时间序列的快速相似性搜索改进算法

肖 晶 黄国兴 赵若韵 黄豫蕾
(华东师范大学计算机系 上海600062)

Efficient Method for Time Series Similar Searching

XIAO Jing HUANG Guo-Xing ZHAO Ruo-Yun HUANG Yu-Lei
(Department of Computer Science, East China Normal University, Shanghai 20062)

Abstract This paper introduces a new method for finding all subsequences similar to a given time series sequence. The method takes into account noise ,offset translation and amplitude scaling. Based on a piecewise linear representation, the speed is exceptionally fast.

Keywords Time series, Piecewise liner representation, Data mining, Similarity search

时间序列(简称时序)数据库是指由随时间变化的序列值或事件组成的数据库,如股票数据、医疗诊断分析、天气数据、化学实验分析等等。数据挖掘在时序数据库上的应用包括相似性搜索、趋势预测等,其中相似性搜索有以下几方面的应用:

·识别具有相似销售量增长模式的不同公司,如微软和Intel公司,从而可以看出两者的利益驱动;

·挖掘股票价格走势规律,如同一支股票出现相似的走势的周期性,或者不同股票相似的价格走势规律等;

·识别两个时序间不明显的关系。如一天中冰激凌的销售量的曲线变化应该与天气温度的曲线变化具有一定的相似性。

对时间序列进行相似性搜索的一般方法:相对较短的搜索时序Q沿着被搜索时序R滑动,文[1]中称之为"序列扫描",其时间复杂度是O((m−n+1)n),m和n分别是时序R和Q上数据点总数。若以上海地区1950年到2000年气温数据为实验对象,m=18250,n=1100,序列扫描方法大概需要执行$2*10^7$个操作,可见效率较差,无法适应实际应用中海量数据的挖掘。

此外,在判定两时序的相似度上还会遇到一些问题,如噪声数据、y轴上偏移和伸缩、时间轴上伸缩[6]等,考虑到这些问题已提出很多相似性搜索方法,如傅里叶变换、离散小波变换[4]、R*树索引等。本文将引用时间序列分段表示思想,以某化学实验随时间变化的温度数据为研究对象,提出一种快速相似性搜索的优化算法。

1 相关概念和结论

1.1 欧几里得距离

判断两个时序是否相似,常见的方法是比较它们之间的欧几里得距离(简称欧氏距离)是否小于给定的阈值。给定两个时序$X=\{x_1,x_2,\cdots,x_n\}$和$Y=\{y_1,y_2,\cdots,y_n\}$,其中x_i和y_i分别代表各个时间点的数值。两时序X和Y之间的欧氏距离表示为:

$$D(X,Y)=\sqrt{\sum_{i=1}^{n}(x_i-y_i)^2}$$

若$D(X,Y)<\varepsilon$,其中ε是>0的常数,则两时序X和Y相似。

但根据欧氏距离判定两时序是否相似具有一定的局限性:

i)若时序X在垂直于时间轴方向向上或向下移动,会得到完全相似的时序Y(如图1)。但该方法就会判定两时序X和Y不相似。

ii)若时序X按比例缩放一定倍数,得到另一个时序Y(如图2)。我们一般认为两时序相似,但根据此判定方法得出错误的结论。

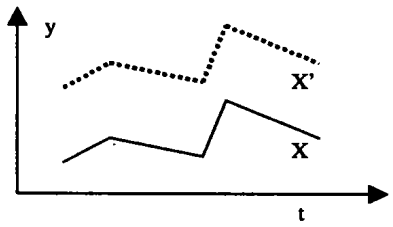

图1 时序X沿y轴向上平移得到X',应该认为X和X'相似

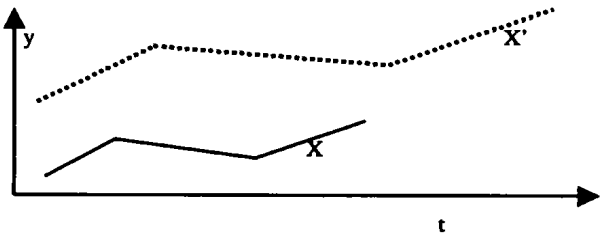

图2 时序X在y方向和t方向同时放大相同倍数得到X',应认为X和X'相似

为了解决以上问题,在对时间序列分析之前,要进行规范化。

1.2 规范化

为解决比例不一致带来的问题,先将原来的时间序列规范化。所谓的规范化,是将每个时间点对应的数值按比例缩放,使它们都落入到较小的区间,如−2.0到2.0,0.0到1.0。有了统一的单位度量,欧氏距离的方法判定相似即可得出正确结论。

定义1 N-序列X是实数集$\{x_1,x_2,\cdots,x_n\}$,其序列X的

肖 晶 硕士研究生,研究方向:数据挖掘。黄国兴 教授,研究方向:智能信息系统,数据挖掘。

平均值 mean(X) 和标准差 std(X) 如下：

$mean(X)=(1/n)\Sigma_{1\leqslant i\leqslant n}x_i$

$std(X)=((1/n)\Sigma_{1\leqslant i\leqslant n}(x_i-mean(X))^2)^{1/2}$

定义2 如果 mean(X)=0且 std(X)=0，则 n-序列 X 是规范化序列。

1.3 分段线性表示

在计算机科学领域中，对现实世界的表现形式一直是个重要的议题，因而须选择一种合适的表现形式，使得其既能抓住时间序列的主要信息，又能方便进行相似性搜索。实验证明，分段线性表示相对于其他表现形式，如傅里叶变换和 R 树，具有更高的效率。本算法将就这种表现形式进行研究。所谓分段线性表示，即将整个时间序列截成若干子序列(不一定等长)，每段用直线近似表示，实质是将连续的时间序列曲线离散成若干条相连接的直线。这种表现形式有以下特点：

·高压缩率。如图3，原始数据是某公司1996年6月到1999年6月股票价格曲线图，该时序上共有369个数据点，经分段处理，分成32个子段(如图4)，数据得到有效的压缩并保留了主要信息；

·对噪声数据的高承受能力，分段过程也是去除噪声的过程；

·视觉直观简洁。

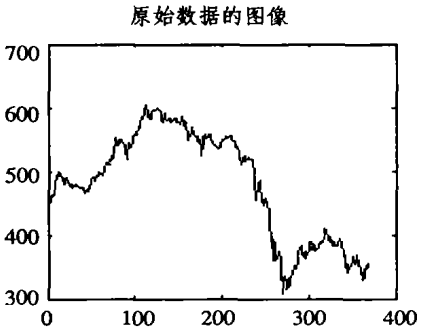

图3 某公司1996年6月到1999年6月股票价格曲线图，有369个数据值

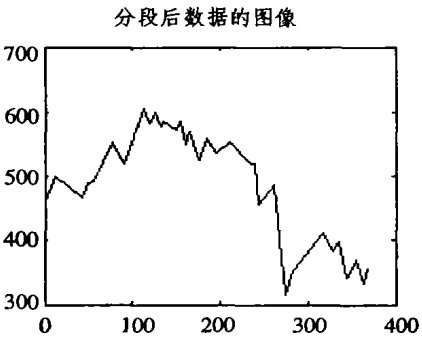

图4 图3中曲线经分段后，不仅除去噪声，而且在保留了主要信息基础上，有效地压缩

目前已知的分段方法有很多，要求保证高压缩率，即段数不能太多，又要保证不丢主要信息。本文在这里不再叙述如何给时间序列分段，下面将着重讨论基于分段时间序列的一种相似性搜索算法。

2 算法描述

2.1 基本搜索算法

输入数据：一个搜索时序 Q(长度相对较短)；一个被搜索时序 R(长度相对较长)；距离阈值 $\epsilon\geqslant 0$。

输出数据：时序 R 中与时序 Q 相似的子时序集合 S。

算法描述

步骤1：规范化两时序 Q 和 R，使得所有时间点表示的数值落入较小范围内，如－2.0到2.0；

步骤2：分段表示，即将两时序 R 和 Q 按一定算法分成若干子段；

步骤3：将时序 Q 的最左端与 R 的最左端固定，并开始滑动；

步骤4：计算时序 Q 与 R 的相应子时序 R_i 的欧氏距离 d_i

i)若 d_i 小于给定的阈值 ϵ，则认为时序 R_i 与 Q 相似，并将 R_i 放入集合 S 中；

ii)若 d_i 不小于给定的阈值 ϵ，则移动时序 Q，使之与 R_{i+1} 对齐，重复步骤3，直到 Q 的最右子段与 R 的最右子段相匹配，算法终止。

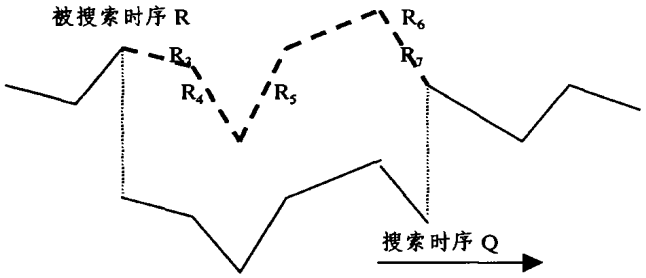

图5 搜索时序 Q 沿着被搜索时序 R 滑动，当前 Q 在与 R 中子时序{R_3 R_4 R_5 R_6 R_7}相匹配

算法分析：该算法时间复杂度是 O(MN)，M 和 N 分别是时序 R 和 Q 的段数。若 M、N 值很大的话，其效率有待提高。其次，认为时间序列 X 在时间轴方向一定程度伸缩后得到的时序 X'，仍与原来的时序 X 相似，该算法并未考虑这一点。

2.2 S7算法

Eamonn Keogh 曾在文[1]中提出 S7算法(Similarity Searching while Shrinking and Stretching using Segmented Sequential Scanning)，有效地解决了效率和时间轴方向的伸缩问题。

S7算法描述：从搜索时序 Q 中提取最左端和最右端的两个子段 Q_l 和 Q_r，将它们与 R 的每个子段 R_i 比较两次，一次是 Q_l 和 Q_r 伸长最大程度后的时序与 R_i 比较，一次是收缩最大程度后的时序。建立表格，前两行分别对应 Q_l 和 Q_r，每列分别对应 R_i，每个单元格有两个值，对应两次比较结果。如单元格(2,9)中两个值表示 Q_r 伸长最大程度 Q_{rmax} 和收缩最大程度 Q_{rmin} 与 R 的第9个子段 R_9 相异度，表示如图6。下一步，扫描表格，找出 Q_l 和 Q_r 与 R_i 相异度小于给定的阈值且都匹配同一个 R_i 的所有 R_i。具体描述详见文[1]。

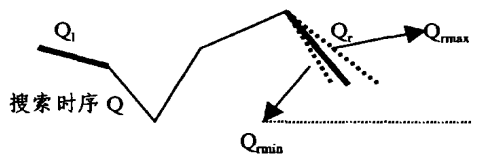

图6 Q_l 和 Q_r 分别代表搜索时序 Q 的左右两端子段；Q_{rmin} 和 Q_{rmax} 分别代表 Q_r 伸长和收缩最大程度后子段

S7算法的不足：首先，只选择Q的最左和最右子段作为Q_l和Q_r与R_i比较，合理的作法是选择Q中与所有R_i相似个数最少的两个Q_i作为Q_l和Q_r，这样扫描表格就很容易找到与Q相似的R_i；其次，该算法大部分执行时间都在构造表，如何使用其他方法构造表以提高效率有待推敲；针对以上两点不足，本文提出一种快速搜索算法。

2.3 快速搜索算法

输入数据：一个搜索时序Q(长度相对较短)；一个被搜索时序R(长度相对较长)；伸长和收缩程度阈值$\epsilon1$、$\epsilon2 \geqslant 0$。

输出数据：时序R中与时序Q相似的子时序集合S。

算法描述：

步骤1、2、3同基本算法；

步骤4：计算两时序R和Q中各个子段的直线倾斜角度r_i和q_i；

步骤5：建表1，扫描时序R每个子段的角度r_i，将所有r_i模10得到的值归入相应列，如第5列可能存放倾斜角度在40度(包括40度)和50度(不包括50度)之间的所有r_i；对Q采用同样方法；删除不存放内容的列；

步骤6：合理选择Q_l和Q_r，两个条件：

i)不考虑存放两个及两个以上Q_i的列；

ii)存有一个Q_i，且存有R_i个数最少两列所对应的Q_i，被作为Q_l和Q_r；

步骤7：先根据给定$\epsilon1$、$\epsilon2$，计算Q_l和Q_r最大伸长和收缩的角度阈值q_{lmax}和q_{limin}；建表2，共两列，分别对应在(q_{lmin}，q_{lmax})和(q_{rmin}，q_{rmax})范围内的R_i，只需在表1中查找符合搜索范围的R_i；

步骤8：在与Q_l和Q_r相似的R_i中找出完整连续的子段，计算该子段的首尾两个子段与Q首尾相似度，如相似，则认为该子段与Q相似。

3 应用实例

以某化学实验随时间变化的温度数据(如图7)为对象，使用MATLAB数学工具，采用本文提出的算法进行相似性搜索的数据挖掘。该实验中记录了温度数据共172个，经分段处理，划分为26个子段，所有子段直线的倾斜角集合为R；搜索时序Q(如图8)的有三个子段，直线倾斜角集合为Q。

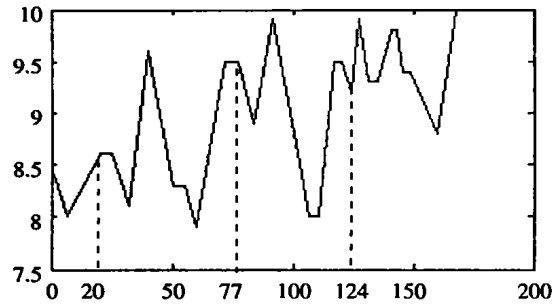

图7 某化学实验随时间变化的温度曲线

图8 搜索时序

R = {− 81.4692 77.7749 0 − 82.0304 86.9471 −84.2978 0 − 82.8750 85.7108 0 − 83.3456 85.4261 − 85.4860 0 86.7094 0 − 82.4054 87.5460 − 86.1859 0 83.1572 0 − 85.7108 0 −79.6111 86.1859}

Q={80.2364 0 −79.4104}

根据本文的算法创建表1，如表1。

表1 时序R和Q按算法分类

−8	−7	0	7	8
R1,R4,R6,R8,R11,R13 R17,R19 R23	R25	R3,R7,R10,R14 R16,R20 R22,R24	R2	R5,R9,R12,R15,R18,R21,R26
	Q3 (1)	Q2 (8)		Q1 (7)

*注：第三行括号中数字为该列R_i个数

根据表1中的信息，按照算法应该选Q_1和Q_3作为Q_l、Q_r，子时序的Q_l、Q_r的直线倾斜角度分别是81.2364和−79.4104。根据给定的扩张和收缩角度阈值5.0度和4.5度，可以得到角度伸缩范围Q_l和Q_r角度伸缩范围(77.2364，85.7364)、(−84.4104，−76.9104)。根据算法建立表格如表2。

表2 Q_l和Q_r角度伸缩范围内所有R_i字段

$Q_l(Q_1)$		$Q_r(Q_3)$	
76.7364	86.7364	−84.4104	−74.9104
R2,R5,R9,R12,R15,R21,R26		R1,R4,R6,R8,R11,R17,R25	

扫描表2，最终在R中找到3个子段与Q相似，分别是{R2,R3,R4}{R9,R10,R11}{R15,R16,R17}。

该算法的时间复杂度是创建表和扫描表的时间之和。创建表的时间复杂度是O(M+N)，其中M、N分别是两时序R和Q的段数；扫描表的时间复杂度视情况而定，但肯定远远小于O(M)。因而合理选择Q_l、Q_r，算法的效率会明显提高。

结束语 对时间序列的数据挖掘研究越来越受到重视，其在金融分析和科学实验分析等方面的应用也越来越广泛。本文基于S7算法的思想，提出改进的时间序列相似性搜索方法，克服了S7中存在的不足，实验过程中表现出较好的性能。但该方法还有待更进一步的研究，例如采用何种算法能够很好地给时间序列分段，且有利于本文的相似性搜索算法进行等，因而更多工作有待深入展开。

参考文献

1 Keogh E. A Fast and Robust Method for Pattern Matching in Time Series Databases. 1997

2 Faloutsos C，Ranganathan M，Manolopoulos Y. Fast Subsequence Matching in Time-Series Databases. SIGMOD-Proceedings of Annual Conference，1994

3 Das G，et al. Finding Similar Time Series. 1996

4 Chan Kin-pong. Ada Wai-chee Fu. Efficient Time Series Matching by Wavelets. 1999

5 Agrawal R，et al. Similarity Search in Sequence Databases. 1993

6 Agrawal R，et al. Fast Similarity Search in the Presence of Noise，Scaling and Translation in Time-Series Databases. 1995