文章编号: 1001-0920(2007)02-0160-04

基于波动特征的时间序列数据挖掘

武红江, 赵军平, 彭勤科, 黄永宣
(西安交通大学 电信学院, 西安 710049)

摘 要: 针对相似度搜索是时间序列数据挖掘的基础, 构造鲁棒的动态时间弯曲距离是相似性研究的关键,考虑时间序列特征点的重要意义,引入一种时间序列波动点的抽取方法,采用二叉特征树结构对原序列进行再表达. 该方法既提取了序列整体趋势信息,又有效约减了数据维数. 对多个数据集的层次聚类实验表明,在保证较高准确率情况下,该方法显著提高了 DTW 的计算效率.

关键词: 数据挖掘; 相似度搜索; 动态时间弯曲距离; 特征抽取; 聚类
中图分类号: TP393 **文献标识码**: A

Data mining based on fluctuation feature in time series

WU Hong-jiang, ZHAO Jun-ping, PENG Qin-ke, HUANG Yong-xuan
(School of Electronics and Information Engineering, Xi'an Jiaotong University, Xi'an 710049, China. Correspondent: WU Hong-jiang, E-mail: hongjiangwu@163.com)

Abstract: Similarity search is the foundation of data mining in time series, while constructing a robust dynamic time warping distance is the first step. Considering the importance of feature points of time series, a feature extraction method based on fluctuation is proposed, and a binary feature tree building algorithm is given. The method extracts the whole changing trend and meanwhile effectively reduces data dimensions. Clustering experiments on several datasets show that the new method is much faster and more accurate than other methods.

Key words: Data mining; Similarity search; Dynamic time warping distance; Feature extraction; Clustering

1 引 言

时间序列广泛存在于气象、电力、经济金融等诸多领域. 采用数据挖掘的方法揭示数据内部规律已成为时间序列分析的一个重要研究方向. Debregeas[1]提出了对大规模时间序列数据库的聚类算法. Gudes 等[2]针对具有连续时间序列集挖掘关联规则,并对股票数据作了性能测试. 李爱国[3]对在线时间序列提出了一种稳健分割算法 OLS 以挖掘序列模式. 然而无论是分类、聚类还是关联规则挖掘,都需要解决时间序列的相似度问题,相似性搜索是时间序列数据挖掘的研究基础[4,5]. 由于时间序列存在各种复杂变形(如平移、伸缩、间断等),且变形时间和变形程度都无法预料,传统的欧氏距离已经无法胜任. 目前,动态时间弯曲(DTW)相似距离的稳定性已在国内外得到验证[4-9],但 DTW 的计算复杂性限制了其进一步应用.

时间序列数据量大,最值得关注的往往是其特征点. 基于序列特征点的重要意义,本文引入一种特征点抽取方法,构造了二叉特征树对原序列进行再表达,并在此基础上采用 DTW 距离进行相似度匹配. 测试表明,本文方法可以显著提高相似性搜索效率,且在多个实验中均维持较高的准确率.

2 相似度距离

广泛采用的欧氏距离及其改进对时间轴的变形非常敏感,一些轻微的改变会导致欧氏距离发生很大变化[4,5,7,9]. 其原因是欧氏距离对数据采取无差别的一一对应计算策略,动态时间弯曲的相似距离可以弥补欧氏距离缺陷.

DTW 最早用于语音处理方面. Berndt 等[7]首先将其引入到数据挖掘领域. 与欧氏距离的一一对应计算不同,它通过计算弯曲路径上选择的匹配点间的距离表征序列的相似度.

给定两条时间序列 $X = (x_1, x_2, \cdots, x_i, \cdots, x_m)$ 和 $Y = (y_1, y_2, \cdots, y_j, \cdots, y_n)$,构造 $m \times n$ 的矩阵 d

收稿日期: 2005-11-21; **修回日期**: 2006-01-05.
基金项目: 国家自然科学基金项目(60373107).
作者简介: 武红江(1977—),女,西安人,博士生,从事数据挖掘及应用的研究; 彭勤科(1962—),男,陕西凤翔人,教授,博士生导师,从事数据挖掘及应用、实时并行信息处理与控制系统等研究.

$=(d(i,j))_{m\times n}$,元素 $d(i,j)$ 表征序列点对 (x_i,y_j) 间的距离.

定义 1(弯曲路径 W)　在距离矩阵 d 中,定义序列 X,Y 间连续的二元匹配关系组 w_k 的集合为 W,其中 $w_k=(i,j)_k$,$1\le i\le n$,$1\le j\le m$,K 是路径长度,即

$$W=(w_1,w_2,\cdots w_k,\cdots,w_K),$$
$$\max(m,n)\le K\le m+n-1. \quad (1)$$

在所有弯曲路径集 R 中,最关心的是序列相似程度最大(距离值最小)的弯曲路径. 因此有如下定义:

定义 2　序列 X,Y 的相似度 $D(X,Y)$ 为弯曲路径中距离最小值. 计算公式为

$$D(X,Y)=\min_{W\in R}\left(\frac{1}{K}\sum_{k}^{K}W_k\right). \quad (2)$$

DTW 算法对时间序列具有很强的适应性[4,5],包括处理各种时域和幅度的变化. 然而经典 DTW 算法复杂度为 $O(mn)$,在改善计算速度方面学者做了很多研究工作. 目前常用的方法是 Yi 等[8] 提出的 LowerBounder 技术,它通过设置计算窗口(如图 1 所示)的方法得到目标下限,即只计算窗口内对应的相似距离,从而尽量避免无谓的 DTW 计算. 文献[7]则通过局部极点线性分段的方法来约减维数,从而提高了计算效率.

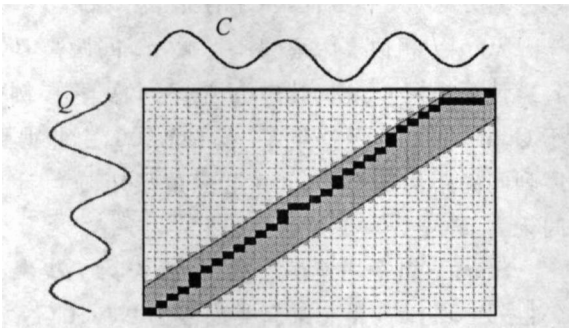

图 1　Sakoe-Chiba 计算窗口(带状区域)

此外,从 DTW 的构造与实现角度来看,其搜索方向为整体向前行进,缺乏一种依据全局形态进行快速匹配的机制,尤其是当两条序列长度差异过大时,容易产生病态弯曲路径[10].

无论是设置计算窗口还是采用极点分段,其目的是在保证一定准确率的同时,约减参与 DTW 计算的数据点数. 与文献[7]思路不同,以下将首先提取序列波动特征点,对这些特征点进行 DTW 计算,可提供一种全局趋势信息,也可显著减少 DTW 计算的数据维数,达到降低计算复杂度的效果.

3　波动特征点提取

时间序列往往数据量大,甚至可能是实时的数据流,但通常最感兴趣的是其变化过程中的少数关键段或关键点. 例如股票技术分析中的头肩顶形态,可以通过少数几个特征点来表征,如图 2 所示.

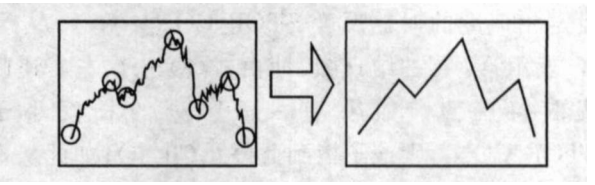

图 2　头肩顶形态及其特征点

本节给出一种基于序列波动情况的特征点提取方法.

定义 3　时间序列 $X=(x_1,x_2,\cdots,x_i,\cdots,x_N)$,定义在某一尺度上 X 的波动特征点为 x_j,且有

$$\{x_j\mid VD(x_j)=\max(VD(x_i)),$$
$$i=1,2,\cdots N\}. \quad (3)$$

其中 $VD(x_i)$ 为 x_i 到序列 X 端点间连线的距离且

$$VD(x_i)=\left|x_i-\left(x_1+(x_N-x_1)\frac{i-1}{N-1}\right)\right|.$$

如图 3 所示,点 P_5 即为该序列特征点. P_5 将序列分为两部分. 一般而言,若特征点 x_j(记为 F_1)将 X 分为前后两段子序列 $X_1(x_1,\cdots,x_m,\cdots,x_{j-1})$ 和 $X_2(x_j,\cdots,x_n,\cdots,x_N)$,则通过递归调用定义 3 分别确定 X_1 和 X_2 的特征点 F_2 和 F_3,依次进行直到子序列长度小于 $\mathrm{STOP\_CON}_1$,从而得到特征点序列 $FS=(F_1,F_2,\cdots)$.

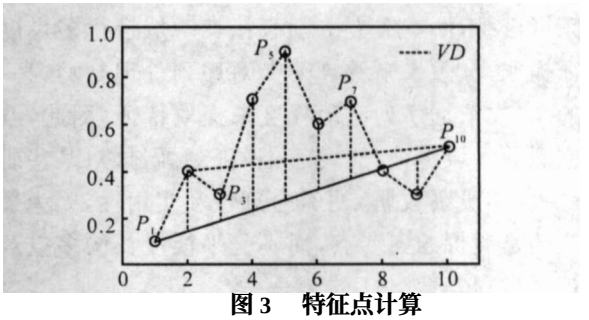

图 3　特征点计算

显然 $\mathrm{STOP\_CON}_1\ge 3$,它表征了特征点的计算粒度. $\mathrm{STOP\_CON}_1$ 的大小可以根据序列长度和要求灵活设定. 本文取 $\mathrm{STOP\_CON}_1=3$,即子序列长度如果小于 3,则停止提取特征点.

易知,上述分层递归提取 FS 的过程也是一个二叉树的构造过程. 因此本文考虑将特征点以二叉树结构实现,便于灵活调整和管理.

定义 4(二叉特征树 FT)　FT 首先是个二叉树,树的结点表示一个特征点 F_k. FT 为空,或者满足:

1) LeftChild 中所有结点的序号都小于根项;

2) RightChild 中所有结点的序号都大于根项;

3) LeftChild 和 RigthChild 均是二叉特征树.

进一步,两条序列 X,Y 的相似度通过计算特征树 $FT(X)$,$FH(Y)$ 上相关序列点近似得到,即

$$D(X,Y) = D(FT(X), FT(Y)). \quad (4)$$

计算相似度(4)时,在 $STOP\_CON_1$ 的基础上设定相似度的对比尺度 $STOP\_CON_2: N_l \geq N_{l+1}$, N_l 表示第 l 层的结点数. 即由式(3)依次左右递归提取特征点,如果某一子序列的长度小于 $STOP\_CON_1$,则该子树到此停止(叶子);如此从根到叶子逐层构造二叉特征树,直到树的结构满足 $STOP\_CON_2$:第 $(l+1)$ 层结点相对于第 l 层并没有增加,二叉特征树已逐渐稀疏,认为序列特征点已基本抽取完毕,无需获取更多的细节信息,提取将截至到第 l 层.

由此可见,$STOP\_CON_1$ 是二叉特征树(除最后一层外)叶子结点的判断条件,而 $STOP\_CON_2$ 则表征了特征点的整体分布情况,是树最后一层的判断条件. 将两个中止条件相结合,既可以有效、灵活控制特征点的提取尺度,又避免树深度过大,显著降低构造过程和后续 DTW 的计算量.

FT 具有以下性质:

1) 先抽取的特征点处于树的根部,体现了序列总体变化趋势;随着层数增加,局部细节信息得以表现. 这种层次结构很自然地体现了多分辨率效果.

基于小波变换 DWT 的方法具有多分辨率特性[10],但本文方法有别于 DWT:DWT 需要变换到频域进行,而本文方法则直接在时域进行多尺度特征提取;从效果来看,DWT 对序列处理是均匀一致的. 本文依据序列波动程度来提取特征,因此,波动剧烈的区域将被"放大"和详细刻画,这对某些研究对象(如股票数据)更具合理性和实用性,因为最关注的是股价的突变点,而不是平淡漫长的多空双方拉锯过程.

2) 由二叉树相关知识推知 FT 构造的复杂度是 $O(\log N)$. 与之对比,DWT 是 $O(N)$.

4 基于层次聚类的实验比较

层次聚类可以考察一条序列同测试集中其他所有序列间的相似距离,以自下而上的方式获得逐层聚类效果,从而全面衡量时间序列相似度的准确性[4,7,9],如图 4 所示.

4.1 数据集

为使实验更具代表性,使用了 UCR 时间序列数据库中 3 个数据集. 采用层次聚类将本文方法(fpDTW)同经典 DTW(tDTW)、改进的 DTW(wDTW[5])在计算效率与准确性方面进行比较,比较结果如表 1 所示.

4.2 实验步骤

为方便起见,记 3 种方法分别为 $M_i, i = 1,2,3$;

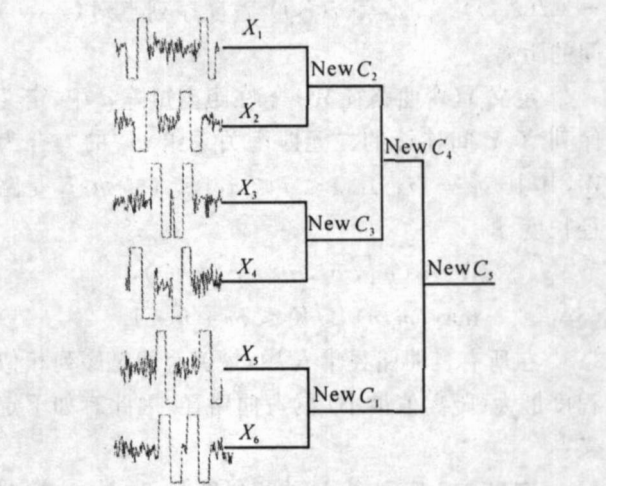

图 4 对 TwoPattern 数据集的层次聚类

表 1 实验采用的数据集

数据集	数据规模	首次提出者
ControlChart(CC)	6 类,每类 100 条,每条 60 点	1999 年 Alcock R J
Trace	4 类,每类 50 条,每条 275 点	2000 年 Roverso D
TwoPattern	4 类,每类 5 000 条,每条 128 点	2002 年 Geurts P

C_{jk} 为数据集 j 的第 k 类; x_m, x_n 为 C_{jk} 中任意两条不同序列,$1 \leq m, n \leq |C_{jk}|$,$|C_{jk}|$ 为第 k 类序列的长度; S 为层次聚类合并时的原序列对(图 4 中序列 X_1 和 X_2); P 为层次聚类的准确率.

对方法 M_i 和数据集 C_j 有如下步骤:

Step1: 相似度距离. 计算 x_m, x_n 间距离 $d\langle x_m, x_n\rangle$,计算规模 C_N^2 次,其中 N 是序列总条数. 例如对于 ControlChart 数据集,$N = 6 \times 100$,由此得到 C_j 序列间的距离矩阵 D_{M_i,C_j}(上三角型矩阵),计算时间记为 T_{M_i,C_j}.

Step2: 组平均连接层次聚类. 设第 i 次聚类时(对序列 x_m, x_n)得到某新类序号为 New $C(N+i)$,准确性为

$$z_i(x_m, x_n) = \begin{cases} 0, (x_m, x_n) \in C_j; \\ 1, (x_m, x_n) \notin C_j. \end{cases}$$

从 S 中删除已发生合并的源序列 x_m, x_n,用新类 New $(N+i)$ 代替.

$z_i = 1$ 时,记新类 New $C(N+i) \in C_j$;否则记为 -1(无效类),且后续任何对无效类产生了聚类的结果都记为 -1,z 记为 0.

更新 New $C(N+i)$ 与 S 中其他序列间的距离,即

$$d\langle x_{N+1}, \cdot\rangle \frac{d\langle x_m, \cdot\rangle + d\langle x_n, \cdot\rangle}{2}.$$

重复 Step2,直到 $S = \varnothing$.

Step3: 聚类准确率统计.

$$P = \frac{\text{实际合并的正确次数}}{\text{理论合并正确次数}} = \frac{\sum_{i}^{M} z_i}{N - 1}.$$

4.3　实验结果

如 4.1 节所述,对 DTW 设置计算窗口可以加快计算速度,但难于选择合适的窗口大小.通常窗口越大,准确率越接近经典 DTW;窗口越小则速度越快.但文献[5]通过大量实验分析指出,一般只需较小的计算窗口即可得到相对较高的准确率,因此对 wDTW 采取相对折衷的方案,选择窗口大小为序列长度的 20%.本文 fpDTW 的中止条件分别为 $STOP\_CON_1 = 3$,$STOP\_CON_2$ 存在 $N_l \geq N_{l+1}$.

对每种方法、每类数据集分别进行 10 次实验,统计平均性能如表 2 和表 3 所示.

表 2　准确率对比　%

	tDTW	wDTW	fpDTW
CC	97.9	96.2	97.5
Trace	100	98.7	100
TwoPattern	96.7	71.9	97.2

表 3　计算时间对比　s

	tDTW	wDTW	fpDTW
CC	1 178.5	381.5	268.2
Trace	10 757.5	1 656.8	256.2
TwoPattern	2 334.6	598.9	167.7

从结果来看,本文方法在 3 种数据集上均取得了较高的聚类准确率,优于 wDTW.尤其是在 Trace 数据集准确率达到了 100%,而对 TwoPattern 数据集则超过了不加约束的 tDTW.时间方面,tDTW 耗时最多,wDTW 由于只计算了有效窗内的序列,有效减小了时间,本文 fpDTW 方法则通过 $STOP\_CON_1$ 和 $STOP\_CON_2$ 两个中止条件进一步提高了计算效率,对 Trace 数据集可以降低为 tDTW 的 1/42(而准确率相同),从而提高了 DTW 的实用价值.

5　结　　语

时间序列存在大量冗余信息,这些冗余信息加剧了相似性搜索的计算代价.基于特征点的实际意义,本文采用一种简单易行的特征点定义方法,以二叉树结构进行层次性地提取,并给出两个中止条件.方法既保留了序列变化中的趋势信息又有效约减了数据维数,可以显著改善 DTW 的计算速度.

参考文献(References)

[1] Debregeas A, Hebrail G. Interactive nterpretation of kohonen maps applied to curves [C]. Proc of 4th KDDACM. MenloPark, CA: AAAI Press, 1998: 179-183.

[2] Ehud Gudes, Litvak Marina. Discovering target events rules based on time-consecutive pattern mining[C]. The 4th ICDM'04 Workshop on Temporal Mining. Brighton, 2004.

[3] 李爱国, 覃征. 在线分割时间序列数据[J]. 软件学报, 2004, 15(11): 1671-1679.
(Li A G, Qin Z. On-line segmentation of time-series data[J]. J of Software, 2004, 15(11): 1671-1679.)

[4] Eamonn Keogh. Data mining and machine learning in time series databases[C]. Proc of the 4th IEEE Int Conf on Data Mining. Seattle, 2004.

[5] 肖辉,胡运发. 基于分段时间弯曲距离的时间序列挖掘[J]. 计算机研究与发展, 2005, 42(1): 72-78.
(Xiao H, Hu Y F. Data mining based on segmented time warping distance in time series database[J]. J of Computer Research and Development, 2005, 42(1): 72-78.)

[6] Chorirat A R, Eamonn K. Making time-series classification more accurate using learned constraints [C]. Proc of SIAM Int Conf on Data Mining. Florida, 2004: 11-22.

[7] Berndt D, Clifford J. Using dynamic time warping to find patterns in time series[C]. AAAI-94 Workshop on Knowledge Discovery in Databases. Seattle, 1994.

[8] Yi B K, Jagadish H V, Christos Faloutsos. Efficient retrieval of similar time sequences under time warping [C]. Proc of the 14th IEEE Int Conf on Data Engineering. Orlando, 1998: 201-208.

[9] 翁颖钧, 朱仲英. 基于动态时间弯曲的时序数据聚类算法的研究[J]. 计算机仿真, 2004, 21(3): 37-40.
(Weng Y J, Zhu Z Y. Novel algorithm for time series data mining based on dynamic time warping [J]. Computer Simulation, 2004, 21(3): 37-40.)

[10] 郑诚,蔡庆生. 一种多尺度的时间序列相似模式匹配算法[J]. 小型微型计算机系统, 2003, 24(3): 546-549.
(Zheng C, Cai Q S. A multi-scale similar pattern match approach for time series databases[J]. Mini-Micro Systems, 2003, 24(3): 546-549.)