文章编号: 1001-0920(2004)03-0299-04

基于增量DFT的相似时序检索算法

郑 烇[1], 朱 明[1], 王俊普[1], 蔡庆生[2]
(1. 中国科学技术大学 自动化系, 安徽 合肥 230027; 2. 中国科学技术大学 计算机系, 安徽 合肥 230027)

摘 要: 首先提出一种在时域上计算时序数据扩展距离的新算法, 该算法时间复杂度为$O(n\times m)$, 能够解决时序数据在Y轴上的漂移和伸缩后仍然保留相似性的问题; 然后提出一种在频域上计算时序数据扩展距离和在长时序中搜索相似子序列的新算法, 该算法时间复杂度仅为$O(n\times f_c)$, 效率很高, 便于在线实现, 而且同样能够适应时序数据扩展距离的定义; 最后给出时序数据和线性加权时序数据的增量式DFT 算法, 可以对长时序的各个窗口进行增量式的降维, 将传统的$O(n\times m\times f_c)$工作改进成$O(n\times f_c)$.

关键词: 时间序列; 子时序; 相似度; 增量DFT
中图分类号: TP18 **文献标识码**: A

Incremental DFT based quick search algorithm of finding similar sequence

ZHENG Quan[1], *ZHU Ming*[1], *WANG Jun-pu*[1], *CAI Qing-sheng*[2]
(1. Department of Automation, University of Science and Technology of China, Hefei 230027, China; 2. Department of Computer Science, University of Science and Technology of China, Hefei 230027, China. Correspondent: ZHENG Quan, E-mail: qzheng@ustc.edu.cn)

Abstract: A new algorithm, with the time complexity $O(n\times m)$, of computing the time series extended distance on time domain is presented. The algorithm can solve the problems brought by the data scaling and shifting on the Y axis. An algorithm of computing the time series extended distance on reduced frequency domain and finding the most similar subsequence from given long sequence is also given. The algorithm, with the time complexity $O(n\times f_c)$, can be implemented online and adapts to the extended distance definition. A incremental DFT algorithm on time series data and linear weighted time series is propesed, which greatly reduces the dimension on each window of a long sequence.

Key words: time series; subsequence; similarity; incremental DFT

1 引 言

时序数据挖掘在现实世界中有着广泛的应用, 而时序数据相似度的比较是时序数据挖掘的基础性问题, 具有广阔的应用前景. 目前, 时序数据相似度比较和快速相似子序列检索的方法除欧几里得技术外, 还有频域法[1~4]、段化法[5,6]和波形描述语言法[7]等.

为了保留时序数据在线性变换之后的相似性, 前人给出了时序的扩展距离. 对于扩展相似时序检索的研究, 比较重要的工作是Chu 等[8]和Agrawal 等给出的[9]. 但是, 文献[8]给出的距离是非对称的, 可能会导致与人类直觉相违背的结果; 另外算法本质上是在时域上计算距离, 代价较高. 文献[9]给出的相似时序的搜索算法存在以下问题: 1) 用简单的

收稿日期: 2003-02-10; **修回日期**: 2003-06-16.
基金项目: 国家自然科学基金资助项目(60272040).
作者简介: 郑烇(1970—), 男, 安徽合肥人, 讲师, 博士, 从事智能控制、模式识别等研究; 蔡庆生(1938—), 男, 江苏南京人, 教授, 博士生导师, 从事人工智能、机器学习等研究.

归一化技术解决子序列的相似度比较中漂移和伸缩问题, 不具备一般性; 2) 算法繁琐, 代价高

本文对基本的频域方法进行了扩展, 使之适用于扩展相似时序的检索. 主要思想是: 给出一个在时域上计算时序数据扩展距离的解析结果; 给出基于频域解析解的时序数据扩展距离的计算技术以及相应的相似时序的快速检索技术, 该技术由于在降维之后的频域上计算, 效率较高, 时间复杂度为$O(n\times f_c)$, 而且能够适应时序数据的扩展距离; 针对在相似子序列检索问题中, 需要对长时序各窗口进行DFT 降维处理问题, 本文提出了长时序各窗口的增量DFT 技术, 并提出了线性加权时序数据的增量DFT 技术, 时间复杂度由传统的$O(n\times m\times f_c)$改进为$O(n\times f_c)$.

2 扩展的时序数据距离及其解析解

定义 1 长度相同的一维时间序列数据 $x=[x_0,x_1,\cdots,x_{m-1}]^{\mathrm{T}}$, $y=[y_0,y_1,\cdots,y_{m-1}]^{\mathrm{T}}$, 扩展的非对称距离定义为

$$d(x,y)=\min_{a,b}\left(\sum_{i=0}^{m-1}(x_i-ay_i-b)^2\right)^{1/2}. \tag{1}$$

这种定义的优点在于能够保持时序数据在伸缩和漂移之后的相似性, 能够适应不同传感器的伸缩比例和漂移量的不同. 但这种距离是非对称的, 不符合人的习惯, 因此文献[1] 给出另外一个时序数据距离的定义, 该距离为两个非对称距离$d(x,y)$和$d(y,x)$的最小值

尽管Chu 等[8] 给出了一个将时序数据映射到移动消除平面上, 从而在该平面计算时序数据距离的算法, 但形式较为复杂. 本文首先给出一种通过计算a和b的最优参数, 从而直接计算时序数据扩展距离的解析方法

定理 1 长度为m 的一维时间序列数据x 和y的非对称距离的解析最优解为

$$d(x,y)=\left(\sum_{i=0}^{m-1}(x_i-a_ty_i-b_t)^2\right)^{1/2}. \tag{2}$$

其中

$$a_t=\frac{\sum_{i=0}^{m-1}x_iy_i-m\bar{x}\bar{y}}{\sum_{i=0}^{m-1}y_i^2-m\bar{y}^2},\ b_t=\bar{x}-a_t\bar{y}.$$

对称距离为$d(x,y)$和$d(y,x)$的最小值. 相应的相似时序检索算法的时间复杂度在全匹配情况下为$O(m)$, 而在子序列搜索情况下为$O(n\times m)$. 其中: n 为长时序的长度, m 为子序列的长度. 这种在时域上计算时序数据距离的算法的优点在于能够避免在(a,b) 平面空间进行搜索, 能够根据时序数据的值很快地计算出它们扩展距离的解析解

3 在频域上计算时序数据距离的技术

定理 1 基于时域解析解的相似时序检索算法, 由于在时域上计算, 代价较高, 不便于在线实现. 而一般的频域法不能适应时序数据的扩展距离的定义. 如何对普通的频域法进行扩展来计算时序数据的距离, 而且能够适应时序数据扩展距离的定义, 是本文要解决的问题

引理 1 设时序数据x 对应的傅立叶系数为X_f, 时序数据x 经线性变换后的时序数据$y=a\times x+b$, 则时序数据y 的第f 项傅立叶参数Y_f 为

$$Y_f=aX_f+\frac{b}{\sqrt{m}}\frac{1-\mathrm{e}^{cfm}}{1-\mathrm{e}^{cf}}. \tag{3}$$

其中: $c=-(\mathrm{j}2\pi)/m$, X_f 和Y_f 分别为时序数据x 和y 的第f 个频率组份.

定理 2 时序数据x 和y 的扩展非对称距离近似为

$$d(x,y)\approx\left(\sum_{f=0}^{f_c-1}\left|X_f-a_tY_f-\frac{b_t}{\sqrt{m}}\frac{1-\mathrm{e}^{cfm}}{1-\mathrm{e}^{cf}}\right|^2\right)^{1/2}. \tag{4}$$

其中

$$a_t=\frac{\sum_{f=0}^{f_c-1}(X_f\oplus Z_f)\sum_{f=0}^{f_c-1}(Y_f\oplus Z_f)-\left(\sum_{f=0}^{f_c-1}(Y_f\oplus Z_f)\right)^2}{\sum_{f=0}^{f_c-1}(X_f\oplus Y_f)\sum_{f=0}^{f_c-1}(Z_f\oplus Z_f)-\sum_{f=0}^{f_c-1}(Y_f\oplus Y_f)\sum_{f=0}^{f_c-1}(Z_f\oplus Z_f)}\rightarrow\leftarrow;$$

$$b_t=\frac{\sum_{f=0}^{f_c-1}(X_f\oplus Y_f)-a_t\sum_{f=0}^{f_c-1}(Y_{\backslash-f}\oplus Y_f)}{\sum_{f=0}^{f_c-1}(Y_f\oplus Z_f)};$$

f_c 是截止频率; $Z_f=\frac{1-\mathrm{e}^{cfm}}{\sqrt{m}(1-\mathrm{e}^{cf})}$ 是一个为便于表达而引入的复数序列; X_f 是时序数据x 的第f 项傅立叶参数; Y_f 是时序数据y 的第f 项傅立叶参数; 函数$\oplus$ 是一个从复数到实数的映射, 实际上是两个复数实部的乘积加上虚部的乘积

相应的子序列搜索算法, 可以一边进行增量式的DFT, 一边在频域上使用频域解析解计算出扩展

距离, 从而进行相似性比较, 时间复杂度为$O(n \times f_c)$. 相对于在时域上搜索相似子序列的时间复杂度$O(n \times m)$, 因为f_c一般取值为2～5, 比m要小2～3 个数量级, 所以该算法的效率很高, 便于在线实现. 而且该算法能够保持时序数据线性变换之后的相似性, 能够适应扩展的距离定义. 为简单计, 将本文提出的带增量DFT 以及能够解决漂移和伸缩问题并在频域上搜索相似子序列的算法称为扩展频域法.

4　时序数据和线性加权时序数据的增量式傅立叶变换

在相似子序列搜索的问题上, 第 3 节描述的算法需要对时序数据中的每一个子序列窗口进行离散傅立叶变换, 而根据传统的DFT 公式, 若要求取其低阶f_c个傅立叶参数, 时间复杂度为$O(n \times m \times f_c)$, 代价较高. 下面提出一种增量式的傅立叶变换算法, 能够大大提高变换的效率, 便于在线实现.

将长时序x分成$n - m + 1$个长度为m的相互重叠的时间窗口, 使用xw_i表示时序数据的第i个窗口, 用$XW_{i,f}$表示该时间窗口的第f个频率成分.

定理 3　时序数据时间窗口xw_i的第f个傅立叶参数$XW_{i,f}$与其前一个时间窗口的第f个傅立叶参数$XW_{i-1,f}$的关系为

$$XW_{i,f} = XW_{i-1,f}/e^{cf} + \Delta_{i,f}. \tag{5}$$

其中

$$\Delta_{i,f} = \frac{1}{\sqrt{m}}\left(xw_{i,m}e^{cfm} - \frac{xw_{i-1,0}}{e^{cf}}\right) = \frac{1}{\sqrt{m}}\left(x_{i+m}e^{cfm} - \frac{x_{i-1}}{e^{cf}}\right).$$

在某些场合, 人们一般把时序数据x离当前时刻$(m-1)$最近的一些点的重要性看得比远一些的点的重要性大一些. 为简单起见, 引入一个斜率为k的线性遗忘函数$f(t)$作为对距离贡献的权重, 即

$$f(t) = z + kt = (1 - km + k) + kt. \tag{6}$$

定义 2　长度为m的两个一维时间序列数据x和y的线性遗忘距离$d_w(x,y)$为

$$d_w(x,y) = \left(\sum_{t=0}^{m-1}(x_t - y_t)^2 f(t)^2\right)^{1/2}. \tag{7}$$

时序数据x第i个窗口的第t个数据$xw_{i,t}$经线性加权变成$xw'_{i,t}$, 其关系为

$$xw'_{i,t} = xw_{i,t}f(t) = xw_{i,t}(1 - km + k + kt). \tag{8}$$

由定义 2, 有

$$d_w(x,y) = \left(\sum_{t=0}^{m-1}(x_t - y_t)^2 f(t)^2\right)^{1/2} = \left(\sum_{t=0}^{m-1}(x'_t - y'_t)^2\right)^{1/2} = d(x',y').$$

可见计算两个时序子序列的加权距离, 相当于计算两个加权时序的欧几里得距离. 由Parseval 规则, 可以通过只取加权时序数据频域上前几个频率组进行近似距离的计算, 从而进行快速相似时序检索.

现在的问题是如何增量式地获得经过线性加权之后的各个窗口的傅立叶参数. 时间窗口xw, 经过加权后得到的时间窗口xw', 已知前一窗口的各项DFT 参数$XW_{i-1,f}$, 线性加权傅立叶参数$XW'_{i-1,f}$和辅助参数$XWT_{i-1,f}$, 如何增量式地得到这个线性加权数据窗口的各项DFT 参数$XW'_{i-1,f}$. 其中

$$XWT_{i,f} = \frac{1}{\sqrt{m}}\sum_{t=0}^{m-1} tx_{w,t}e^{cft}.$$

不难得到以下两个引理:

引理 2

$$XWT_{i,f} = \frac{1}{e^{cf}}(XWT_{i-1,f} - XW_{i,f}) + \frac{(m-1)x_{w,m}e^{cfm} + x_{w,0}}{e^{cf}\sqrt{m}}.$$

引理 3　经过线性遗忘的时序窗口XW_i'的第f项傅立叶参数$XW'_{i,f}$为

$$XW'_{i,f} = (1 - km + k)XW_{i,f} + kXWT_{i,f}.$$

由此不难得出, 计算经过线性遗忘的时序窗口$XW'_{i,f}$的傅立叶参数的增量算法:

定理 4　增量式计算线性遗忘时序窗口的傅立叶参数的递推公式为

$$XW_{0,f} = \frac{1}{\sqrt{m}}\sum_{t=0}^{m-1} x_{0,t}e^{cft}, \tag{9}$$

$$XWT_{0,f} = \frac{1}{\sqrt{m}}\sum_{t=0}^{m-1} tx_{0,t}e^{cft}, \tag{10}$$

$$XW_{i,f} = \frac{XW_{i-1,f}}{e^{cf}} + \frac{1}{\sqrt{m}}\left(x_{w,m}e^{cfm} - \frac{x_{w-1,0}}{e^{cf}}\right), \tag{11}$$

$$XWT_{i,f} = \frac{1}{e^{cf}}(XWT_{i-1,f} - XW_{i-1,f}) + \frac{(m-1)x_{w,m-1}e^{cfm} + x_{w-1,0}}{e^{cf}\sqrt{m}}, \tag{12}$$

$$XW'_{i,f} = (1 - km + k)XW_{i,f} + kXWT_{i,f}. \tag{13}$$

获得了各个窗口的加权傅立叶参数后, 便可以在降维之后的频域上计算时序数据的近似加权距

离, 从而获得相似时序检索的高效率. 不难看出, 增量式DFT 算法的时间复杂度为$O(n \times f_c)$, 比传统的DFT 算法的时间复杂度$O(n \times m \times f_c)$ 要小得多.

5 实 验

扩展频域法与时域法运行时间的比较如表1 和表2 所示. 表1 中时序长度n = 200 000, 截止频率f_c = 3; 表2 中子序列长度m = 2 000, 截止频率f_c = 3. 从实验结果可以看出, 扩展频域法计算时序数据距离的算法的耗时仅相当于时域法的1/10～1/50, 而且能够适应时序扩展距离的定义, 提高了检索算法的效率.

表1 时域法、扩展频域法运行时间随子序列长度变化

子序列长度m	时域法/s	扩展频域法/s
500	32.32	3.391
1 000	64.44	3.375
1 500	98.98	3.359
2 000	132.36	3.406
2 500	164.97	3.390

表2 时域法、扩展频域法运行时间随时序长度变化

序列长度n	时域法/s	扩展频域法/s
50 000	32.65	0.844
100 000	69.53	1.687
150 000	100.79	2.547
200 000	132.36	3.406
250 000	169.06	4.297

6 结 语

本文给出了一种在频域上计算时序数据扩展距离的解析算法, 从而提供了搜索相似子序列的新技术. 通过实验验证了该算法的效率要比基于时域算法的效率高很多, 便于在线实现, 而且能够适应时序数据扩展距离的定义. 本文还给出了时序数据和线性加权时序数据增量式DFT 的算法, 该算法能够大大提高长时序中各个窗口DFT 降维的效率, 将传统时间复杂度为$O(n \times m \times f_c)$ 的工作改进为$O(n \times f_c)$.

参考文献(References):

[1] Agrawal R, Faloutsos C, Swami A. Efficient similarity search in sequence database[A]. *FODO*[C]. Evanston, Illinois, 1993. 69-84.

[2] Faloutsos Christos, Ranganathan M, Manolopulos Yannis. Fast subsequence matching in time series databases [A]. *Proc ACM SIGMOD* [C]. Minnerapolis, 1994. 419-429.

[3] Rafiei D, Mendelzon A O. Efficient retrieval of similar time sequences using DFT [A]. *FODO* [C]. Kobe, 1998. 203-212.

[4] Chan K P, Fu A W C. Efficient time series matching by wavelets[A]. *ICDE*[C]. Sydney, 1999. 126-133.

[5] Keogh Eamonn, Padhraic Smyth. A probabilistic approach to fast pattern matching in time series databases[A]. *Proc of the 3rd Conf on Knowledge Discovery in Databases and Data Mining*[C]. Menlo Park: AAAI Press, 1997. 24-30.

[6] Keogh Eamonn, Michael J Pazzani. An enhanced representation of time series which allow fast and accurate classification, clustering and relevance feedback[A]. *Proc of the 4th Int Conf of Knowledge Discovery and Data Mining*[C]. Menlo Park: AAAI Press, 1998. 239-241.

[7] Rakesh Agrawal, Giuseppe Psaila, Edward L. Wimmers, mohamed zait, querying shapes of histories [A]. *Proc of the 21st VLDB Conf*[C]. Zurich, 1995. 502-514.

[8] Chu K K W, Wong M H. Fast time-series searching with scaling and shifting [A]. *Proc of the ACM Symposium on Principles of Database Systems*[C]. Philadelphia, 1999. 237-248.

[9] Agrawal R, Lin K I, Sawhney H S, et al. Fast similarity search in the presence of noise, scaling and translation in time-series databases[A]. *Proc* 1995 *Int Conf Very Large Data Bases* (*VLDB* '95)[C]. Zurich, 1995. 490-501.