高效时序相似搜索技术

冯玉才 蒋 涛 李国徽 朱 虹
(华中科技大学计算机科技与技术学院 武汉 430074)

摘 要 时序相似搜索被认为是将来最有前途的技术之一.然而,时序数据是典型的高维海量数据,如何开发高效算法非常关键.文中概述了时序相似搜索技术的研究现状和进展以及研究的主要内容,讨论了该技术的几个重要应用范例,并对一些典型算法进行了定量分析;然后重点论述了高效时序相似搜索的关键技术,包括边界过滤、三角不等式修剪、多辨析率检索方法、过滤精炼方案等.最后讨论并分析了时序的近似相似搜索技术.上述所有技术通过对比,其正面和反面都被深入分析.最后指出了存在的问题和未来的研究热点和方向.

关键词 时间序列;相似搜索;高效搜索方法;子时间序列
中图法分类号 TP311 **DOI 号**: 10.3724/SP.J.1016.2009.02107

Underlying Techniques of Efficient Similarity Search on Time Series

FENG Yu-Cai JIANG Tao LI Guo-Hui ZHU Hong
(*College of Computer Science & Technology, Huazhong University of Science and Technology, Wuhan* 430074)

Abstract Time series similarity search is regarded as one of the most promising technologies in the future. However, time series data is a typical high dimensional and massive data. Developing efficient algorithms is very important for fast time series similarity queries. The paper provides an overview of research progress, and gives main research content and directions in the field. Then, some paradigms in time series applications are introduced and the performance of some typical algorithms is analyzed quantitatively. Next, this paper surveys the underlying technologies of efficient similarity queries on time series, such as bounding filtering, triangle inequality pruning, multi-resolution approach, and filter-refine scheme, etc. Furthermore, the main methods for approximate similarity search are summarized and analyzed. All above-mentioned technologies, the pros and cons of the techniques are discussed by comparison. Finally, some possible research hotspot and directions in the future are given.

Keywords time series; similarity search; efficient searching methods; subsequence

1 引 言

时序数据在医学、金融、传感器网络、移动对象、图像、音频等领域广泛存在,同时它已经在生物序列分析[1]、金融数据分析[2-3]、移动对象跟踪[4-5]、传感器网络监控[6]、运动捕获[7]等领域成功应用.由于时序相似搜索技术存在巨大的潜在应用价值,它一直是学术界研究的热点.许多研究机构和一些著名的大学纷纷参与进来,包括 IBM 公司的 Almaden 和 Watson 研究中心、Maryland 大学、Carlifornia University 的一些研究小组(Irvine、Riverside 和

收稿日期:2007-12-20;最终修改稿收到日期:2009-01-03.本课题得到国家"八六三"高技术研究发展计划项目基金(2007AA01Z309,2006AA01Z430)、国土资源部三峡库区三期地质灾害防治重大科研专项基金(SXKY3-6-3)资助.**冯玉才**,男,1946 年生,教授,博士生导师,主要研究领域为数据库技术.**蒋 涛**,男,1973 年生,博士,研究方向为数据挖掘.E-mail: jiangtao_albert@yahoo.cn.**李国徽**,男,1973 年生,博士,教授,博士生导师,主要研究方向为移动时空数据库.**朱 虹**,女,1965 年生,博士,教授,博士生导师,主要研究领域为数据库安全.

Santa Barbara 等)以及 Carnegie Mellon 大学等.我国的复旦大学、浙江大学、南京大学和中国科学技术大学以及香港科技大学、香港城市大学等也参与研究.近十年来著名的国际学术会议如 SIGMOD、VLDB、PODS、ICDE 以及期刊如《ACM TODS》、《IEEE TKDE》、《VLDB Journal》等都呈现了大量高水平的研究成果.

时序相似搜索技术已经从早期的一般化研究阶段,即研究时序度量、时序维度约间和时序索引等方面,进入到深入的研究阶段,即如何针对不同的应用领域开发高效的时序搜索算法,同时保持高的精度并维持低的时间和空间成本.然而,时序是典型的高维、海量数据类型.开发高效的时序相似搜索技术仍然面临极大的挑战,很多问题有待解决,具有广阔的研究空间,我们认为研究高效时序相似搜索的核心支撑技术具有重要的意义.本文主要从高效性这个角度来阐述时序相似搜索的核心技术,及应遵循的基本技术框架和解决思路.

本文首先概述时序相似搜索的基本概念及其研究的主要内容,然后分析了它的应用场景并就有关典型算法的性能进行了定量分析,接着综述了高效时序相似搜索技术的核心支撑技术以及近似的时序搜索技术,最后总结全文,指出时序相似搜索技术可能的研究热点和方向.

2 时序相似搜索基本概念及研究内容

2.1 时序及时序相似搜索

时间序列是指随着时间变化而形成的有序数据列表,简称时序.它反映了某个事务/事件随着时间变化的状态,其状态可以用实数值或符号来表示.通常提到的时序是指通过等间隔时间取样形成的具有实数值的有序数据序列,也即 Time Series,例如,股票价格变化序列,其定义如下.

定义 1. 时间序列(Time Series).时间序列 S 是指按时间顺序排列的,具有相等时间间隔的实数数据列表,记为 $S(=\{s_1,s_2,\cdots,s_n\})$.其中时间序列长度为组成 S 的实数值个数,记为 $|S|=n$.包括现实世界对象或事件通过某种映射转换而来的时间序列,例如,图像形状映射的时序、英文手迹映射的时序等.

时序数据通常存储在文件中,以数据库文件形式存放,这种数据库称为时间序列数据库 TSDB.文献[8]最早提出时序相似搜索问题,它是指在 TSDB 中寻找与查询时序 Q 具有相似特征的数据序列 R,其定义如下.

定义 2. 时间序列相似(Time Series Similarity).给定一个查询序列 $Q(=\{q_1,q_2,\cdots,q_m\})$,一个数据序列 $S(=\{s_1,s_2,\cdots,s_n\})$,如果序列 Q 和序列 S 满足 $dist(Q,S)\leq\varepsilon$,则说时间序列 S 和 Q 是相似的.其中,ε 是时序相似门限值,$dist(Q,S)$ 是一个距离函数,例如,$L_p(1\leq p\leq\infty,p\in\boldsymbol{N})$ 距离函数[8]或动态时间弯曲(Dynamic Time Warping,DTW)[9]距离函数.

L_p 距离函数主要包括 3 种形式:当 $p=1$ 时,L_1 表示 Manhattan 距离,它实际是两时序所有点对差值绝对值的累积和,具有单调递增性;当 $p=2$ 时,L_2 称为欧氏距离;当 $p=\infty$ 时,L_∞ 称为最大距离,其形式变为:$L_\infty=\max_0^{n-1}(|x_i-y_i|)$,表示两时序所有点对差值绝对值的最大值.$L_p$ 距离函数是度量函数,因为它满足度量空间的 3 个性质:(1) 自反性.$d(x,y)=0\Leftrightarrow x=y$;(2) 对称性.$d(x,y)=d(y,x)$;(3) 三角不等式.$|d(x,y)-d(y,z)|\leq d(x,z)\leq d(x,y)+d(y,z)$.这 3 个性质也是度量空间的 3 个充分必要条件.时序距离函数满足距离的度量性质具有非常重要的意义:(1) 可以利用三角不等式修剪搜索空间以加快搜索效率;(2) 时序聚类算法要求距离函数具有对称性;(3) 它也是度量空间搜索策略(例如深度优先)能够正确执行的必要条件.而源于语音识别的 DTW 采用动态规划的思想递归定义,其形式为

$$\begin{cases} D_{tw}(\langle\rangle,\langle\rangle)=0, D_{tw}(S,\langle\rangle)=D_{tw}(\langle\rangle,Q)=\infty, \\ D_{tw}(S,Q)=(|L_p(First(S),First(Q))|^p+ \\ \quad |\min\{D_{tw}(S,Rest(Q)),D_{tw}(Q,Rest(S)), \\ \quad D_{tw}(Rest(S),Rest(Q))\}|^p)^{1/p}, \end{cases}$$

其中 $\langle\rangle$ 表示空时序,$First(S)$ 表示时序 S 的第一个元素,$Rest(S)$ 表示时序 S 从第 2 个位置到最后的子时序,$First(Q)$ 和 $Rest(Q)$ 可以类似解释,$\min()$ 是求 3 个元素中的最小值函数.DTW 的缺点在于:它不是度量函数,且直接实现的算法具有较低的效率.

DTW 函数的优点在于:去掉了数据时序 S(长度 n)和查询时序 Q(长度 m)保持等长的要求(总长 L 满足条件:$m,n\leq L\leq(m+n)$),容许序列点自我复制后再进行等长匹配,从而具有较高精度.运用 DTW 时,须建立一个 $m\times n$ 的累积矩阵 $\boldsymbol{M}$ 以存放动态时间弯曲距离,运算从 $\boldsymbol{M}[1][1]$ 开始至 $\boldsymbol{M}[m][n]$ 结束,最终在 $\boldsymbol{M}$ 上经过的路径称为动态时间弯曲路径.L_p 距离函数与其不同,它要求两个

时序的点之间一一对应匹配(等长匹配),相对于DTW的$O(mn)$时间效率其效率为$O(n)$.

时序相似性搜索可以分成全序列匹配和子序列匹配两类.文献[10]最早提出子序列相似搜索技术,该技术也称为通用多维索引框架GEMINI(GEneric Multimedia INdexIng),有的文献也称为FRM.其主要思想是:首先使用一个尺寸w的滑动窗口在长$Len(S)$的数据序列S的每个可能偏移位置,将数据序列S分成$Len(S)-w+1$个子数据序列,同时通过DFT将它们映射成特征空间中的特征点,再建立以包含一定特征点的最小边界矩阵(Minimal Bounding Rectangle,MBR)为索引节点的R*-tree[11]索引结构(称为ST-index),类似地处理查询序列Q;然后构造一个查询序列Q和数据序列R的MBR的范围搜索获得候选项;最后使用一个后处理过程去掉多报的数据子序列.从搜索方式来看,FRM属于ε邻域范围搜索,而如果对ε值进行从小到大排序,取最小ε值的搜索则称为最近邻(1NN)搜索,取前k个搜索结果的为kNN(k近邻)搜索.

2.2　时序相似搜索应考虑的因素

从需求层面来看,一方面应该保证时序相似搜索技术具有足够快、正确性、小的空间负荷、动态性及能够处理变长的序列搜索等特性[8].足够快是指搜索的时间效率高,当前的许多搜索算法的效率有待提高;正确性即指应该返回满足要求的合格时序,而不能丢失任何符合条件的时序,例如,不应出现漏报(false dismissals或false negatives)现象,但多报(false alarms或false positives)是容许的,通常添加一个后处理(post processing)过程可以去除多报;动态性指索引除了可以供搜索外,还可在其上删除、插入或追加时序.另一方面,我们还应该考虑不同领域时序的本身特性,即需要考虑噪声以及时序的各种变形[2]对时序相似搜索技术的影响.例如,L_p度量方法就对噪声非常敏感[12].

从技术层面来看,应考虑时序相似搜索技术需要解决的关键问题,这包括时序相似度量、时序转换、高效的时序索引等.这些构成了高效的时序相似搜索技术基础,也是一般化研究阶段的主要内容.之后,时序相似搜索技术逐渐向深度和广度转移,也即第二代时序相似搜索技术.

(1)时序相似度量

时序相似度量是高效时序相似搜索技术的基础.建立何种度量函数来实现时序相似度量非常关键,这里不但要考虑各种度量函数的特性,还应该考虑具体应用领域的实际需求.相似度量一般可分为基于形状的相似度量[8-9]、基于特征的相似度量[3]、基于模型的相似度量[2,13]以及基于数据压缩的相似度量几种情形.已有的度量函数主要包括L_p-norms[8]、DTW[9]、最长公共字串(Longest Common Subsequence,LCSS)[4]、实序列编辑距离(Edit Distance on Real Sequence,EDR)[5]和实补偿编辑距离(Edit Distance with Real Penalty,ERP)[14]、空间装配距离(Spatial Assemble Distance,SpADe)[13]等.随着研究的深入必将出现新的时序相似度量方法.表1比较了几种距离函数的特性.

表1　距离函数对比

函数名称	运算成本	度量函数	支持平移	支持噪声	特性
L_p-norms	$O(n)$	√			简单高效,不支持平移,时间和幅度缩放,易受噪声影响,精度差
DTW	$O(n^2)$		√		高精度,支持平移、非等长匹配
LCSS	$O(n^2)$		√	√	用于移动轨迹匹配,对异常和噪声有较强适应能力
EDR	$O(n^2)$		√		相对LCSS具有更强的Robust
ERP	$O(n^2)$	√	√		可利用三角不等式,支持偏移(综合了L_p和DTW的优点)
SpADe	—		√	√	适于形状模式匹配,支持时间和幅度的偏移、缩放以及噪声

(2)时序转换

由于时序是典型的高维数据,为了避免由于高维(大于16维)而引起相似搜索算法性能急剧下降,即所谓的维度灾难(dimension curse).时序降维也称为时序转换,是一类相对比较成熟的技术,主要包括:离散傅里叶变换(Discrete Fourier Transform,DFT)[8]、离散小波变换(Discrete Wavelet Transform,DWT)[15]、主成分分析(Principle Component Analysis,PCA)、奇异值分解(Singular Value Decomposition,SVD)[16]、点对线性近似(Piecewise Linear Approximation,PLA)[17]、点对累积近似(Piecewise Aggregate Approximation,PAA)[18]、自适应累积常量近似(Adaptive Aggregate Constant Approximation,APCA)[19]、切比雪夫多项式(Chebyshev Polynomials,CP)[20]、符号累积近似(Symbolic Aggregate approximation,SAX)[21]和界标模型(Landmarks)[2]等.降维方法要求能保留大部分时序特征信息,同时满足降维下界定理,即特征

空间距离小于源空间距离：$D_{\text{index}}(S,Q)\leq D_{\text{src}}(S,Q)$[10]，这是保证无漏报的必要条件.但不管哪种技术和方法，都应考虑下面的因素：是否能够有效地降低数据的维度；尽量保留原时间序列中的信息；是否有利于索引的建立以及在索引中的搜索；是否剔除噪声和冗余以利于算法的效率和精度；是否支持变长时间序列；是否采用符号表示法等.

(3)时序索引技术

由于时序广泛存在现实世界的各个领域，同时许多领域数据也可以转换成时序数据，因而时序是典型的海量数据类型.为了提高时序搜索效率，索引是非常必要的搜索机制.索引技术的关键问题是如何划分数据空间或向量空间以及如何根据划分方法将数据组织起来.最早的基于向量空间索引技术是1984年由Guttman提出的R-tree，后来出现了一些变形种类：R*-tree[11]、SS-tree以及SR-tree等.另一种类型是基于度量空间特征值聚类的索引方法：VP-tree[22]、MVP-tree、M-tree[23]、SA-tree[24]等.另外，Park等人在基于字符的后缀索引[25]和前缀索引[26]方面也作了大量工作.

2.3 时序相似搜索研究的主要内容

从数据挖掘的角度来看，时序模式匹配[27-28]、时序关联分析[29-30]、时序聚类[31]、时序分类以及时序异常[32]是时序相似搜索技术的主要研究任务.就时序模式匹配而言，特定模式匹配[28]、聚集模式(Burst)查询与发现[6]、周期模式发现[33]、主题模式(Motif)发现[34]、异常模式(Discord)发现[32]是其主要研究内容；从关联分析来看，局部相关或整体相关是主要研究角度，同步相关[29]、滞后相关[30]以及突发相关是主要研究类型.如何定义局部模式并指定模式窗口的大小、基于序列的相关(使用皮尔森Pearson系数)还是序列提取的特征的相关(须定义关联函数)、采用何种特征提取方法(例如，SVD、DFT)、建立何种索引和算法来高效报告相关等都是时序关联分析的重要研究方向；就时序聚类而言，由于时序维度高，时序空间中的点将变得非常稀疏而难以具备聚集特性，因此时序聚类仍面临很大挑战.如何提取时序特征(feature extraction)、如何基于子空间(或投影)进行聚类、如何基于网格进行聚类、如何采用监督或约束的方法进行聚类、在分层聚类中聚类合并和分裂策略、如何定义聚类的密度函数等都是时序聚类的重要研究内容；在时序分类中，如何利用已有的机器学习分类方法，例如，kNN分类、神经网络、多分类系统等是须考虑的重要问题.目前，时序分类研究相对较少，在这方面仍有待进一步的深入.在时序异常分析中，基于距离的时序异常检测方法是研究的热点.另外，基于局部密度的时序异常挖掘算法和基于聚类的时序异常检测算法也是研究的重要方向.在这个方面，如何基于索引并考虑页面的I/O效率检测异常时序、如何定义对象在局部空间或聚类中的背离函数、如何定义网格并对网格进行合理分区以及如何综合距离、密度、聚类的异常检测方法都是重要的研究要点.

从维度方面来看，基于空间的多维时序挖掘是研究的热点，这包括移动对象跟踪[4]、运动捕获[7]等领域.开发能适应噪声环境及各种变形的时序距离函数是首要解决的问题，建立基于向量空间时序索引的搜索算法是重要的研究方向.从现实应用层面来看，相对于静态时序来说，数据流、传感器网络将是时序新的应用环境.目前，基于数据流环境下的时序搜索技术是仍有待进一步研究的重要方向之一，这其中包括高效的流时序模式匹配[28]、流时序关联分析(Correlation)[29-30]、流时序分段、流时序摘要(Summarization)、流时序聚类、流时序异常分析等方面.由于数据流具有高速、不可存储的特性，具有高效、一次扫描(One-Pass)和增量计算特征[30,35]的算法是研究的热点.在传感器网络环境中，数据点可能丢失、延迟，传感器有限的电源和处理特性，以及传感器数据非集中分布的特点，使得高效而具有高精度的算法具有重要的应用价值.如何建立正确的模型以预测或填充迷失的数据是重要的研究课题，如何开发分布式时序处理算法非常重要.

从时序研究的效率和精度方面来看，具有较高复杂度的时序算法一直是研究的热点.例如，kNN算法、时序关联分析、时序主题发现、异常时序搜索、子时间序列搜索等都具有$O(n^2)$的复杂度，且常是重点研究对象.由于时序高维度和大规模数据量的特性，通常的计算一般须较长的时间，目前许多研究者提出近似计算和anytime特性(在很短的时间可获得大部分的精度，而且计算可随时终止；然后，可从中断点恢复计算，且随计算时间的增加可获得更高的精度，并最终获得精确的结果而退出)计算的思想是在效率和精度间达到平衡的新的思路.基于近似的相似搜索技术已经成为一个重要的分支，也是最近几年研究的热点.从实际的需求方面来看，成本敏感(Cost Sensitive)查询、噪声敏感(Noise Sensitive)查询、查询敏感(Query Sensitive)搜索、错误边界(Error Bound)控制查询、概率查询(Probabilistic

Query)、近似查询(Approximate Query)、分布式查询(Distributed Query)将成为新的研究方向和热点.

3　高效时序相似搜索技术的应用

3.1　主要应用领域介绍与分析

高效的时序相似搜索技术已经在医学、网络流量、移动对象轨迹、生物序列、金融、音乐、天文观测、传感器网络等诸多领域的序列数据处理方面成功应用.而且随着更多领域序列数据的出现,新的应用领域将不断被开发出来.随着不同领域数据量的急剧增加,对设计的算法将提出更高的要求.本文总结了时序几个主要的应用领域如下:

(1)生物序列数据(例如,DNA 序列)分析是当前生物信息学(Bioinformatics)的重要研究领域[1,33].尽管出现了许多研究成果,但随着生物序列数据的急剧增加,该领域仍存在广泛的研究前景.

(2)移动对象跟踪与识别在许多场景下具有重要的应用价值[4-5].例如,在辽阔的草原上,借助远程传感器网络,可以通过动物迁移路线挖掘来发现某些类型动物的迁移模式;在运动领域,可以通过对优秀运动的运动轨迹进行挖掘,发现其有价值运动模式;在大型超市监控视频中,通过视频顾客运动轨迹挖掘,以辅助商品的摆放;银行监视系统,通过对顾客运行轨迹挖掘,发现可疑的运动模式,以辅助报警系统报警等.这里应用的主要技术为多维的空间或时空度量方法,例如 LCSS、EDR 等.

(3)基于医学序列数据(例如,ECG、EEG)的相似搜索技术能够为医生提供重要的医学信息[28,36],以发现某些异常的病例,或辅助他们进行病例诊断;基于音乐数据的相似匹配可以作为重要的工具,以对音乐进行分类,识别或搜索出同类的音乐[37].

(4)基于数据流方式的监控在网络、金融、天文等领域已经深入应用.例如,基于网络流量监控可以实时发现可疑的流量模式,进而提高网络的安全管理;基于金融数据流监控、分析可以提供重要的交易参考价值[29];基于天文流序列(如 gamma 射线、太阳黑子 sunshots 等)的监控能实时监测一些重要的天文现象[6,28,30].

(5)基于传感器网络获得的序列数据处理是正在快速发展的领域.由于传感器可以连续、自动地获得某些应用场景下的序列数据(如温度、湿度、风力、水文信息)[30,38],结合传感器网络技术的时序相似搜索技术正在不断开发、应用.由于传感器网络能够应用的场景非常广阔,使得时序相似搜索技术在这个领域具有广阔的应用空间.

(6)基于图像的形状数据处理在考古学、法律诉讼、历史手迹(manuscript)、医学、运动捕获、机器人以及气象学等领域具有重要的应用价值.例如,发现考古学的文化迁移现象;在法律诉讼中为真假图像的辨别提供帮助;在历史手迹中识别类似的手迹或对他们进行分类等.

(7)基于时序相似模型的数据预测也是一个重要的应用分支.例如,在医学领域的放射治疗中,可以针对患者呼吸运动的相似模型,预测病人的病变位置;在数据流环境下,时序数据可能由于网络拥塞或其它原因而滞后到达,可以基于数据流序列的相似模型预测迷失的数据点.

3.2　不同领域典型算法介绍及性能比较

为了比较不同领域不同算法的性能,下面就一些典型算法使用的数据集和效率进行介绍和讨论.

在关联分析算法中,StatStream 算法[29]和 BRAID[30]是典型的代表.StatStream 算法使用 DFT 约简维度,建立基于约简维度 k 的球形网格投影方法来寻找近邻.扫描方法时间与数据流数目 N_s 平方成比例,而 StatStream 系统的时间主要依赖网格的计算时间,它与网格单元平均流数目 N_{gs} 的平方成比例,显然 $N_{gs} \ll N_s$,从而 StatStream 计算成本大大减少.在基于秒记录的 500G 美国股票交易数据的实验中,它能在 150s 的时间内报告 10000 个以上数据流的相关,而扫描方法最多只能报告 700 个左右.BRAID 算法的主要思想是分解相关系数公式成增量计算形式和多辨析过滤策略(看 4.2.2 小节).BRAID 算法在 25900 至 100000 长的太阳黑子(Sunspots)、地震记录(Kursk)和尖峰序列(SpikeTrains)的实验中,它最大的错误率大约 1%,能够检测半无限长的滞后相关.在极端长度(例如 10^7)的序列中,它相对直接扫描实现方法(Naive)最高快 40000 倍;因为 Naive 的时间和空间复杂度为 $O(n)$,而其空间复杂度为 $O(\log n)$,更新统计值仅须 $O(1)$ 时间,修改窗口信息值只须 $O(\log n)$ 时间.StatStream 和 BRAID 的共同缺点在于难于选择一个合适的窗口尺寸以及不适应非等长的相关性监测.

在特定模式匹配中,FTW 算法[38]和 SPRING 算法[28]非常典型而具代表性.FTW 算法使用了低边界过滤、多辨析修剪和尽早终止距离计算 3 个关键技术(看第 4 节),在长度从 512 到 2048 尺寸为

25000 至 100000 的静态数据集：来自传感器的温度序列(Temperature)、股票金融时序 FinTime 以及自动生成的随机漫步序列(RandomWalk)测试中，相对于文献[37]中的 LB_PAA 方法最高要快 222 倍，而变化数据集尺寸、序列长度以及弯曲程度，其计算时间只有稍微的变化. SPRING 算法的主要思想为：对长度为 m 的查询序列 Q 填充一个特别符号(例如，“*”号)，并使用一个弯曲矩阵 STWM[28]，以记录长度为 n 的序列 R 的每个候选子序列的起始位置和累积距离值，从而极大减少了 DTW 计算的成本. 报告范围查询模式的直接实现方法(Naive)需要 $O(n)$ 大小的矩阵和每个滴答 $O(mn)$ 次更新；然而 SPRING 报告最好子序列仅须 $O(m^2)$ 大小矩阵和每个滴答 $O(m)$ 次更新. SPRING 使用包括文献[30]的 Kursk、Sunpots 和文献[38]的 Temperature 等数据集，在使用长度为 256 的查询序列搜索时，相对 Naive 最快达到 650000 倍.

在异常时序匹配中，文献[32]提出的直接序列扫描方法是典型范例. 直接的异常时序搜索算法(Naive)通过两两比较对象间距离从而需要 $O(n^2)$ 次计算，而文献[32]通过下面 3 个方法：基于最近邻距离分布以确定异常门限值 ε 的启发式策略、Filter-Refine 过滤策略和尽早终止距离计算，大大降低了搜索成本. 在序列长度为 512 共 10^6 条 RandomWalk 时序(3.57GB 磁盘空间)的测试实验中，该算法仅仅花费 27min 的 I/O 时间和 14min 的 CPU 计算时间；而在另一个长度为 140 由运动捕获、EEG 记录和气象记录数据合成的 1.2×10^6 条序列的数据集(1.17GB 磁盘空间)的测试实验中，Filter 阶段和 Refine 阶段的时间分别为 15min 和 16min.

4 高效时序相似搜索关键技术

从技术层面来分析，为了提高时序相似搜索技术的效率，一方面可通过修剪搜索空间来实现，它即通过已计算出的距离信息来约简不必要的距离计算，从而减少计算成本，这里的技术基础是三角不等式和度量空间索引方法(例如，M-tree). 它们须配合启发式的搜索机制，例如，分支界限法(Branch and Bound)[39]、深度优先(Depth First，DF)[39]、最好优先(Best First，BF)[40]等一起使用. 常见的修剪策略包括：三角不等式过滤、过滤精炼(Filter-Refine)策略(或多步过滤策略)等. 另一方面可通过提高距离计算的效率来达到，有时须辅助向量空间索引方法(例如，R-tree)一起实现，包括以下几种方法：(1)转换原始空间 $\mathcal{O}$ 到距离计算成本远小的特征空间 $\mathcal{F}$ 中. 它要求特征空间具有”收缩”特性，即特征空间的对象间距离小于原始空间的距离，以确保不会产生漏报.(2)多辨析计算方案(Multi-resolution)，它要求能寻找一个合适的多辨析函数和结构；(3)边界过滤方法. 它要求寻找一个计算成本远低于原始距离函数的边界函数(包括低边界函数和上边界函数)；(4)尽早终止计算算法(Early Stopping)；(5)高效的距离计算算法，例如，FastDTW[41]、FTW[38]以及 SPRING[28]等算法. 另外，高效的子序列搜索算法也是须重点考虑的问题，例如，Dual-Match[42]、General-Match[43]算法等. 在上述方法中，索引方法和时序转换是讨论较多且较成熟的技术，本文不作讨论.

4.1 修剪搜索空间

4.1.1 问题描述

修剪搜索空间通过减少距离比较次数来降低计算成本. 它一般按照距离大小将时序对象划分成具有明显层次特征和聚类特性的度量空间索引结构，然后通过三角不等式过滤来修剪不必要的索引分支或对象. 例如，对于 3 个对象 x,y,z，如果它们的距离函数 d 满足三角不等式且距离 $d(x,y)$ 和 $d(y,z)$ 都已经计算出来，那么则可利用 x 和 z 满足 $|d(x,y)-d(y,z)|\leq d(x,z)\leq d(x,y)+d(y,z)$ 这样上下边界的信息，并结合距离门限值(例如，ε)来判断是否需要真正计算距离 $d(x,z)$，以减少距离计算成本.

事实上，修剪搜索空间的方法已经在时序的范围查询[8]、k 近邻查询(kNN)[12]广泛运用. 另外，可逆近邻(RkNN)查询[44]以及 Skyline 查询[45]也可运用. RNN 的重要作用是识别查询对象 q 对其它对象的影响力. 对给定一集合 D 和一查询对象 q，可逆近邻查询 $RNN(q)$ 检索出所有以 q 作为其近邻 NN 的对象 $o\in D$. RkNN 是 RNN 的推广，它检索出以 q 作为其 k 个近邻的所有对象. 形式化地，$RkNN(q)=\{p\in D|q\in NN_k(p)\}$，$NN_k(p)\subseteq D$ 对应 q 的 k 个近邻集合. 目前，它在时序中仍然是一个未研究的课题，我们正在进行这方面研究. Skyline 查询从 2001 年提出以来一直是研究的热点. 给定一个 d 维的数据集 D，一个 Skyline 查询返回一个子集，该子集中的任意一对象都不被 D 中其它对象所控制. 所谓控制关系是指给定 d 维空间中的多个对象，如果对象 p 至少在某一维上优于另一个对象 q，而在其它的

维度上都不比对象 q 差(p 优于或等于 q),则说 p 控制 q. 尽管,Skyline 点不同于时序,但高维的 Skyline 点可当作时序对象对待. 基于高维的 Skyline 查询算法是重要的研究方向之一.

4.1.2　研究概述与分析

(1) 三角不等式修剪方法

从三角不等式修剪方法的使用来看,一方面可以通过预计算方法来实现,例如,文献[14]的 kNN 计算. 假定对象 Q 与对象 $R_1,R_2,\cdots,R_m$ 的距离已知,且 $R_1,R_2,\cdots,R_m$ 和当前对象 S 两两之间的距离也已预计算,现在要计算当前对象 S 是否为对象 Q 的 kNN 时,就可利用三角不等式修剪方法. 具体过程如下:首先求出对象 Q 与 R_i 的距离与 R_i 与 S 的距离之差的最大值($1\leq i\leq m$),即最大修剪距离 $maxPruneDist=\max\limits_{1\leq i\leq m}\{dist(Q,R_i)-dist(R_i,S)\}$;然后将当前的 kNN 距离 $sofarDist[k]$ 与 $maxPruneDist$ 比较,如果 $maxPruneDist>sofarDist[k]$,那么对象 S 即可修剪掉.

另一方面可以通过度量索引来实现. 因为,索引构造过程将会把对象间的距离信息计算好,并存于索引当中,从而在查询时可直接利用它们消除冗余计算. 这里的对象间距离一般是指一个聚类 C 的其它对象 O_i 相对于聚类中心对象 O_c(或枢轴 pivot 对象,即参考点)的距离. 因而,开发好的基于度量空间的聚类算法也是重要的研究课题. 事实上早期的度量空间索引方法都利用了三角不等式修剪方法,包括 VP-tree、M-tree 和 Sa-tree 等. 文献[23]Ciaccia 等人提出的 M-tree 是运用三角不等式修剪方法非常典型的多维索引例子,它在范围查询和 kNN 查询中都可使用,能修剪相应的索引分支和索引对象以消除大量的冗余计算. 但是应注意到不同的搜索策略将会极大地影响算法的效率. 比较成熟的搜索策略包括:分支界限法、DF、BF 等,而在 kNN 算法中通常可使用一个保存了 k 个对象距离(相对于查询对象 q)升序排列的优先队列(prior queue)以提高搜索效率. 我们认为这些搜索策略也是开发新的高效算法应遵循的基本框架. 文献[45]中的 Chen 等人基于 M-tree 将三角不等式修剪策略引入到动态的 Skyline 查询,提出了度量空间动态 Skyline 查询方法. 其修剪策略概括为:给定枢轴点为 p 的对象集合 S 及查询对象 q,则对象 $o_i\in S$ 和 q 之间的距离下界和上界分别为 $LB_i=|dist(q,p)-dist(p,o_i)|$ 和 $UB_i=dist(q,p)+dist(p,o_i)$,当发现一对象 o_k 的上界不比 o_i 的下界差($UB_k\leq LB_i$)时,则可修剪掉对象 o_i.

2001 年 Yu 等人[46]提出了著名的 kNN 算法 iDistance,是一维度量索引利用三角不等式修剪的经典案例. 该算法根据参考点将每个聚类的高维数据点转换成单维数据点,然后索引到一个 B^+ 树中,并依据三角不等式进行修剪. 其本质思想在于:(1) 三角不等式保证了相对于参考点,易判断查询点和数据点是否相似;(2) 数据点相当于参考点的距离可以排序且能被索引到一个 B^+ 树当中. 最近,Lian 等人[47]提出了基于多枢轴(multipivot)成本模型的任意子空间的相似搜索方法,它是结合统计分析、多枢轴点索引,并利用三角不等式修剪的一维典型案例. 他们的主要贡献之一在于定义了两个可以运用三角不等式,对象 o 与枢轴距离的一维边界函数:$minscore(o^{(k)})$ 和 $maxscore(o^{(k)})$[47]. 这样便可以利用 $minscore(o^{(k)})>L_p(q^{(k)},piv^{(k)})+\varepsilon$ 或 $maxscore(o^{(k)})<L_p(q^{(k)},piv^{(k)})-\varepsilon$ 修剪对象 o,k 指查询对象 q 或枢轴 piv 的子空间维度,$k\in[k_{\min},k_{\max}]$.

(2) Filter-Refine 修剪策略

前述基于度量索引的修剪策略的缺点在于:不能适应非常高维(例如,256 维以上)的情形. 然而,这个问题可以通过 Filter-Refine 修剪策略来解决. 在该策略下,首先将原始空间 R_o 映射到低维空间 R_f(也称为特征空间)中,然后在 R_f 中使用类似于度量索引的三角不等式修剪方法,以过滤掉大部分的不相似的对象,此即 Filter 阶段的任务,而在 Refine 阶段再在 R_o 中使用真实的距离函数进行相似比较获得精确的结果. 不过,须强调的是 Filter-Refine 修剪策略是一个通用的修剪策略,并非一定要在特征空间中才可使用,而原始空间中也可使用,例如,文献[32]Yankov 等人提出的直接序列扫描的 Filter-Refine 过滤算法,这是因为在极高维(例如,512 维)的时序数据中,结合启发策略的直接序列扫描算法可能比索引方法更有效.

传统的高维 kNN 算法是运用 Filter-Refine 经典的范例. 它分两步完成:① Filter 步骤通过在原始数据集 $\mathcal{D}$ 上的满足下届定理的索引空间 $\mathcal{F}$(或称作特征空间)中执行 kNN 算法获得一个初步的候选集 $\mathcal{P}$,然后在原始空间 $\mathcal{O}$ 中确定 $\mathcal{P}$ 的对象 $o\in\mathcal{P}$ 与查询对象 q 的最大距离 $d_{\max}=\max\{d_o(o,q)\mid o\in\mathcal{P}\}$;② Refine 步骤再回到索引空间执行一个门限值为 $d_{\max}$ 的范围查询获得最终的候选集 $\mathcal{G}=\{o\in\mathcal{D}\mid d_f(\mathcal{F}(o),\mathcal{F}(q))\leq d_{\max}\}$,然后在原始空间 $\mathcal{O}$ 中计算 $\mathcal{G}$

的精确距离 d_o 并排序，取距离最小的前 k 个对象作为最终结果. 这里，d_o 和 d_f 分别指原始空间 $\mathcal{O}$ 和特征空间 $\mathcal{F}$ 的距离函数，$\mathcal{F}(o)$ 指对象 o 在 $\mathcal{F}$ 中的映射对象. 然而，Seidl 等人[48]研究发现传统算法 $d_{\max}$ 比实际的第 k 个近邻距离 $distNN_k$ 大，从而造成候选集 $\mathcal{P}$ 比候选集 $\mathcal{G}$ 大得多（例如，$\mathcal{G}$ 仅是 $\mathcal{P}$ 的 40%），引起大量冗余计算. 为此，他们提出了一个适合高维数据的最优多步 kNN 算法. 该算法是迄今为止最好的遵循 Filter-Refine 框架的高维 kNN 算法，其关键思想在于获得精确的 $d_{\max}$（$d_{\max}=distNN_k$）. 这可通过下列步骤获得：

（1）通过具有增量计算特性的 kNN 算法（例如，采用 DF 策略的 kNN 算法）迭代地产生一个特征距离 d_f 升序排列的候选集；

（2）利用一个升序排列的结果列表 *result* 存储 k 个近邻对象及其 d_o 距离，并使用 $d_{\max}$ 存储 *result* 的第 k 个近邻距离 *result*[k].*key*，当出现 $d_o\leq d_{\max}$ 的新对象时，将其插入 *result* 中并更新 $d_{\max}$（$d_{\max}=$ *result*[k].*key*），同时移除距离 $d_o>d_{\max}$ 的对象，经过一定步骤最终 $d_{\max}$ 将递减到实际的 k 近邻距离 $distNN_k$.

文献[44]中 Tao 等人最近将 Filter-Refine 过滤策略引入到 RkNN 查询中，提出了度量空间的 RkNN 查询算法. 该算法首先利用 Filter 步骤在 M-tree 上执行一个 DF 遍历，利用三角不等式修剪策略获得一个很小的初步候选集 $\mathcal{C}$；然后利用 Refine 步骤通过维护一个最小堆（min-heap）在 $\mathcal{C}$ 上执行 BF 遍历，进一步精炼候选 $\mathcal{C}$ 获得最终的候选集. 它是目前 RkNN 查询应用 Filter-Refine 过滤策略最好的范例.

4.2 多辨析加速方法

4.2.1 问题描述

多辨析（Multi-resolution）加速方法借助于数据的多辨析表示方法. 多辨析表示是一种由粗到精的多层表示数据方法，高辨析率相对于低辨析率具有更高的数据表示精度和计算成本，同时高辨析率表示与低辨析率表示具有某种函数依赖关系. 其工作过程为：先在使用低辨析率过滤数据对象，如果不能过滤掉则在更高一级的辨析率下过滤，直至最高的辨析率或达到预先指定的某个终止条件. 这样，如果能够在低辨析率表示下，过滤掉大部分不满足条件的时间序列，那么在整体上将会极大地节省计算成本. 图 1[27]显示了文献[27]中基于段平均的多辨析率表示方法 MSM. 从中可以看出，level 1 层（最高层）是所有 16 个数据点的平均值表示，而 level 4 层是基于每两个相邻点分段的平均值表示. 这种方法的关键在于：在保证无漏报的情况下，寻找到一个时序特征抽取函数（例如，段平均函数、haar 小波变换等），并找出低辨析层 S_{i-1} 和高一级辨析层 S_i 之间的函数关系，例如文献[18]提出的函数 $\alpha_p\cdot L_p(S_{i-1})\leq L_p(S_i)$，其中 $\alpha_p=\sqrt[p]{l}$（l 指每段的段长）为常数.

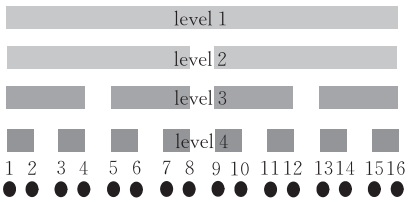

图 1 多辨析率分段平均近似

4.2.2 研究概述与分析

（1）L_p 距离函数情形

Chan 等人[15]1999 年最先提出了基于离散小波变换（DWT）的多辨析率过滤方案，其主要思想可概述为：给定两时间序列 X 和 Y 及其 haar 小波变换序列 S 和 R，它们的长度都为 n（其中 $n\geq 2$ 且为 2 的幂次方），且定义 $R-S=(C\ D_1\ D_2\cdots D_{n-1})$，那么 X 与 Y 的欧氏距离 $L_2(X,Y)=W_{\log_2^n}$ 可以用小波系数（$C\ D_1 D_2\cdots D_{n-1}$）递归得出，其中 $0\leq i\leq \log_2^n-1$，

$W_0=C$，

$W_{i+1}=\sqrt{2\times(W_i^2+W_{2^i}^2+W_{2^i+1}^2+\cdots W_{2^{i+1}-1}^2)}$，

其实质是小波系数表示的 $\log_2^n$ 层多辨析结构. Yi 等人[18]在 2000 年首次提出了针对任意 L_p 度量的两层多辨析率的段平均方法 SM（Segment Mean），即两时序等分段后的段平均序列之间的 L_p 距离的 $\sqrt[p]{l}$ 倍是原始序列 L_p 距离的下界函数. 同时也从理论上证明了段平均表示和段平均序列的 haar 小波变换的 L_2 度量存在系数为 $\sqrt{l}$ 的函数依赖关系.

文献[6]将基于 haar 小波的多辨析搜索方法引入到变长的子序列模式检测中，算法主要依赖于提出的多层次小波树（Shifted Wavelet Tree，SWT）. SWT 层次越高聚集的时间跨度越大段数越少，同层的各分段半重叠，这样长为 w（$w\leq 2^i$）的任意子序列将被包含在 SWT 中的第 $i+1$ 层的某个分段中. SWT 很好地解决了子序列模式持续时间长度不好确定的问题. 2005 年 Sakurai 等人[30]提出的监控流滞后相关的 BRAID 算法也利用了基于平均值的多辨析计算思想，他们将其称为平滑（smoothing）. 2007 年 Xiang 等人[27]将文献[18]的两层辨析率方法推广到多层，并将其应用到数据流的相似匹配中，

提出了一种多辨析率的层次段平均表示方法(Multi-scaled Segment Mean,MSM)[27],该方法实际是文献[18]中取段长为 $l=2$ 时 2 层多辨析率段平均方法 SM 和 2 层多辨析率 haar 小波变换方法的推广(如图 1 所示).

从以上的案例可以发现,如何寻找适合多辨析的距离函数或结构至关重要,同时它也是可能的研究方向之一.

(2)DTW 距离函数情形

2004 年由 Salvador 等人[41]提出的 FastDTW,是多辨析率方法使用的另一范例,也是第一个出现的加速 DTW 执行的算法.其主要思想是:先粗略计算然后再逐步修正,通过图 2[41]可以说明,它包括 3 个关键操作,即变粗糙(Coarsening)、投影(Projecting)以及精炼(Refinement).由于低辨析率弯曲路径可以作为高辨析率弯曲路径的参考,从而可约束其搜索范围,进而减少计算量.文献[41]指出在保持 r 较小的情况下,算法的运行时间复杂度和空间复杂度都近似为 $O(N)$,N 为累积矩阵对角线中方格的数目.2005 年由 Sakurai 等人[38]提出的 FTW 算法利用多粒度(对应不同的粗糙程度)逐步求精,使得出现在累积矩阵中的动态弯曲路径的搜索范围从起点到终点逐渐减少,计算逐渐精确,它是多辨析方法的又一范例.文献[38]指出 FTW 算法相对当时最好的 DTW 算法效率提高了 222 倍以上,实际上它是目前最好的 DTW 算法.综合以上两例可以看出,DTW 算法的累积矩阵是多辨析计算的重要基础.

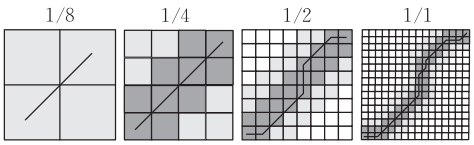

图 2　4 个不同解析率的弯曲路径

4.3　边界距离过滤方法

4.3.1　问题描述

边界过滤方法是通过加速时序距离计算的效率来减少计算成本.其关键在于寻找一个低计算成本、相对真实距离 D_{RL} 更小的下界函数 D_{LB} 以过滤掉满足 $D_{LB}\geq\varepsilon$ 的时间序列,或一个低计算成本、稍大于 D_{RL} 的上界函数 D_{UB},以选出那些满足 $D_{UB}\leq\varepsilon$ 的时间序列.形式化地,对于两时间序列 S 和 Q,$\exists D_{LB1}$ 和 D_{LB2},且 $D_{LB1}\leq D_{LB2}\leq D_{RL}$,则称 D_{LB2} 是比 D_{LB1} 严格的低边界距离函数,且称 $D_{LB1}(S,Q)$ 下界于 $D_{LB2}(S,Q)$,$D_{LB2}(S,Q)$ 下界于 $D_{RL}(S,Q)$.类似地,$\exists D_{UB1}$ 和 D_{UB2},且 $D_{UB1}\geq D_{UB2}\geq D_{RL}$,则称 D_{UB2} 是比 D_{UB1} 严格的上边界距离函数.不过注意,这种策略需要一个后处理过程来保持没有漏报.

4.3.2　研究概述与分析

(1)L_p 度量函数

文献[39]最早定义了基于 R-tree 的 kNN 查询的二个边界距离函数:即查询 q 和节点 N 的子树任意点的下界距离 $MinDist(N,q)$,查询 q 与节点 N 中最近点的上界距离 $MinMaxDist(N,q)$.该算法从根开始访问,然后递归地访问与查询 q 保持最小 $MinDist$ 距离的节点,在回朔到上层的过程中,它仅访问比当前 k 近邻距离更小的节点,同时利用上下边界距离修剪分支节点或叶子对象.文献[49]中的 Liu 等人通过 harr 小波变换提出了基于欧氏距离的时间序列的严格上下界函数,然后把它们运用到时序的相似搜索中.其主要思想反映在文献[49]的定理 1 中.但我们发现运用定理 1 在进行搜索时并不能明显改善搜索时间,主要原因在于他们采用顺序扫描的方法;之后,文献[50]以预计算的方式建立索引并利用聚类和三角不等式过滤方法提高了他们的算法.

(2)DTW 距离函数

由于 DTW 算法的时间复杂度为 $O(nm)$,相对 L_p 的复杂度 $O(n)$ 来说时间成本过大.为此,许多研究者针对 DTW 提出了许多边界距离函数,这包括 LB_Yi[36]、LB_Kim[51]、LB_Keogh[12]、LB_PAA[37]、LB_HUST[31]、LB_Z[35]、UB_Z[35] 等.下面我们将对其原理进行深入的分析和讨论.

LB_Yi 是由 Yi[36] 等人提出的首个针对 DTW 的低边界函数.它利用了下面的一个事实,即一条时间序列比另一序列的最大值(或最小值)要大(或小)的所有点,将至少为 DTW 贡献它们与另一序列的最大值(或最小值)的差值的平方距离.此事实也即两序列值的范围分别为不相交(disjoint)、交错(overlap)以及包含(enclose) 3 种情形下所得出的下界距离 D_{LB}.但是,LB_Yi 只能起到有限的过滤作用.

之后,Kim 等人[51]提出了比 LB_Yi 更接近真实 DTW 的下界距离函数 LB_Kim,其主要思想是:首先抽取两序列的 4 个特征值(即第一个元素值、最后一个元素值、最大值以及最小值),然后取两序列对应特征值的绝对差值中的最大值作为低边界距离.LB_Kim 使用 L_∞ 距离度量函数作为其基本度量函数,而对其它形式没有讨论.为此,Park 等人[26]扩

展了 LB_Kim 在使用 L_1 距离度量时的情况，为了方便我们把它称为 NLB_Kim. 上述的 LB_Kim 和 NLB_Kim 相对于 LB_Yi，其距离更接近于 D_{TW}.

显然，LB_Yi 和 LB_Kim 利用最大值、最小值或首尾元素时序特征得到的下界距离不可能很接近 DTW 真实下界距离. 为此，Keogh 等人利用全局的时间弯曲约束，从约束动态弯曲路径的上下边界入手，提出了下界距离函数 LB_Keogh[12]. LB_Keogh 的优点在于：它比 LB_Kim 和 LB_Yi 更接近真实 DTW 下界距离，同时支持形状时序表示和度量保持旋转不变特性. 然而，它不满足三角不等式，不能在低维索引空间中使用. 为此，Keogh 对其进行了扩展，将依据 PAA[18] 分段后由段平均组成的数据序列与查询序列，依照类似于 LB_Keogh 的定义得到了索引空间中的下界距离，即 LB_PAA. LB_PAA 下界于 LB_Keogh，这样便可以对 DTW 进行精确的索引. 但 LB_PAA 相对文献[37]的 LB_PAA 来说，并不是严格的索引空间的低边界距离. 将文献[37]中低边界距离称为 NLB_PAA，则有 $LB\_PAA \leq NLB\_PAA$. 它们的主要区别在于：LB_PAA 取时序上下边界曲线(文献[37]将其称为信封-Envelope)分段后的绝对最大值和绝对最小值作边界约束，而 NLB_PAA 取时序信封分段后的上边界段平均值和下边界段平均值作约束.

虽然，LB_Keogh 是很好的下界距离函数，然而它不是一个对称的距离函数，即 $D(S,Q) \neq D(Q,S)$，这使得它不能用于度量时序聚类的距离. 因为，使用 $D(S,Q)$ 计算时发现时序 S 和 Q 属于同一聚类，然而使用 $D(Q,S)$ 计算时却发现 S 和 Q 可能属于不同聚类[31]. 为此，Li 等人提出了一个对称的低边界度量函数 LB_HUST[31]，并把它应用于时间序列的聚类中. LB_HUST 中的思路非常直观，即既然一条时间序列可以有上下边界，那么另一条时间序列为什么不可以有呢，因为它们处于平等的位置. LB_HUST 下届于 LB_Keogh，即 $D_{LB\_HUST}(S,T) \leq D_{LB\_Keogh}(S,T)$，当将其应用于聚类时，可以定义聚类的低边界距离函数 $LB\_HUST_{cluster}(C_S,C_T)$，其定义类似于 LB_HUST，只是 C_S 和 C_T 指两个时间序列聚类.

然而，上面提到的所有 DTW 的边界距离函数都不具备增量计算的特性. 换句话说，对于仅一个元素不同的两个相邻的子序列 S_i 和 S_{i+1} 以及一个查询序列 Q，如果采用前述的边界函数，则须分别计算 $DTW(Q,S_i)$ 和 $DTW(Q,S_{i+1})$，从而计算 $DTW(Q,S_{i+1})$ 时并不能利用前面 $DTW(Q,S_i)$ 的计算结果，引起很多冗余计算[35]. 为此，我国学者 Zhou 等人[35] 最近同时提出了针对于 DTW 的低边界距离函数 LB_Z 和上边界函数 UB_Z 解决了上述问题，使得他们的边界距离可以运用到高速的数据流环境中. 而且，LB_Z 和 UB_Z 是迄今为止最好的低边界和上边界函数. LB_Z 和 UB_Z 利用了 DTW 两个重要的特性：(1) 在计算出最终的动态弯曲距离. $\gamma(n,m)=DTW(Q,S)$ 后，其它的 $\gamma(i,j)$ 也已计算出；(2) DTW 计算既可前向(forward)计算，也可后向(backward)计算.

为了对比，我们将上述的边界距离函数总结于表 2 中. 实际上许多时序相似搜索技术都可运用边界技术，寻求新的边界函数也是重要的研究课题之一.

表 2 边界距离函数比较

名称	公式来源	时间成本	适应范围	出现时间	技术思路
MinDist MinMaxDist	文献[39]	—	L_2	1995	对角线端点性质及 R-tree 中最小边界矩阵 MBR 每边必包含至少一个点的性质
LB_Yi	文献[36]	$O(n)$	DTW	1998	利用最大值和最小值特征
LB_Kim	文献[51]	$O(n)$	DTW	2001	利用首尾元素及最大值、最小值特征
LB_Keogh	文献[12]	$O(n)$	DTW	2002	利用全局约束缩短累积矩阵中动态弯曲路径的搜索范围和局部极值特征
LB_PAA	文献[37]	$O(n)$	DTW	2003	利用 LB_Keogh、分段线性表示和上下边界平均值
LB_HUST	文献[31]	$O(n)$	DTW	2006	两个时间序列地位是平等的，即对称性
文献[49]算法	文献[49]	$O(n)$	L_2	2006	利用小波变换中的欧氏距离不变性
LB_Z,UB_Z	文献[35]	$O(n)$	DTW	2008	利用 DTW 累积距离特性及双向计算策略，满足增量计算特性

4.4 Early Stopping

尽早终止 Early Stopping 算法的思想是相当直观的，它是指在计算两个对象的距离时，发现本次距离计算所积累的距离信息已经足够判断结果，则放弃本次计算或本次计算的部分计算(例如，DTW 累积计算矩阵的某些单元格)来缩减计算成本. 这种消除冗余计算的方法对于高维距离计算非常有效，不但能减少 CPU 运行时间，同时也可减少部分 I/O 成本.

(1) L_p距离函数

L_p距离函数的形式为 $L_p=\left\{\sum_{i=1}^{n}|x_i-y_i|^p\right\}^{1/p}$,显然当 $p\neq\infty$时,L_p具有单调性,故在计算两 n 维时序 R 和 S 的 L_p 距离 $L_p(R,S)$,当计算到第 k 维($1\leq k<n$)发现距离超过门限值 ε 时,则可终止后面 $n-k$ 维的距离计算,此即 L_p 及早终止计算思想.另外,针对 L_2的一个改进措施是,消除距离的平方根计算,直接使用 ε^2 进行距离比较.此方法可以推广到 L_p-norm($p\neq1$ 且 $p\neq\infty$).例如,文献[7,32]就利用这种方法来加速距离计算.

(2) DTW 距离函数

DTW 距离函数运用动态规划算法思想实现,它具有 3 个重要的性质:即从两序列的起始点(s_1,q_1)开始至终止点(s_m,q_n)结束;相邻单元格在弯曲路径 w 是相邻的(连续性);弯曲路径 w 在时间上是递增的(单调性).因此,DTW 算法存在行和列的累加效应,即该行(或列)的当前单元格的累加距离值将不超过该行(或列)之后单元格的累加距离值.这样,将有两种方法来加速 DTW 距离计算.一种方法是当累积矩阵的当前值超过门限值 ε 时,可放弃本次 DTW 距离计算,从而可过滤掉当前正在比较的序列;另一种是当累加矩阵中出现比当前累积值大的单元格时,那么该单元格后的行(或列)之后的所有单元格的计算可以终止,而从对角线的新单元格开始计算,此即缩小搜索范围来减少计算成本.如果是 kNN 计算,还可以维护第 k 个当前最好的近邻距离值 D_{cb},并在与其它的时序对象比较时不断地更新它,通过使用它作为门限值来过滤其它时序.显然,D_{cb}会不断减少从而使过滤时序越来越容易.文献[38]的 FTW 算法是运用这种 Early Stopping 算法思想很好的例子,不过须指出的是:FTW 还同时运用了多辨析计算和低边界函数过滤方法.

4.5　子序列搜索策略

通过长度为 m 的查询序列 Q,在长度为 n 的长序列 R 中搜索相似的子序列,直接实现的方法具有 $O(mn)$的时间复杂度.通常 $n\gg m$,因而子序列搜索具有极高的时间成本.为了提高效率,通常的子序列搜索分成以下几种类型:(1)基于空间索引和序列划分的子序列搜索方法,包括:GEMINI 框架[10]、Dual-Match 算法[42]、General-Match 算法[43]、Rank-Match 算法[52]等.这些方法中需要考虑:查询序列和原始序列的划分方法以及划分窗口的大小;索引的磁盘页面 I/O 效率;特征抽取方法、窗口尺寸效应以及点过滤效应[42]等;(2)基于直接距离计算的子序列搜索方法[28].由于 L_p 距离对噪声敏感且要求序列等长,而 DTW 算法克服了这些缺点,故这类算法通常采用快速 DTW 距离计算方法来实现,例如,FastDTW 算法[41]、FTW 算法[38]以及 SPRING 算法[28];(3)转换成字符的前缀和后缀索引的子序列搜索方法[25-26],这种方法的好处在于可以利用成熟的字符搜索技术.(4)基于模型的子序列搜索方法:LandMarks 模型[2]、SpADe 模型[13]以及文献[3]提出的算法等.这些算法将序列转换成对应的模式,然后定义模式距离函数进行子序列搜索.进一步地,基于 kNN 子序列算法[52]具有更高的时间成本,相对于前述的范围查询算法它将是重要的研究方向.

2001 年 Moon 等人[42]在分析了 GEMINI 框架引起多报的 3 个原因:即特征抽取、窗口尺寸效应和点过滤效应之后,提出了与 GEMINI 相反的子序列划分策略:将数据序列 S 划分成长度为 ω 的离散子序列,而将查询序列 Q 划分成 $Len(Q)-\omega+1$ 个滑动子查询序列,此即 Dual-Match 方法[42].然而,它仍留下窗口尺寸效应问题未解决,为此,2002 年 Moon 等人又提出了通用的子序列搜索方法,即 General-Match[43],此方法结合了 GEMINI 和 Dual-Match 的共同优点:既能运用 GEMINI 框架中的大窗口,又能利用 Dual-Match 中的点过滤效应.它将数据序列划分成 J-滑动窗口(J-sliding windows),将查询序列划分为 J-离散窗口(J-disjoint windows).不同于前述 Dual-Match 和 General-Match 的范围搜索方法,2007 年文献[52]Han 等人在它们的基础上提出子序列的 kNN 排序算法 Rank-Match,它的主要思想在于:(1)通过最小距离匹配窗口对 MDMWP 来定义所有匹配的窗口对中的最小边界距离;(2)利用延迟分组子序列检索方法提高搜索的 I/O 效率.

不同于基于空间索引和直接距离的子序列搜索方法,文献[25]提出了一种适应变长序列的子序列匹配的后缀树方法:通过增量构造把数据时序的所有后缀索引在后缀树中,并在搜索过程中如果累积矩阵某行中所有列的累积距离都大于 ε 即终止后面元素的匹配直接判断两序列不相似;文献[25]基于前缀查询的思想:如果时间序列 S 和查询序列 Q,它们的弯曲距离在 ε 范围内,那么至少有一个查询前缀序列满足它们与 S 的距离在 ε 内,扩展了使用 L_1距离的前缀查询.而不同于序列转换成字符的方法,基于模型的方法通过抽取序列的特征,并建立基

于特征的距离函数来匹配子序列. 文献[2]通过定义界标序列距离函数来匹配模式,文献[3]定义终端点序列距离函数在线搜索金融流子序列,而文献[13]提出了一种能够适用时间和幅度的偏移和缩放以及噪声,基于流时序形状的模式检测方法 SpADe. 然而,一般来说基于模型的子序列匹配方法可能会有较低的精度.

5 近似相似搜索技术

近似相似搜索通过放松相似查询的正确性限制来减少搜索成本,它已经发展成为高维相似搜索技术的一个重要分支. 这是因为:(1)通常等待完成精确相似搜索须较长的时间,而用户可能仅希望快速获得最先出现的近似结果;(2)用户理解的相似查询跟实际的实现存在差距,他可能希望通过循环反馈的方式来指定查询,从而需要获得初步结果以改进查询对象或距离函数. 文献[53]指出通常以精确相似搜索算法 1%的成本可以获取 99%的搜索结果,这表明近似相似搜索具有重要的实践意义.

5.1 前沿研究成果介绍与讨论

近似相似搜索算法设计的关键在于:(1)如何在精度(precision)和召回率(recall)之间保持平衡;(2)如何在运行效率和运行成本上取得平衡. 例如,如果在 5%的时间内已经获得 95%的结果,而剩余 5%的结果将需 95%的运行时间,那么可终止算法搜索过程;(3)如何快速地修剪搜索空间,包括整体的搜索空间(例如,基于 M-tree 修剪空间的方法)或单一距离计算空间(一般为尽早终止算法)两种类型;(4)如何高效地转换空间并能保留较多的原始距离信息,这类方法包括:特征抽取、维度约简(如 SVD 方法)以及映射(或称为嵌入 Embedding)对象等技术;(5)如何提供算法的概率保证,这包括利用距离分布信息建立概率模型,通过概率条件(例如,基于距离标准和基于精度标准)终止搜索过程等.

基于传统技术(如基于索引修剪搜索空间、空间转换或维度约简、特征抽取)是近似搜索技术的一个重要分支. 1998 年 Zezula 等人[54]提出了 3 种基于 M-tree 的近似相似搜索算法:(1)通过修剪搜索空间来缩减当前 kNN 查询的搜索半径,并保持相对距离错误为门限值 ε 的算法;(2)利用距离分布建立概率模型,并在发现的更好结果不超过用户指定的概率 ρ 时终止搜索的算法;(3)当第 k 个距离提高低于门限值 κ 时,然后终止搜索的算法; Castelli 等人[55]将同质的数据点分组成聚类,然后利用奇异值分解 SVD 技术分别对每个聚类进行维度约简,提出了近似处理近邻查询的 CSVD(Clustering with Singular Value Decomposition)算法;Gionis 等人[56]提出了基于局部敏感散列(Locality-Sensitive Hashing, LSH)方法的提高算法,它转换一个 D 维对象 p 成为一个包含 C 位二进制的向量 $v(p)$,并近似两个对象间的距离为 $v(p)$的编辑距离,然后使用散列技术索引 $v(p)$. 其缺点在于仅适应于 L_1 距离度量.

基于对象映射(有时也称为嵌入,Embedding)是近似搜索技术的另一个重要分支. 这类技术的关键在于:(1)如何定义一个较好的映射函数,以保证映射空间的距离能够保留较多原始空间信息;(2)映射函数是否具有“收缩”性质,即映射空间距离小于原始空间距离;(3)映射函数是否具有“距离保序”(proximity preservation)性质,即原始空间的距离顺序关系在映射空间中仍然不变. 映射技术通常使用异常(distortion)和压缩(stress)两个指标,分别评估映射空间中单一距离和整个映射空间距离的背离程度. Faloutsos 等人[57]从正交投影的角度首次提出了映射度量空间对象到 K 维欧氏空间的高效算法 FastMap,它既可作为高效近似相似搜索方法也可作为数据挖掘可视化工具; Wang 等人[58]从应用点积的角度提出了另一对象映射方法 MetricMap,它类似于 FastMap 算法,但具有更高的效率.

提供概率保证的近似相似搜索算法也是一种重要的类型,例如,PAC 算法[59]、文献[53,60-61]中的算法. PAC 算法的主要思想在于避免搜索太靠近查询对象,换句话说是在提供质量保证($1-\varepsilon$ 精度和 $1-\delta$ 概率)的基础下尽早终止搜索,以避免太多的时间成本花费在较少的精度提高上;文献[53]基于紧密的球面分区方法(例如,SA-tree)和增量的近邻搜索算法,提出了增量的概率近似搜索算法;文献[60]通过自动获取距离分布信息来提供概率保证,以缩减搜索过程中度量空间枢轴的半径; Bennett 等人[61]通过 K 聚类算法和数据高斯分布假设,并索引构造阶段估计均值向量和协方差矩阵参数,引入了基于索引聚类密度的概率 kNN 近似搜索算法 DBIN(density based indexing).

综上可以看出,近似搜索算法一方面可以利用精确搜索算法的原有技术,另一方面又需开发新的算法,例如,文献[62]中的基于有序排列的方法以及文献[63]中基于基因搜索的近似算法.

5.2 小　结

目前,已经出现大量基于高维的近似相似搜索技术.尽管它们各不相同,一般可从4个角度对它们进行分类:(1)使用的数据类型-基于向量空间(VS)还是基于度量空间(MS),VS又可分为基于L_p的VS_{L_p}和自由定义距离函数的VS两种;(2)使用的近似类型-通过改变搜索空间(CS)还是减少比较计算的方法(RC),RC又可分为减少整体比较空间的RC_{AP}和尽早结束单次比较计算的RC_{ES};(3)结果的质量保证-分成没有保证(NG)、确定的保证(DG)以及概率的保证(PG)3种类型,而PG包括基于参数的概率质量保证PG_{par}和无参数的概率质量保证PG_{upar};(4)交互方式-分为静态的方法(SA)和交互的方法(IA),在SA中用户不能自由选择参数,而在IA中用户可在查询时指定参数.我们总结上面一些典型的近似相似搜索算法于表3中.

表3　近似相似搜索算法分类

算法名称	数据类型	近似类型	质量保证	用户交互
LSH[56]	VS_{L1}	CS	PG_{upar}	SA
DBIN[61]	VS	RC_{ES}	PG_{par}	IA
CSVD[55]	VS	CS	NG	IA
FastMap[57]	MS	CS	NG	SA
Approximate Search with M-tree[54]	MS	RC_{AP}/RC_{ES}	DG/NG	IA
PAC[59]	MS	RC	PG_{upar}	IA
Probabilistic Proximity Search[60]	MS	RC_{AP}	PG_{upar}	IA
Probabilistic Incremental Search[53]	MS	RC_{ES}	NG	IA
Genetic Search[63]	MS	RC	NG	IA

6　总结和展望

随着时序相似搜索技术研究的不断进步,其应用领域广度正在不断扩展,它已经扩展到金融数据分析、医学诊断、DNA序列分析、网络流量监控、动物学图像分析、考古学文化迁移、视频监视、气象分析、天文学监控、传感器网络监视、移动对象跟踪以及运动捕获等诸多领域,其应用深度也在不断取得新的进展.然而,随着时序相似搜索技术应用领域的不断扩展、时序数据的高速增长(例如,DNA序列数据)以及新的应用场景的变化,使得它仍然面临巨大的挑战.一方面要求进一步提高算法的精度,因为许多算法计算出的结果仍然不能满足实际的需要;另一方面要求最大限度地降低算法的成本,这在kNN查询、RkNN查询、Skyline查询、关联时序搜索、异常时序搜索、主题时序搜索等应用场景下尤为突出.通过对国内外的已有成果的总结、分析和讨论,我们认为高效的时序相似搜索技术仍然具有广阔的前景,以下几个主题将可能成为未来的研究方向或研究热点:

(1)基于数据安全和数据压缩的时序相似搜索技术将成为一个重要的研究领域[64-65].实际上,这是源于数据隐藏和隐私保护(Privacy Preservation)的实现需要.而且,基于隐私保护的数据安全技术已经成为目前的研究热点.例如,最近文献[64]Papadimitriou等人研究了在时序中引入干扰(perturbation)造成部分不确定数据来隐藏时序的关键数据点,同时不丢失时序的固有模式并保持较好的数据压缩性的方法.

(2)基于生物信息学(Bioinformatics)序列数据的相似搜索技术仍将是重要的研究方向[1,33].近年来,随着生命科学的快速发展,产生了海量的生物序列数据(例如,DNA序列),如何处理这样大规模的数据是生物学家迫切需要解决的问题.相似搜索技术是获取生物数据属性的一种重要方法.尽管,这个方面取得了部分研究成果,但仍需提出新的算法或进一步提高算法的性能.

(3)基于数据流和传感器网络动态环境下时序相似搜索技术依然是重要的研究方向[3,13,28,30,35].一方面,传感器网络具有广阔的应用前景,另一方面基于数据流的时序相似搜索具有不同于静态时序处理方法的应用要求,例如,不可存储、一次扫描、实时处理等特点.由于数据流和传感器网络存在安全需要,因而结合安全技术的数据流相似搜索技术也是新的研究热点,例如,基于授权数据流的相似搜索[66].

(4)基于分布式处理和并行处理方法的时序相似搜索技术将是一个新的研究方向.首先,并行处理是提高海量时序处理效率的一种有效方式;其次,传统的时序相似搜索技术都是基于集中处理方式.而实际上,在很多的场景下海量的时序数据将分布在不同地方,分布式处理方式是时序相似搜索的必然要求.例如,最近文献[67]提出的可扩展分布式R-tree索引SD-Rtree.

(5)基于概率和不确定数据(Uncertain Data)的时序相似搜索方法将是新的研究热点之一.一方面概率的相似搜索可以在成本和效率上达到一种平衡,同时它能提供可确定的质量保证;另一方面在一些特殊的场合(例如,传感器网络、移动对象)存在

不确定数据是一种必然现象[68],这些数据通常要求统计概率处理方法.例如,空间移动轨迹不确定数据的处理.

(6)基于移动轨迹的相似搜索技术仍是研究的重要方向.一方面移动轨迹在一些场合(例如,交通管理、军事情报等)具有重要的应用价值,且它具有较强的安全需求;另一方面基于移动轨迹的研究仍有待进一步深入,大部分的研究成果基于空间模型进行匹配.因而,研究基于时间和空间的移动轨迹跟踪仍需继续研究,同时结合安全技术的移动轨迹相似搜索方法将是重要的研究方向,例如,弹道数据的版权保护问题、基于授权的空间索引[65]等.

综上,本文详细综述了时间序列中的高效相似搜索算法,提出了未来在该技术上要取得进一步进展的几个重要研究方向和趋势.我们已经拥有了部分的研究成果,同时也正在从事进一步的研究.

参 考 文 献

[1] Zhu Yang-Yong, Xiong Yun. DNA sequence data mining technique. Journal of Software, 2007, 18(11): 2766-2781(in Chinese)
(朱扬勇, 熊赟. DNA序列数据挖掘技术. 软件学报, 2007, 18(11): 2766-2781)

[2] Perng C S, Wang H, Zhang S et al. Landmarks: A new model for similarity-based pattern querying in time series databases//Proceedings of IEEE ICDE Conference. San Diego, CA, 2000: 33-42

[3] Wu H, Salzberg B, Zhang D. Online event-driven subsequence matching over financial data streams//Proceedings of the ACM SIGMOD Conference. Paris, 2004: 23-34

[4] Vlachos M, Kollios G, Gunopulos D. Discovering similar multidimensional trajectories//Proceedings of the 18th International Conference on Data Engineering (ICDE'02). San Jose, CA, 2002: 673-684

[5] Chen L, Ozsu M T, Oria V. Robust and fast similarity search for moving object trajectories//Proceedings of the ACM SIGMOD Conference. Baltimore, Maryland, 2005: 491-502

[6] Zhu Y, Shasha D. Efficient elastic burst detection in data streams//Proceedings of the 9th ACM SIGKDD. Washington, 2003: 336-345

[7] Keogh E, Palpanas T, Zordan V B et al. Indexing large human-motion databases//Proceedings of the VLDB Conference. Toronto, Canada, 2004: 780-791

[8] Agrawal R, Faloutsos C, Swami A. Efficient similarity search in sequence databases//Proceedings of the 4th International Conference on Foundations of Data Organization and Algorithms (FODO'93). Chicago, 1993: 69-84

[9] Berndt D J, Clifford J. Finding patterns in time series: A dynamic programming approach//Proceedings of the Advances in Knowledge Discovery and Data Mining. Menlo Park, CA, 1996: 229-248

[10] Faloutsos C, Ranganathan M, Manolopoulos Y. Fast subsequence matching in time-series databases//Proceedings of the ACM SIGMOD Conference. Minneapolis, Minnesota, 1994: 419-429

[11] Beckmann N, Kriegel H-P, Schneider R et al. The R^*-tree: An efficient and robust access method for points and rectangles//Proceedings of the ACM SIGMOD Conference, Atlantic City, NJ, 1990: 322-331

[12] Keogh E. Exact indexing of dynamic time warping//Proceedings of the VLDB Conference on Very Large Data Bases. Hong Kong, China, 2002: 406-417

[13] Chen Y, Nascimento M A, Ooi B C et al. SpADe: On shape-based pattern detection in streaming time series//Proceedings of the IEEE ICDE Conference. Istanbul, 2007: 786-795

[14] Chen L, Ng R T. On the marriage of Lp-norms and edit distance//Proceedings of the 30th International Conference on Very Large Data Bases (VLDB'04). Toronto, 2004: 792-804

[15] Chan Franky, Fu Wai-Chee. Efficient time series matching by wavelets//Proceedings of the 15th IEEE International Conference on Data Engineering (ICDE'99). Sydney, Australia, 1999: 126-133

[16] Korn F, Jagadish H, Faloutsos C. Efficiently supporting ad hoc queries in large datasets of time sequences//Proceedings of the ACM SIGMOD Conference, Birmingham U.K, 1997: 289-300

[17] Chen Qiu-Xia, Chen Lei, Lian Xiang et al. Indexable PLA for efficient similarity search//Proceedings of the ACM VLDB Conference. Vienna, Austria, 2007: 435-446

[18] Yi B K, Faloutsos C. Fast time sequence indexing for arbitrary Lp norms//Proceedings of the 26th International Conference on Very Large Databases (VLDB 2000). Cairo Egypt, 2000: 385-394

[19] Keogh E. Chakrabarti K, Pazzani M. Locally adaptive dimensionality reduction for indexing large time series databases//Proceedings of the ACM SIGMOD Conference. Santa Barbara, California, 2001: 151-162

[20] Cai Y, Ng R. Indexing spatio-temporal trajectories with Chebyshev polynomials//Proceedings of the ACM SIGMOD. Paris, France: ACM, 2004: 599-610

[21] Lin Jessica, Keogh E, Londardi S et al. A symbolic representation of time series, with implications for streaming algorithms//Proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery. San Diego, CA, 2003: 2-11

[22] Fu A, Chan P, Cheung Y, Moon Y. Dynamical VP-tree indexing for n-nearest neighbor search given pair-wise distances. VLDB Journal, 2000, 9(2): 154-173

[23] Ciaccia P, Patella M, Zezula P. M-tree: An efficient access method for similarity search in metric spaces//Proceedings of the 23rd International Conference on Very Large Databases. Athens, Greece, 1997: 426-435

[24] Navarro G. Searching in metric spaces by spatial approximation. VLDB Journal, 2002, 11(1): 28-46

[25] Park S, Chu W, Yoon J et al. Efficient searches for similar subsequences of different lengths in sequence databases//Proceedings of the 16th International Conference of Data Engineering. San Diego, CA, 2000: 23-32

[26] Park S, Kim S W. Prefix-querying with an L1 distance metric for time-series subsequence matching under time warping. Journal of Information Science, 2006, 32: 387-399

[27] Lian X, Chen L, Yu Jeffrey Xu et al. Similarity match over high speed time-series streams//Proceedings of the IEEE 23rd International Conference on Data Engineering (ICDE'07). Istanbul, 2007: 1086-1095

[28] Sakurai Y, Faloutsos C, Yamamuro M. Stream monitoring under the time warping distance//Proceedings of the IEEE 23th International Conference on Data Engineering (ICDE'07). Istanbul, 2007: 1046-1055

[29] Zhu Y Y, Shasha D. StatStream: Statistical monitoring of thousands of data streams in real time//Proceedings of the VLDB Conference. Hong Kong, China, 2002: 358-369

[30] Sakurai Y, Papadimitriou S, Faloutsos C. Braid: Stream mining through group lag correlations//Proceedings of the ACM SIGMOD Conference. Baltimore, Maryland, 2005: 599-610

[31] Li J, Wang Y, Li X. LB HUST: A symmetrical boundary distance for clustering time series//Proceedings of the 9th International Conference on Information Technology (ICIT'06). New Delhi, India, 2006: 203-208

[32] Yankov D, Keogh E, Rebbapragada U. Disk aware discord discovery: Finding unusual time series in terabyte sized datasets//Proceedings of the IEEE ICDE Conference. Istanbul, 2007: 381-390

[33] Zhang M, Kao R, Cheung D W, Yip K Y. Mining periodic patterns with gap requirement from sequences. ACM Transactions on Knowledge Discovery from Data, 2007, 1(2): 1-39

[34] Jiang T, Feng Y C, Zhang B et al. Finding motifs of financial data streams in real time//Kang et al eds. ISICA 2008. LNCS 5370. 2008: 546-555

[35] Zhou M, Wong M H. Efficient online subsequence searching in data streams under dynamic time warping distance//Proceedings of the IEEE ICDE Conference. Cancun, Mexico, 2008: 686-695

[36] Yi B K, Jagadish H V, Faloutsos C. Efficient retrieval of similar time sequences under time warping//Proceedings of the 14th International Conference of Data Engineering (ICDE'98). Orlando, Florida, 1998: 23-27

[37] Zhu Y, Shasha D. Warping indexes with envelope transforms for query by humming//Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD'03). San Diego, California, 2003: 181-192

[38] Sakurai Y, Yoshikawa M, Faloutsos C. FTW: Fast similarity search under the time warping distance//Proceedings of the ACM Symposium on Principles of Database Systems. Baltimore, Maryland, 2005: 326-337

[39] Roussopoulos N, Kelley S, Vincent F. Nearest neighbor queries//Proceedings of the ACM SIGMOD Conference. San Jose, CA, 1995: 71-79

[40] Hjaltason G R, Samet H. Distance browsing in spatial databases. ACM Transactions on Database Systems, 1999, 24(2): 265-318

[41] Salvador S, Chan P. FastDTW: Toward accurate dynamic time warping in linear time and space//Proceedings of the KDD Workshop on Mining Temporal and Sequential Data. Seattle, WA, 2004: 70-80

[42] Moon Y S, Whang KY, Loh W K. Duality-based subsequence matching in time-series databases//Proceedings of the 17th International Conference on Data Engineering. Heidelberg, Germany, 2001: 263-272

[43] Moon Y S, Whang K, Han W. General match: A subsequence matching method in time-series databases based on generalized windows//Proceedings of the ACM SIGMOD. Madison, Wisconsin, 2002: 382-393

[44] Tao Y F, Yiu M L, Mamoulis N. Reverse nearest neighbor search in metric spaces. IEEE Transactions on Knowledge and Data Engineering, 2006, 18(9): 1239-1252

[45] Chen L, Lian X. Dynamic skyline queries in metric spaces//Proceedings of the ACM EDBT Conference. Nantes, France, 2008: 333-343

[46] Yu C, Ooi B C, Tan K L et al. Indexing the distance: An efficient method to KNN processing//Proceedings of the ACM VLDB Conference. Roma, Italy, 2001: 421-430

[47] Lian X, Chen L. Similarity search in arbitrary subspaces under Lp-norm//Proceedings of the IEEE ICDE Conference. Cancun, Mexico, 2008: 317-326

[48] Seidl T, Kriegel H P. Optimal multi-step k-nearest neighbor search//Proceedings of the ACM SIGMOD Conference. Seattle, WA, USA, 1998: 154-165

[49] Liu B, Wang Z, Li J. Tight bounds on the estimation distance using wavelet//Proceedings of the 7th Conference on Web-Age Information Management (WAIM'06). Huangshan (Yellow Mountain), China, 2006: 460-471

[50] Feng Yu-Cai, Jiang Tao, Zhou Ying-Biao et al. An efficient similarity searching algorithm based on clustering for time series//Perner P ed. ICDM 2008. LNAI 5077. 2008: 360-373

[51] Kim S, Park S, Chu W. An index-based approach for similarity search supporting time warping in large sequence databases//Proceedings of the IEEE ICDE Conference. Heidelberg, Germany, 2001: 607-614

[52] Han W-S, Lee J, Moon Y-S, Jiang H. Ranked subsequence matching in time-series databases//Proceedings of the VLDB Conference. Vienna, Austria, 2007: 423-434

[53] Bustos B, Navarro G. Probabilistic proximity searching algorithms based on compact partitions. Journal of Discrete Algorithms, 2004, 2(1): 115-134

[54] Zezula P, Savino P, Amato G, Rabitti F. Approximate similarity retrieval with M-trees. The VLDB Journal, 1998, 7(4): 275-293.

[55] Castelli V, Thomasian A, Li C-S. CSVD: Clustering and singular value decomposition for approximate similarity search in high-dimensional spaces. IEEE Transactions on Knowledge and Data Engineering, 2003, 15(3): 671-685

[56] Gionis A, Indyk P, Motwani R. Similarity search in high dimensions via hashing//Proceedings of the VLDB Conference. Edinburgh, Scotland, UK, Morgan Kaufmann, 1999: 518-529

[57] Faloutsos C, Lin K-I. FastMap: A fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets//Proceedings of the ACM SIGMOD. San Jose, CA, 1995: 163-174

[58] Wang J T L, Wang X, Shasha D, Zhang K. MetricMap: An embedding technique for processing distance-based queries in metric spaces. IEEE Transactions on Systems, Man, and Cybernetics, Part B, 2005, 35(5): 973-987

[59] Ciaccia P, Patella M. PAC nearest neighbor queries: Approximate and controlled search in high-dimensional and metric spaces//Proceedings of the IEEE ICDE. San Diego, CA, 2000: 244-255

[60] Chávez E, Navarro G. Probabilistic proximity search: Fighting the curse of dimensionality in metric spaces. Information Processing Letters, 2003, 85(1): 39-46

[61] Bennett K P, Fayyad U M, Geiger D. Density-based indexing for approximate nearest-neighbor queries//Proceedings of the ACM SIGKDD. San Diego, CA, ACM Press, 1999: 233-243

[62] Chávez E, Figueroa K, Navarro G. Effective proximity retrieval by ordering permutations. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2008, 30(9): 1647-1658

[63] Bueno R, Traina A J M, Traina Jr C. Genetic algorithms for approximate similarity queries. Data and Knowledge Engineering, 2007, 62(3): 459-482

[64] Papadimitriou S, Li F, Kollios G, Yu P S. Time series compressibility and privacy//Proceedings of the VLDB Conference. Vienna, Austria, 2007: 459-470

[65] Yang Y, Papadopoulos S, Papadias D, Kollios G. Authenticated indexing for outsourced spatial databases. VLDB Journal, DOI 10.1007/s00778-008-0113-2, 2008: 1-18

[66] Papadopoulos S, Yang Y, Papadias D. CADS: Continuous authentication on data streams//Proceedings of the VLDB Conference. Vienna, Austria, 2007: 135-146

[67] Mouza C, Litwin W, Rigaux P. SD-Rtree: A scalable distributed Rtree//Proceedings of the IEEE ICDE Conference. Istanbul, Turkey, 2007: 296-305

[68] Soliman M, Ilyas I F, Chang K C-C. Top-k query processing in uncertain databases//Proceedings of the IEEE ICDE Conference. Istanbul, Turkey, 2007: 896-905

FENG Yu-Cai, born in 1946, professor, Ph. D. supervisor. His research interests focus on database technologies.

JIANG Tao, born in 1973, Ph. D.. His research interests focus on data mining.

LI Guo-Hui, born in 1973, Ph. D., professor, Ph. D. supervisor. His research interests focus on mobile temporal-space database technologies.

ZHU Hong, born in 1965, Ph. D., professor, Ph. D. supervisor. Her research interests include database security and XML technologies.

Background

At present, there are more and more time series data owing to its wide application in many domains, such as finance data analysis, Internet traffic analysis, sensor network monitoring, moving object tracking and motion capture. On one hand, it is owing to the increase of user requirement; on the other hand, many data in other domains can be transformed into time series. However, time series data is a typical high dimension and massive data. How to improve the efficiency of similarity search is a key problem on time series. The paper focuses on the efficiency analysis and discussion of time series similarity search.

This subject is supported by the National High Technology Development Program (863 Program) of China under grant Nos. 2007AA01Z309, 2006AA01Z430. These projects focus on research and development of database management system. The team has made a lot of progress in the area of DBMS and published nearly 20 papers in international and domestic journals or conference proceedings. Although many similarity search algorithms are proposed for time series, however the efficiency of these algorithms still needs to improve and can't satisfy the practical demand. The content of this paper mainly provides a summary for previous works and helps researchers pay attention to the interesting issues need to address.