序列 Q 的长度为 24. 收集了在上海证券交易所上市的所有公司 2004 年度股票的收盘价格，对每一个股票从 1 月 2 号开始，截

取 200 个数据点.

3.2　实验结果

从数据库中任意选取某个股票的数据,所取的序列长度为 200,给定 $C=31$,$r=2$,$\varepsilon=0.25$,分别用 Brute Force 算法和 DTW-QS 算法进行处理,表 1 和表 2 是对应的处理结果[1].

表 1　Brute Force 算法的处理结果

子序列编号	子序列起始点	子序列终止点	相似性距离
1	47	70	0.137 6
2	91	114	0.185 6
3	92	115	0.092 3
⋮	⋮	⋮	⋮
17	106	129	0.092 3
18	107	130	0.185 6

表 2　DTW-QS 算法的处理结果

子序列编号	子序列起始点	子序列终止点	相似性距离
1	1	21	0.157 6
2	47	68	0.137 6
3	93	114	0.092 3
4	107	130	0.185 6

比较上面两个表的结果可以看到,通过时间轴的弯曲,DTW-QS 算法找到了新的子序列 $S[1:21]$,而且大大减少了子序列匹配的冗余.在表 1 中,总共有 18 个子序列和序列 Q 相似,而在表 2 中减少为 4 个.同时,由于时间弯曲效应,在表 2 中得到了和 Q 长度不等的子序列.

分别从数据库中随机选取 100、200、300、400、500 只股票的数据,序列长度均为 200,给定 $\varepsilon=0.25$,表 3 是处理结果.

表 3　不同数目的数据用 DTW-QS 算法的处理结果

序列数目	处理时间/s	有效过滤率	总的子序列数	候选子序列数
100	472.52	0.018 6	17 900	334
200	699.27	0.016 4	35 800	586
300	902.11	0.018 6	53 700	998
400	1 100.23	0.021 2	71 600	1 515
500	1 369.76	0.019 5	89 500	1 724

表 3 中"有效过滤率=候选集中的子序列数/总的子序列数".从表 3 可以看到,DTW-QS 算法在大数据量下的性能表现是很好的.随着序列数目的线性增长,处理时间的增长基本上也是线性的.所以,DTW-QS 算法是一个具有良好伸缩性的算法.限于篇幅,本文不再给出实验结果的详细图表和算法参数分析.

4　结 束 语

数字趋势序列是一种全新的趋势序列,DTW-QS 算法具有良好的伸缩性,大大加快了数字趋势序列的搜索过程.目前,该算法已经在项目组开发的数据挖掘软件中进行了初步应用,取得了较为满意的数据挖掘结果.

参考文献(References)

[1] John F R, Myra S. A survey of temporal knowledge discovery paradigms and methods [J]. IEEE Transaction on Knowledge and Data Engineering, 2002, 14(4): 750－767

[2] Faloutsos C, Ranganathant M, Manolopoulos Y. Fast subsequence matching in time series databases[C]//Richard T, Snodgrass, Marianne W. Proceedings of the ACM SIGMOD International Conference on Management of Data. Berlin: Springer-Verlag, 1994: 419－429

[3] Perng C, Wang H, Zhang S. Landmarks: a new model for similarity-based pattern querying in time series databases [C]//Philip S Y, Arbee L P. Proceedings of 16th International Conference on Data Engineering. San Diego: IEEE Computer Society, 2000: 33－44

[4] Keogh E J. Exact indexing of dynamic time warping[C]//Gupta A, Shmueli O, Widom J. Proceedings of the 28th VLDB Conference. Berlin: Springer-Verlag, 2002: 173－185

[5] Wong S F, Wong M H. Efficient subsequence matching for sequences databases under time warping[C]//Osmar R, Za ane. Proceedings of the 7th International Database Engineering and Applications Symposium. Berlin: Springer-Verlag, 2003: 139－148

[6] Rabiner L, Juang B. Fundamentals of speech recognition [M]. New Jersey: Prentice Hall Press,1993

(责任编辑:张　婧)