软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software, 2022,33(3):968−984 [doi: 10.13328/j.cnki.jos.006453]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

时间序列对称模式挖掘*

李盼盼[1], 宋韶旭[1,2,3], 王建民[1,2,3]

[1](清华大学 软件学院, 北京 100084)

[2](大数据系统软件国家工程实验室(清华大学), 北京 100084)

[3](北京信息科学与技术国家研究中心(清华大学), 北京 100084)

通信作者: 宋韶旭, E-mail: sxsong@tsinghua.edu.cn

摘 要: 随着信息化和工业化的融合, 物联网和工业互联网蓬勃发展, 由此产生了以时间序列为代表的大量工业大数据. 时间序列中蕴含着很多有价值的模式, 其中, 对称模式在各类时间序列中广泛存在. 挖掘对称模式对于行为分析、轨迹跟踪、异常检测等领域具有重要的研究价值, 但时间序列的数据量往往高达几十甚至上百 GB. 使用直接的嵌套查询算法挖掘对称模式可能花费数月乃至数年的时间, 而索引、下界和三角不等式等典型加速技术最多只能产生一两个数量级的加速. 因此, 基于动态时间规整算法的启发, 提出了一种能够在 $O(w\times|T|)$的时间复杂度内挖掘出时间序列所有对称模式的方法. 具体来说, 给定对称模式长度约束, 基于区间动态规划算法计算出对称子序列,进而依据贪心策略选择数量最多且不重叠的对称模式. 此外, 还研究了在时间序列数据流挖掘对称模式的算法,并根据窗口内数据的特征动态调节窗口大小, 保证了对称模式数据的完整性. 采用 1 个人工数据集、3 个真实数据集在不同数据量下对上述方法进行实验. 由实验结果可知, 与其他对称模式挖掘方法相比, 该方法在模式挖掘结果及时间开销方面均有较好的表现.

关键词: 时间序列; 对称模式; 距离度量; 动态规划

中图法分类号: TP311

中文引用格式: 李盼盼, 宋韶旭, 王建民. 时间序列对称模式挖掘. 软件学报, 2022, 33(3): 968–984. http://www.jos.org.cn/1000-9825/6453.htm

英文引用格式: Li PP, Song SX, Wang JM. Time Series Symmetric Pattern Mining. Ruan Jian Xue Bao/Journal of Software, 2022, 33(3): 968–984 (in Chinese). http://www.jos.org.cn/1000-9825/6453.htm

Time Series Symmetric Pattern Mining

LI Pan-Pan[1], SONG Shao-Xu[1,2,3], WANG Jian-Min[1,2,3]

[1](School of Software, Tsinghua University, Beijing 100084, China)

[2](National Engineering Laboratory for Big Data Software (Tsinghua University), Beijing 100084, China)

[3](Beijing National Research Center for Information Science and Technology (Tsinghua University), Beijing 100084, China)

Abstract: With the integration of informatization and industrialization, the Internet of Things and industrial Internet have flourished, resulting in a large amount of industrial big data represented by time series. There are many valuable patterns in time series, among which symmetric patterns are widespread in various time series. Mining symmetric patterns has important research value in the fields of behavior analysis, trajectory tracking, anomaly detection, etc. However, the data volume of time series is often as high as tens or even hundreds of gigabytes. It can take months or even years to mine symmetric patterns using a direct nested query algorithm, and typical acceleration techniques such as indexing, lower bounds, and triangular inequalities can only produce speedup of one or two orders of

* 基金项目: 国家重点研发计划(2019YFB1705301, 2019YFB1707001); 国家自然科学基金(62072265, 62021002, 71690231); 工信部 2020 年新兴平台软件项目

本文由“数据库系统新型技术”专题特约编辑李国良教授、于戈教授、杨俊教授和范举教授推荐.

收稿时间: 2021-06-30; 修改时间: 2021-07-31; 采用时间: 2021-09-13; jos 在线出版时间: 2021-10-21

magnitude at most. Therefore, based on the inspiration of the dynamic time warping algorithm, this study proposes a method that can mine all the symmetric patterns of the time series within the time complexity of O(w×|T|). Specifically, given the symmetric pattern length constraint, the symmetric subsequences can be calculated based on the interval dynamic programming. Then the largest number of non-overlapping symmetric patterns can be selected according to the greedy strategy. In addition, we also study the algorithm for mining symmetric patterns in the time series data stream, and dynamically adjusts the window size according to the characteristics of the data in the window to ensure the integrity of the symmetric pattern data. Using one artificial data set and three real data sets to experiment with the above method under different data volumes, it can be seen from the experimental results that compared with other symmetric pattern mining methods, this method has better performance in terms of pattern mining results and time overhead.

Key words: time series; symmetric pattern; distance measurement; dynamic programming

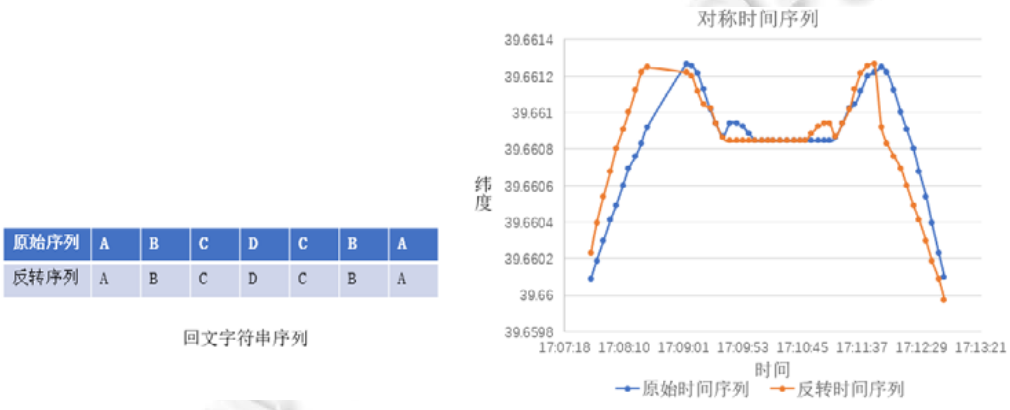
1 引 言

互联网、物联网、云计算等计算机科学技术经过长时间的共同发展与不断融合，积累了规模庞大、种类繁多的海量数据，涉及计算机科学、宏观经济、军事科技、医疗卫生等诸多领域. 在这些海量数据之中，有一类按照数据生成的时间顺序，把同一个变量或记录的数据值，或者高维数据的一个元组，排列而成的记录数据信息，被称为时间序列. 时间序列是工业界应用广泛的、与时间维度相关的高维数据，也是数据挖掘技术的一种主要研究对象. 挖掘时间序列中的对称子模式，可以分析时间序列的对称特征，方便时间序列的分类与预测，抽取出时间序列中蕴含的模式与规律，既可以为未来的决策提供理论与数据支持，又可以检测、判断、预防突发错误的出现，指导实际生产.

1.1 问题背景

随着信息技术的普及和发展，各行各业，尤其是工业领域，通过相应的传感器和信息系统积累了大量的时间序列数据，对特定领域或者特定模式的时间序列数据，利用数学建模、机器学习等方法进行建模和分析，已经成为一项意义重大且研究价值极高的项目课题.

大部分时间序列数据中数据点会随着时间的变化而产生一定的变化规律[1]，例如股票走势图、温度变化图和行车路线图等时间序列数据，在某些时间周期内呈现出较强的对称性. 然而，如图 1 所示，时间序列的对称性并不像回文字符串序列的对称性具有非常严格的数学定义，时间序列的对称性与时间间隔、序列模式和数据特征密切相关. 因此，需要立足于时间序列的时间和数据特征挖掘其子序列的对称性. 一般来说，只要原始时间序列和其反转时间序列的距离在合理阈值之内，就可以认定为对称时间序列. 对称时间序列在挖掘机、运输车等具有大量重复作业的工业场景频繁出现，且每个对称模式都意味着一个完整作业的生命周期. 因此，挖掘对称模式对于其轨迹跟踪和作业分析具有重要意义.

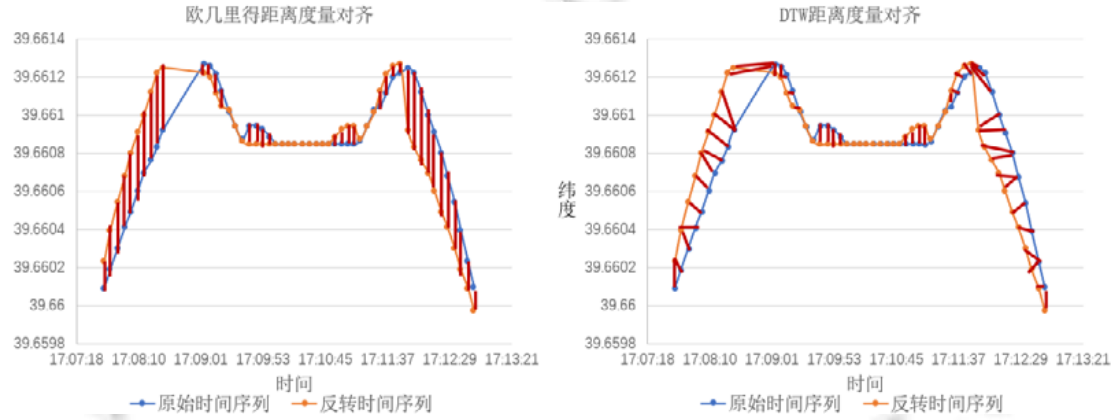


(a) (b)

图 1 回文字符串序列和对称时间序列对比

对称模式挖掘是伴随工业大数据而产生的问题，据调研，现在并没有挖掘时间序列对称模式的方法．因此，本文根据其他模式挖掘算法思想设计了对称模式挖掘算法的基准方法．例如，Yeh 等人提出了基于子序列相似性连接的 Matrix Profile 方法[2]挖掘时间序列模式．为了度量两个时间序列的距离和相似性，Matrix Profile 使用了归一化的欧几里德距离，然而，欧几里德距离并不适合度量时间序列的对称性[3]．由于采样频率和数据随机性，即使同一辆车也可能导致同一段轨迹拥有不同的时间长度，比如由于路况变化导致的速度问题等等．这些复杂情况下，使用传统的欧几里德距离强行将原始时间序列和反转时间序列一一对应，无法为每个数据点找到最合适的匹配方案，因而无法有效地求得时间序列之间的对称度．

例 1：公司对运输车运输轨迹进行跟踪，通过对往返一趟运输轨迹进行分析，计算得到往返过程中的路况和运输情况．通过比较原始和反转时间序列，本文发现，原始时间序列数据点与反转时间序列的最佳匹配点并不一定处于相同时间．如图 2(a)所示，欧几里德距离度量采用的是一一对应的方式计算原始和反转时间序列的相似性，这就导致在匹配过程中该方法并不能找到两个时间序列的最佳匹配点，使得两个时间序列的距离过大，从而难以度量序列真实的对称性．此外，由于存在数据缺失和偏移的情况，仅通过平移很难将时间序列完全对齐．而 DTW 距离度量方式则弥补了欧几里德距离度量的缺点，如图 2(b)所示，通过对时间序列在某些时间点进行动态压缩，为每个数据点选择距离最小的匹配点，从而度量出两个时间序列的最小距离．因此，本文使用了基于动态时间规整(DTW)[4]的挖掘算法度量时间子序列的对称度，保证模式挖掘效果．



(a) (b)

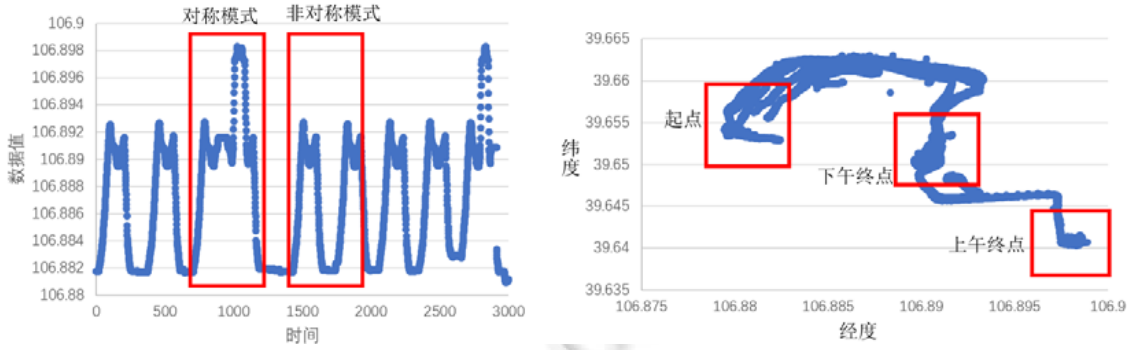
图 2　运输往返轨迹对称匹配

公司对运输车运行轨迹进行跟踪，从而计算运输车在不同起止点间的运输趟数．在此时间序列中，尽管出发点相同，但终点不同，导致车辆的运输轨迹不一致，如图 3(b)所示．车辆上午由煤矿出发，到距离较远的工厂运煤．下午由同样的煤矿出发，却出发前往另一个工厂．这必然导致车辆的运输轨迹、采样点数等的差异，即该时间序列中存在两个不同的子模式．如果设置时间子序列的时间跨度为 48 分钟(根据真实数据情况，上午的一趟运输约为 48 分钟，下午一趟运输约为 22 分钟)．如图 3(a)所示，由于全天的时间序列均是由传感器连续采集的，长度为 48 分钟的时间窗口必然跨越但并不完全包含多个长度为 22 分钟的对称子序列，则第 2 种时间子序列模式无法识别．

Matrix Profile 明确限定了时间序列子模式的长度．在真实应用场景中，一个时间序列中可能蕴含着多种子模式，人为限定时间子序列的长度会导致挖掘结果的缺失，如图 3 所示．因此，对称模式的窗口大小需要能够自适应的调整[5]，以保证挖掘结果的完整性．

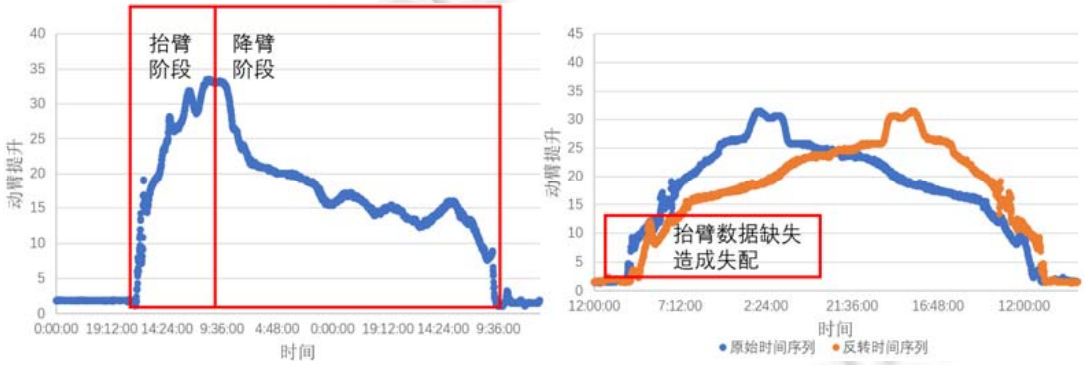
例 2：公司在挖掘机上安装了很多传感器，用于监控在挖掘作业中机器的工作状况．多种传感器采集到了多种类型的数据，以动臂抬升工况为例，其抬升工况在一次作业中的变化情况如图 3 所示．总体而言，一次作业中动臂抬升工况的变化具有对称性．然而，如图 4(a)所示，一次挖掘作业中，由于抬臂阶段和降臂阶段的时

长不一致, 采集点的个数不同, 其对称中心并不严格位于时间的中点. 因此, 在计算序列对称性时, 不能简单地查找对称中心并根据对称中心将时间序列分段, 以免计算的对称度不准确. 除此之外, 由于物理设备、数据采集和数据传输中遇到的问题, 实际工业场景中的时间序列可能不是等间隔采样的, 如图 4(b)所示, 抬臂阶段的数据缺失会导致在降臂阶段没有对应的匹配子段, 直接采用 Euclidean distance 度量中的一一对应匹配方法可能导致失配甚至误配. 因此, 需要定义一种新的对称度度量方式, 提出对称度计算与对称模式挖掘的新方法.



(a) (b)

图 3 运输车数据示例



(a) (b)

图 4 起重机抬降臂做功图

给定时间序列的长度约束, 挖掘出的对称子序列可能存在重复的情况, 从而导致后续数据分析和统计存在误差. 如图 5 所示为挖掘机斗杆外摆工况的时间序列, 该时间序列中的对称模式表示一次装车完成, 具有实际的物理意义. 然而, 尽管 a, b, c 都是符合对称规则的对称子序列, 若把三者都识别为对称模式, 最后统计的装车数量则与真实数量有偏差. 因此, 需要根据时间序列中对称子序列的分布, 挖掘出不重叠的对称子序列集合.

总之, 本文所提出的时间序列对称模式挖掘方法面临着以下问题和挑战.

(1) 不同于回文字符串, 对称时间序列并不要求原始时间序列和反转时间序列完全相同, 只需要两者的距离在指定阈值范围内即可, 而阈值与时间序列的整体数据特征有关.

(2) 时间序列数据有时间戳标识, 在采集过程中也可能出现缺失和偏差, 计算其对称性需要充分考虑数据点的时间间隔和采集频率, 选择最佳匹配点, 不能采用欧几里德距离一一对应的方案.

(3) 同一个时间序列中可能蕴含着多种对称模式, 其时间跨度差别明显. 因此, 挖掘时间序列中全部的对称模式, 需要算法可以自适应调整时间窗口的大小.

(4) 挖掘出的对称子序列可能存在重叠，需要根据给定约束挖掘出不重叠的对称子序列集合.

基于此，本文主要针对时间序列数据的对称模式挖掘问题进行研究，通过子序列长度约束挖掘出时间序列中的所有子模式.

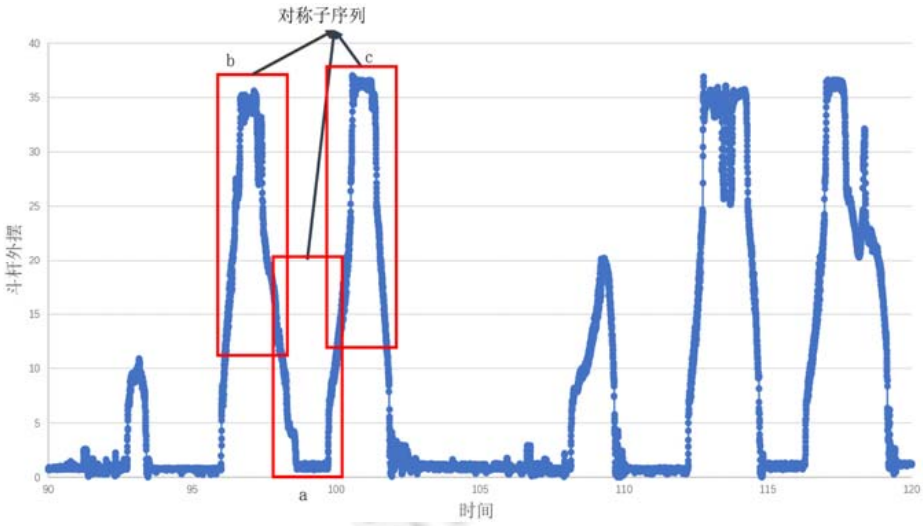


图 5 不重叠对称时间子序列挖掘

1.2 主要贡献

本文主要贡献如下:

(1) 定义了对称子序列和对称模式. 本文所挖掘的模式是关于中心点前后对称的子序列模式，其对称性由原子序列和反转子序列的 DTW 距离进行度量. 对称模式在实际工业生产中非常常见，具有很高的应用价值.

(2) 提出了静态时间序列对称模式的挖掘方法，并给出了具体的算法，通过对称子序列的长度约束和对称性约束，采用区间动态规划算法计算出所有长度限定的对称子序列，再根据获取数量最多对称子序列的贪心策略，计算得到所有不重叠的对称模式.

(3) 将对称子序列的挖掘算法扩展到了数据流上，每生成一个新的数据点，实时计算出包含当前数据点的子序列对称性，根据对称性阈值及时筛选出对称子序列.

(4) 研究了时间子序列对称性和窗口大小的关系，分析建模了对称序列数据点差分的数据范围，并根据范围自适应调整窗口大小.

(5) 采用 1 个人工数据集和 3 个真实工业时间序列数据集在挖掘对称模式的数量、精确率、完整性以及时间开销方面进行了实验，并与多种现有相关方法进行了对比. 结果表明: 本文所提出的时间序列对称模式挖掘算法在挖掘效果方面表现最优; 同时，在时间开销方面有着几乎最佳的性能.

1.3 论文结构

本文第 2 节给出本文研究相关的基础定义，对时间序列、对称子序列以及对称模式进行解释，并提供问题形式化定义即模式挖掘的目标. 第 3 节提出具体基于子序列长度和对称性约束的动态规划挖掘方法，包括具体方法描述和具体算法等. 第 4 节提出对称模式的流式挖掘方法，可以在每个时刻实时得到对称模式. 第 5 节介绍自适应窗口以提高所挖掘对称模式的数据完整性. 第 6 节通过 1 个人工数据集和 3 个真实工业时间序列数据集，在模式挖掘效果和时间性能等方面与现有方法进行对比分析，并通过多个数据集，在模式挖掘精确率上与现有方法进行比较验证. 然后，在第 7 节中对相关工作进行介绍. 最后，在第 8 节对本文的工作进行分析总结.

2 基础定义

作为大数据技术及人工智能技术在工业领域应用的数据支撑, 工业大数据正成为数据相关研究领域的热点. 本文主要研究工业数据中的时间序列数据, 为便于叙述, 本节将对时间序列、时间子序列、对称子序列、对称模式等进行定义[6].

定义 1. 时间序列指一系列包含时间戳的数据点.

具体而言, 在一条数据序列 $x=x[1],x[2],\ldots$中, 数据点 $x[i]$指第 i 个数据点, 每个数据点 $x[i]$都具有一个时间戳 $t[i]$. 为简便起见, 下文中将 $x[i]$简写为 x_i, $t[i]$简写为 t_i.

定义 2. 反转时间序列指对原始时间序列的所有数据点前后逆序排列后得到的时间序列.

参考转置矩阵的定义[7], 本文提出了反转时间序列. 具体而言, 如果原始时间序列为 $x=x_1,x_2,\ldots$, 那么其反转时间序列为 $x^T=x_n,\ldots,x_2,x_1$.

定义 3. 时间子序列指时间序列中点的顺序子段.

从时间序列 $x=x_1,x_2,\ldots$中截取一段连续的子序列 $s=x_i,x_{i+1},\ldots,x_{i+m-1}$, 其中, m 表示子序列 s 的长度.

定义 4. 对称子序列指具有对称性的时间子序列.

换言之, 如果一个时间子序列 $s=x_i,x_{i+1},\ldots,x_{i+m-1}$ 沿某条直线折叠后, 直线两侧的数据能够近似重合, 那么这个子序列称为对称子序列.

定义 5. 对称模式指从原始时间序列挖掘出的一组对称子序列.

从同一个时间序列中挖掘出的对称子序列保存了原有序列的信息, 可作为一组对称模式.

定义 6. 挖掘对称模式指在时间序列中计算不重叠的一组对称子序列.

给定一个时间序列 T, 子序列长度约束 w, 子序列相似度约束 d, 时间序列对称子序列挖掘指: 计算时间序列 T 中长度为 w 范围且对称度小于相似度约束 d 的所有不重叠时间子序列, 同时保证时间子序列的数目最多, 即

$$\max|\{x_i,x_j,\ldots|symmetry(x_i)\leqslant d\cap|x_i|=w\cap 1\leqslant i+|x_i|\leqslant j\leqslant|T|\}|.$$

如图 6 所示, 计算得到的不重叠时间子序列可以为后续的数据分析和引用提供关键参考. 具体方法可见第 3 节.

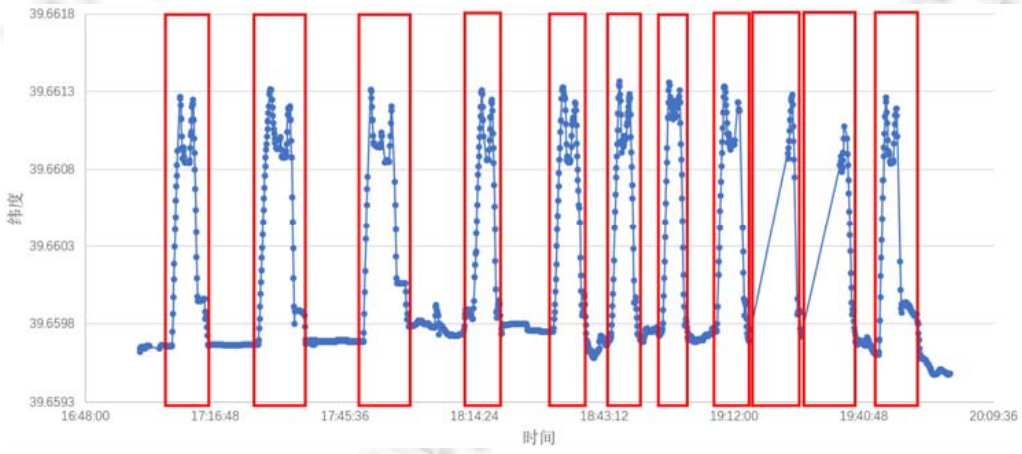


图 6 对称子序列挖掘结果

3 对称模式挖掘算法

基于上述各定义, 本节将进一步对所提出的对称模式挖掘方法进行具体阐述. 整体来看, 若要基于长度和距离约束挖掘时序数据的对称模式, 首先要得到子序列对称度的计算方式. 此外, 通过动态规划方法计算

给定长度约束内子序列的对称度. 最后, 根据挖掘数量最多不重叠对称子序列的贪心策略, 挖掘出所有对称模式.

3.1 定义时间序列对称度

给定对称子序列的长度约束 w, 需定义时间序列所有长度为 w 的子序列对称度的计算方式. 根据一般数学定义方式, 计算序列的对称性需先确定对称中心, 然后计算对称中心两侧的序列距离[8]. 然而, 时间序列的对称度则不能如此衡量. 时间序列数据是由真实工业场景中, 由传感器采样得到的. 由于数据产生的速率不一致, 比如起重机提升重物的过程中, 由于提臂和降臂做功的不同, 必然导致提臂和降臂阶段的时间长度和采样点个数存在较大差别, 如图 4 所示, 在从高处向低处运输重物时, 起重机主要在降臂时做功, 因此, 传感器在降臂阶段采集的数据点个数远多于升臂阶段. 因此, 时间序列的中心点较难确定. 此外, 由于传感器、终端记录器等设备在数据采集、数据传输、数据记录等过程中会受到主观和客观因素的影响, 受制于物理和技术上的约束, 最终所得到的数据会存在一定的数据质量问题. 这些数据质量问题也会导致逐点计算时间序列距离的方式必然使得度量方式不精确. 如例 2 所示, 升臂阶段速度过快产生的数据缺失, 导致降臂阶段的数据在此时间段的匹配点缺失, 产生失配现象. 而 Matrix Profile 等传统匹配方式采用一一配对的方式计算时间序列距离, 导致升臂阶段数据点的最佳匹配点滞后于降臂阶段, 从而影响匹配效果.

基于此, 时间序列的对称性度量方式, 不能立足于时间序列中心点的确定, 也不能采用逐点匹配的距离度量方案. 本文定义了一种新的基于动态规整算法的对称度度量方式, 即使用时间序列与其反转序列的 DTW 距离来度量该时间序列的对称度. 换言之, 如果一个时间子序列 $s=x[i],x[i+1],\ldots,x[i+m-1]$与它的反转序列 $s=x[i+m-1],x[i+m-2],\ldots,x[i]$的 DTW 距离小于给定阈值, 那么这个子序列称为对称子序列. 如图 2 所示, DTW 算法可以通过找到两个匹配度最高的点来度量时间序列的距离和对称度, 很好地解决了缺失点和失配问题. 然而, 利用 DTW 算法暴力求解所有时间子序列对称度的算法复杂度过高, 为 $O(w^2\times|T|)$, 因此, 本文提出了区间动态规划的改进算法, 大大降低了时间复杂度. 此算法将在下一节中详述.

3.2 计算子序列对称度

给定对称子序列的长度约束 w, 需要计算时间序列 T 所有长度为 w 的子序列的对称性. 基于动态时间规整算法的启发, 本节提出了子序列对称性的计算方法.

首先考虑给定长度时间子序列的对称性计算方法. 如图 7 所示, 假设时间序列 T 的长度为 m, 则 $x[1]$应当与 $x[m]$进行匹配, 可以使用欧氏距离或曼哈顿距离直接度量其相似度. 然而, 由于时间序列的采集频率和位置不同, $x[2]$不仅仅可以与 $x[m-1]$进行匹配, 也可以与 $x[m]$进行匹配; 同理, $x[m-1]$也可以与 $x[2]$和 $x[1]$进行匹配. 对称时间子序列可能由于采集频率的不同, 前后某段匹配点的时间并不完全对称, 采用线性扩张的方式进行对齐, 可以弥补由于采样偏差导致的失配问题.

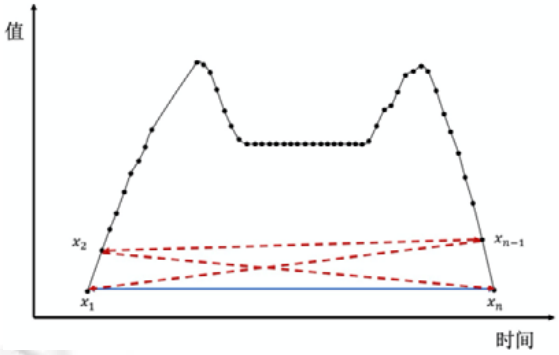


图 7 对称子序列首尾匹配候选点

根据如上思想, 本节使用 $D(i,j)$表示时间子序列 $T[i,..,j]$的对称度, 计算该对称度的方法如下:

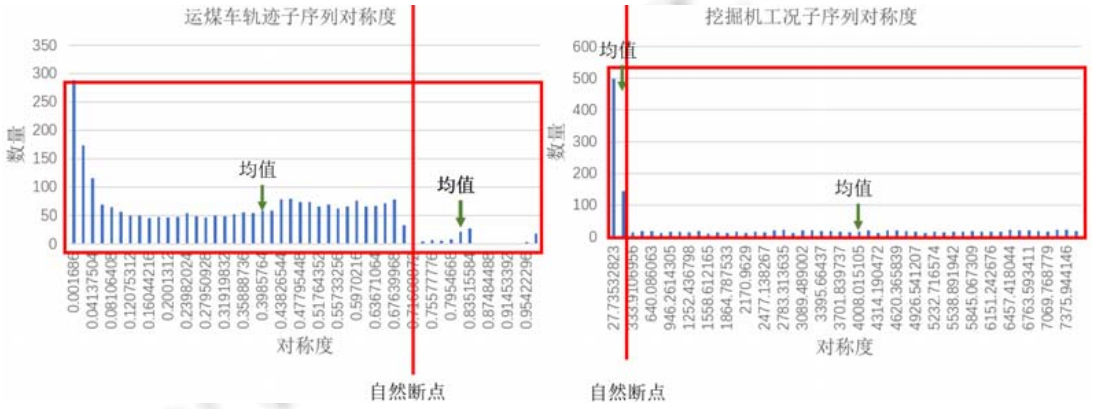
搜索以 $x[i],..,x[j]$等所有点为中心点、其左侧时间序列和右侧时间序列的逆序序列的相似度. 尽管中心点的位置可能不同, 但是(i,j)的上一个匹配点只可能是以下 3 种情况之一: $(i+1,j)$, $(i,j-1)$和$(i+1,j-1)$.

因此, 该匹配过程可以归约成如下形式, 从每个长度为 2 的子序列出发, 每次合并一个新数据点, 新数据点的匹配选择由 $dp(i+1,j)$, $dp(i,j-1)$和 $dp(i+1,j-1)$的最小值决定, 即

$$dp(i,j)=dp(i,j)+\min\{dp(i+1,j),dp(i,j-1),dp(i+1,j-1)\} \quad (1)$$

3.3 确定对称度阈值

计算出所有子序列的对称度之后, 需要确定满足对称特征的距离阈值, 即假设距离阈值为δ, 若一个时间子序列的对称度大于δ, 则该时间序列不具备对称性; 否则, 若对称度小于或等于δ, 则具有对称性. 而不同领域的时间序列数据往往具有不同种类和数量的对称模式, 相应地, 其子序列对称度的数值和分布也往往有较大差异. 如图 8 所示, 尽管运输车运输轨迹和挖掘机单次作业的工况数据具有对称性, 然而这两种时间序列的子序列对称度在分布和度量量纲上区别明显. 因此, 需要根据数据特征确定一个唯一的距离阈值.



(a) (b)

图 8 运输车轨迹和挖掘机工况子序列对称度分布

在概率统计中, 方差用于衡量数据集中数据的偏离程度. 因此, 很多算法使用方差作为度量指标变化的阈值[9]. 本算法需要根据时间序列数据分布特征确定一个对称度阈值, 从而将子序列分为两个类别, 即对称子序列和非对称子序列. 根据聚类方法[10]和自然断点分类的划分原则[11,12], 同一个类簇中数据的相似度高而不同类簇中数据的相似度低. 换言之, 两个对称子序列的对称度差距应尽可能小, 而对称子序列与非对称子序列的对称度差距应尽可能大. 本文提出了基于方差度量的对称度阈值确定原则, 即在子序列对称度组成的集合中, 通过选择某个合适的值作为对称阈值, 对称度小于该阈值的子序列为对称子序列集合, 对称度大于该阈值的子序列为非对称子序列集合, 使得对称子序列集合和非对称子序列集合的对称度方差之和最小.以图 8 为例, 在运输车子序列对称度和股票子序列对称度中选择合适的自然断点, 可使得对称模式和非对称模式的均值适中, 方差之和最小. 具体算法如下.

算法 1. 对称度阈值划分算法.

输入: 子序列对称度列表 dp.

输出: 对称度阈值 d.

```
1:  sorted(dp);
2:  a[1]=dp[1] ;  //a 为对称度值列表
3:  for i←2 to |dp| do
4:    a[i]=a[i−1]+(dp[i]−a[i−1])/i;
5:    l[i]=(i−1)/(i×i)×(dp[i]−a[i−1])×(dp[i]−a[i−1])+(i−1)/i×l[i−1];  //前 i 小的时间子序列对称度方差
```

```
6:    end for;
7:    reverse(dp);  //反转时间序列以计算非对称子序列集合方差
8:    a[1]=dp[1];
9:    for i←2 to |dp| do
10:     a[i]=a[i−1]+(dp[i]−a[i−1])/i;
11:     r[i]=(i−1)/(i×i)×(dp[i]−a[i−1])×(dp[i]−a[i−1])+(i−1)/i×r[i−1];  //前 i 大的时间子序列对称度方差
12:   end for;
13:   reverse(r);
14:   reverse(dp);  //恢复对称度列表原始顺序
15:   d=l[1]+r[2];
16:   idx=1;
17:   for i←1 to |dp|−1 do
18:     if  l[i]+r[i+1]<d then
19:       d= l[i]+r[i+1];
20:       idx=i;
21:     end if;
22:   end for;
23:   return dp[idx];
```

3.4 挖掘对称子序列

计算出时间序列 T 所有在长度约束范围之内的子序列对称度之后，可以利用贪心算法挖掘得到所有满足对称性的子序列. 以正弦时间序列为例，若其周期为 t，在挖掘过程中，则存在长度为 t 的对称子序列相互包含的情况. 如例 2 所示，子序列 a, b, c 均是对称子序列. 由于要挖掘不重叠对称子序列，若结果集合中选择了 a，则不能再选择 b 和 c. 因此，为了充分利用时间序列的信息，本文以对称子序列的数量最大值为优化目标. 根据贪心算法的思想，若上一个选择的对称子序列为 s_i 其长度为 w，则下一个对称子序列的贪心选择策略为从数据点 $i+w$ 开始，第 1 个长度为 w 的对称子序列，可以保证能得到数量最多的不重叠子序列.

本文根据上述对称模式挖掘方法提出了相应的具体算法，如算法 2 所示. 第 1 行−第 11 行根据第 3.2 节提出的对称度计算方式计算了给定时间长度约束下所有子序列的对称度，第 12 行根据自然断点的选择原则确定了对称度阈值，第 13 行−第 21 行根据贪心策略计算出数量最多的不重叠对称子序列.

算法 2. 时间序列对称模式挖掘算法.

输入：顺序时间序列 T，长度约束 w，相似度约束 d.

输出：对称模式 S.

```
1:    for i←1 to |T| do  //初始化时间子序列对称度
2:      dp[i][i]=inf;
3:    end for;
4:    for i←1 to |T|−1 do
5:      dp[i][i+1]=distance(T_i,T_{i+1});
6:    end for;
7:    for len←3 to w do  //计算长度为 w 的子序列对称度
8:      for i←len to |T|−len+1 do
9:        dp[i][i+len−1]=distance(T_i,T_{i+len−1})+min(dp[i][i+len−2],dp[i+1][i+len−1],dp[i+1][i+len−2]);
10:     end for;
11:   end for;
```

12: *d*=*calculate_natural_break*(*dp*); //调用算法 1 计算相似度阈值
13: **for** *i*←1 to |*T*|−*w*+1 **do**
14: **if** *dp*[*i*][*i*+*w*−1]≤*d* **then**
15: 把 *T*[*i*,*j*]放入对称子序列集合 *Q*;
16: *i*←*i*+*w*;
17: **else**
18: *i*←*i*+1;
19: **end if**;
20: **end while**;
21: **return** *Q*;

由前述定义可知, 第 1 行–第 11 行计算子序列对称度的时间复杂度为 O(*w*×|*T*|), 第 12 行自然断点的确定和不重叠对称子序列的挖掘均只需要 O(|*T*|)时间. 因此, 本算法的整体时间复杂度为 O(*w*×|*T*|), 其效率远远高于直接利用 DTW 计算的 O(w^2×|*T*|), 为后续的数据分析提供数据基础.

4 对称模式流式挖掘算法

在工业场景中, 经常有在对称模式发生时即时报警的应用. 例如, 金融时间序列中的对称模式意味着当前经济周期的结束, 地震预测中的对称模式往往意味着下一个余震的开始. 因此, 快速实时的对称模式发现非常重要. 因此, 本节将对称模式挖掘算法扩展到了流式应用场景.

4.1 流式对称子序列计算

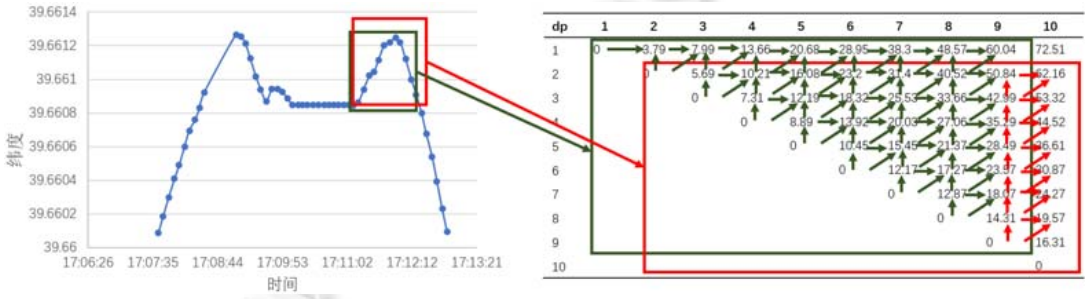
流式时间序列是一组顺序、大量、快速且连续到达的时间序列数据. 一般情况下, 时间序列数据流可被视为一个随时间延续而无限增长的动态数据集合. 检测流式时间序列的对称性需要满足实时性, 即给定一个实时时间序列 *T* 和时间窗口 *w*, 需要在每个数据点 *i* 生成时, 检测时间序列 *T*[*i*−*w*+1,..,*i*]的对称性.

由第 3.2 节可知, 使用 *dp*[*i*,*j*]表示时间序列 *T*[*i*,..,*j*]的对称度. 因此, 当新到来一个数据点时, 可以用公式(2)计算 *dp*[*i*+1,*j*+1]:

$$D(i+1,j+1)=dist(i+1,j+1)+\min\{D(i+2,j+1),D(i+1,j),D(i+2,j)\} \tag{2}$$

然而, 公式中 *D*(*i*+2,*j*+1)尚未计算, 不可直接使用. 若直接用 DTW 算法计算 *T*[*i*+1,..,*j*+1]与其反转序列的 DTW 距离, 则其时间复杂度又退化为 O(w^2). 因此, 需要考虑如何利用已计算的状态优化流式算法.

如图 8 所示, 通过观察动态规划状态的推导过程可知, 当新增一个数据点 *j*+1 时, [*i*+1,*j*]窗口范围之内的状态不须重新推导, 因为其推导顺序未改变, 起始状态亦未改变. 因此, 只需计算 *D*[*k*,*j*+1](*i*+1≤*k*≤*j*+1,即图 8 所示最后一列状态)即可. 由图 9 的转移路径可知, 只需在 O(*n*)时间内即可计算出 *dp*[*k*,*j*+1], 相比直接计算对称性的 O(w^2)算法, 大大降低了时间复杂度.



(a) (b)

图 9 对称模式流式挖掘算法

4.2 算 法

本文根据上述对称模式流式挖掘方法提出了相应的具体算法，如算法 2 所示.

算法 3. 时间序列对称模式流式挖掘算法.

输入：新数据点 $T[j+1]$，长度约束 w，相似度约束 d.

输出：布尔型变量，标志子序列 $T[j+2-w,..,j+1]$是否为对称子序列.

1: $dp[j][j+1]=distance(T_j,T_{j+1})$;
2: **for** $i\leftarrow j-1$ to $j+2-w$ **do** //按区间长度递增的顺序计算 $dp[k][j+1]$
3: $dp[i][j+1]=distance(T_i,T_{j+1})+\min(dp[i][j],dp[i+1][j+1],dp[i+1][j])$;
4: **end for**;
5: **if** $dp[j+2-w][j+1]\geqslant d$ **then**
6: **return false**;
7: **else**
8: **return true**;
9: **end if**;

5 自适应窗口

时间序列的数据和模式均具有很强的随机性，同一个时间序列中很可能存在不同的模式. 以挖掘机为例，挖掘不同的坑道时，斗杆移动的轨迹就不同. 人为设置时间序列对称模式的长度尽管可以挖掘出所有的对称模式，但是每个模式的信息却经常存在缺失，如图 10 所示. 若设定对称子序列的数据点个数为坑道 a 的序列长度，则坑道 b 的对称子序列不能完全挖掘. 除此之外，一个时间窗口往往只能挖掘出固定的一种对称模式，挖掘多个模式就需要不同长度的时间窗口. 因此，当需要挖掘的对称模式长度不一时，窗口需要自适应地变化. 为避免引入新的参数，本节利用数据点差分的分布特征，使得窗口能够自动调节大小.

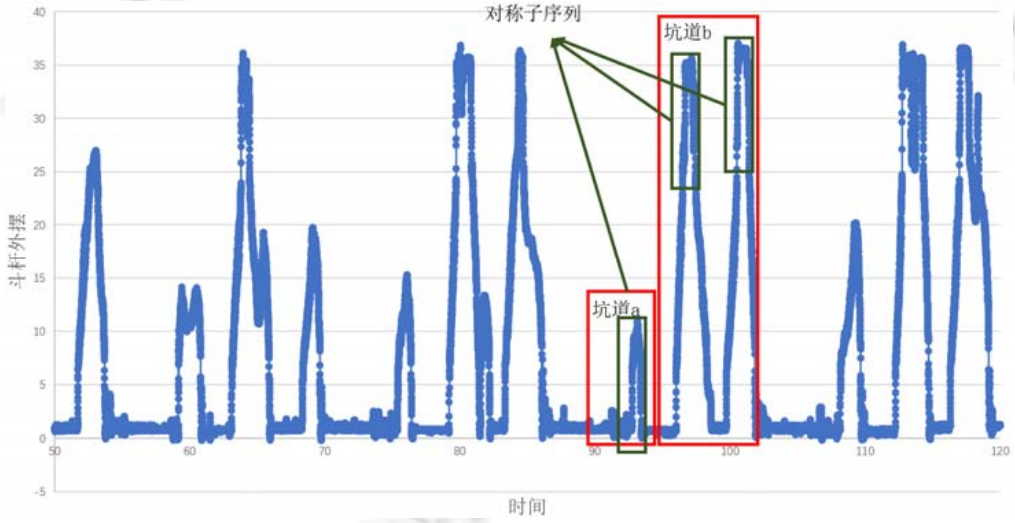


图 10 挖掘机斗杆外摆

根据时间序列对称性的定义，在设定对称子序列的最小长度 w 之后，对称时间序列的首尾点 $x[i]$和 $x[i+w-1]$必定匹配. 换言之，首尾点的距离若接近 0，则证明该窗口大小合适；若远离 0，则需调大窗口，以完整获取对称子序列的全部信息. 因此，可以使用时间子序列首尾点的距离来衡量窗口大小设置得是否合适.

然而，真实工业场景中的数据具有很强的随机性，并不一定满足首尾点距离为 0. 因此，本文考虑利用相

邻点距离所组成数据集的分布特点, 判断窗口是否应当扩张或者闭合. 根据工业大数据时间序列点的随机性, 首尾数据点的距离所组成的数据集应满足正态分布的特点[13], 且其均值为 0. 也就是说, 若一个窗口内的相邻点距离集合为 $d_i=\{dist(T_i,T_{i+1}),dist(T_{i+1},T_{i+2}),\ldots,dist(T_{i+w-2},T_{i+w-1})\}$, 则首尾数据点的距离应满足均值为$\mu=0$ 且方差 $\sigma=\sigma_{d_i}$ 的正态分布. 按照 3σ原则, 首尾数据点的距离应在 $[0,3\sigma_{d_i}]$ 范围之内.

然而, 考虑到数据集 $d_i=\{dist(T_i,T_{i+1}),dist(T_{i+1},T_{i+2}),\ldots,dist(T_{i+w-2},T_{i+w-1})\}$为相邻点的距离, 属于差分计算, 因此, 相比标准差, 绝对中位差更加符合数据集的特点, 且绝对中位差已有较为成熟的近似算法加快计算[14]. 因此, 本节使用窗口内相邻数据点的距离绝对中位差作为σ, 首尾点的距离应在$[-3\sigma,3\sigma]$之间. 指定窗口的最小时长约束 w, 若窗口的最终点和起始点的距离在 $[0,3\sigma_{d_i}]$ 范围之内, 则窗口闭合; 否则, 继续调大窗口, 直到窗口大小为 $2w$ 为止(若窗口大小超出 $2w$, 则挖掘出的对称子序列可能覆盖多个不重叠对称子序列, 使得最终结果不精确).

算法 4. 自适应窗口算法.

输入: 时间子序列起始点 T_i, 长度约束 w.

输出: 调整后的窗口大小.

1: $\sigma=sigma(T_i,w)$ //计算窗口内相邻数据点距离的方差
2: $t=w$
3: **while** $i+t<|T|$ && $t<2w$ && $dist(T_i,T_{i+t-1})>3\sigma$ **do**
4: t++;
5: **end while**;
6: **return** t

6 实 验

为验证本文所提出的时间序列对称模式匹配方法, 本节将选择多个数据集, 根据相应的评价标准进行实验评估, 同时将实验结果与多个现有方法进行对比. 具体实验环境、实验数据集、评价标准、现有对比方法以及实验结果如下所述.

6.1 实验设置

- 实验环境

本文使用 Python 语言在如下环境下对各部分内容进行实现, 处理器为 2.8 GHz Intel Core i7, 内存为 8 GB 2 400 MHz SODIMM.

- 实验数据

本文采用 1 个人工数据集和 3 个真实数据集进行实验. 人工数据集包含 30 000 个数据点, 除部分噪声影响之外, 原始数据为以 360 个点为一个完整周期的正弦函数采样数据. 真实数据集 1 为 GPS 轨迹数据, 共 33 500 个数据点, 主要采集了运输车运输货物 5 天内的行驶轨迹, 结合实际情况以及数据采集信息. 由于车速和运输路线的不同, 运输车运输一趟的平均时间为 30 分钟, 最长为 48 分钟. 为了能够挖掘到所有的对称模式,将时间窗口大小设置为 30 分钟. 真实数据集 2 为等时间间隔采样的挖掘机挖斗数据, 共 35 189 个数据点, 主要采集了挖掘机在挖掘过程中各个工况的变化. 由于挖掘机每次挖掘的坑道不一致, 且两次作业间存在等待期, 导致挖掘机工况对称模式持续的时间长度变化较大. 可通过设定时间窗口大小为 8 分钟, 以过滤噪声数据, 从而挖掘出合格的对称模式. 真实数据集 3 为股票数据, 由于股票数据具有长期增长的趋势性特征, 其对称模式出现较少, 可通过人工标注对称模式作为正确结果. 实验采用的 4 个数据集各有特点: 人工数据集具有明显的周期性; 运输车轨迹数据集运行时间较长, 对称性较为明显; 挖掘机工况数据集停机时间较长, 噪声较多, 对称性不易挖掘; 股票数据集没有明显的对称性和对称模式. 若算法在这 4 类数据集上的表现均比较良好, 则证明了对称模式挖掘算法的有效性.

6.2 评价标准

虽然时间序列模式包含的数据点个数或多或少，但是给定时间序列中模式的个数通常是固定的. 因此，本文采用对称模式的数量偏差评价真实结果和挖掘结果的相近程度. 令 x_{truth} 作为时间序列对称模式的真实数量，x_{cal} 作为挖掘出的时间序列对称模式的数量. 为了评估挖掘结果和真实结果的相似程度，令$\Delta(x_{truth},x_{cal})=\frac{|x_{cal}-x_{truth}|}{x_{truth}}$作为真值 x_{truth} 和挖掘结果 x_{cal} 的误差率. $\Delta(x_{truth},x_{cal})$越小，即挖掘结果越精确.

此外，考虑到自适应窗口算法致力于挖掘出对称模式的所有信息，因此，在与其他算法进行对比时，自适应窗口算法应当使用对称模式的数据完整性作为衡量指标. 令 s_{truth} 为真实对称模式所包含的数据点集合，s_{cal} 为挖掘出的对称模式所包含的数据点集合，参考分类和聚类算法中精确率[15]的定义，本文使用$\Delta(s_{truth},s_{cal})=\frac{|s_{cal}\cap s_{truth}|}{|s_{truth}|}$作为对称模式数据信息完整性的评价标准. $\Delta(s_{truth},s_{cal})$越大，挖掘的对称模式数据完整性越高.

6.3 现有方法

本文新提出的对称模式挖掘方法将与现有的多种模式挖掘方法进行比较，包括基于 Fast-DTW[16]算法的全局对称模式挖掘算法、基于 Matrix Profile[4]的对称模式挖掘算法和基于时间序列分解的 STL[17]算法等.

6.4 实验结果

本文选择 4 个数据集来对挖掘结果进行验证，并对各数据集提供了多种对称模式挖掘方法在不同窗口长度和距离阈值下的对称模式准确性和时间开销结果. 此外，本文还比较了不同数据集在自适应窗口下与其他挖掘算法的完整性结果对比. 具体实验结果如下文所述.

(1) 对称模式挖掘算法

合成数据集、运输车数据集和挖掘机数据集以秒或毫秒为单位采集数据，数据量较大，本文选择了数据点数从 3 k 到 30 k 不等的数据量. 而股票数据集以天为单位采集数据，数据量较小，故本文选择了数据点数从 0.3 k 到 3 k 不等的数据量(如图 11 所示). 如图 12 所示，通过对比不同数据集和不同数据量下所有算法识别对称模式的误差率可以发现：本文提出的对称模式挖掘算法的误差率最低，可以挖掘出和真实数据最接近的对称模式.

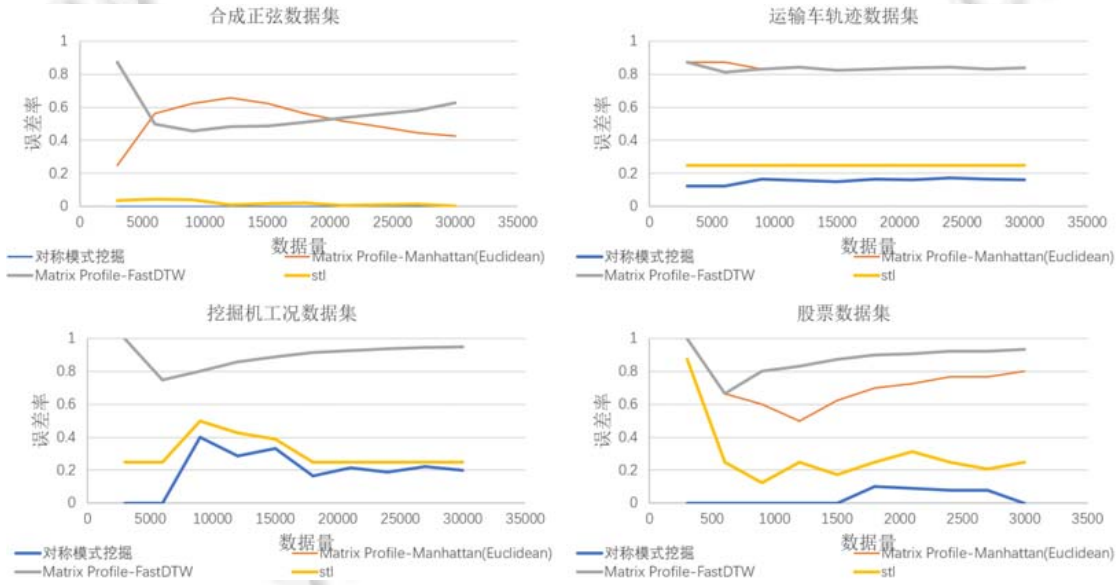


图 11 多数据集不同数据量下的对称模式挖掘效果

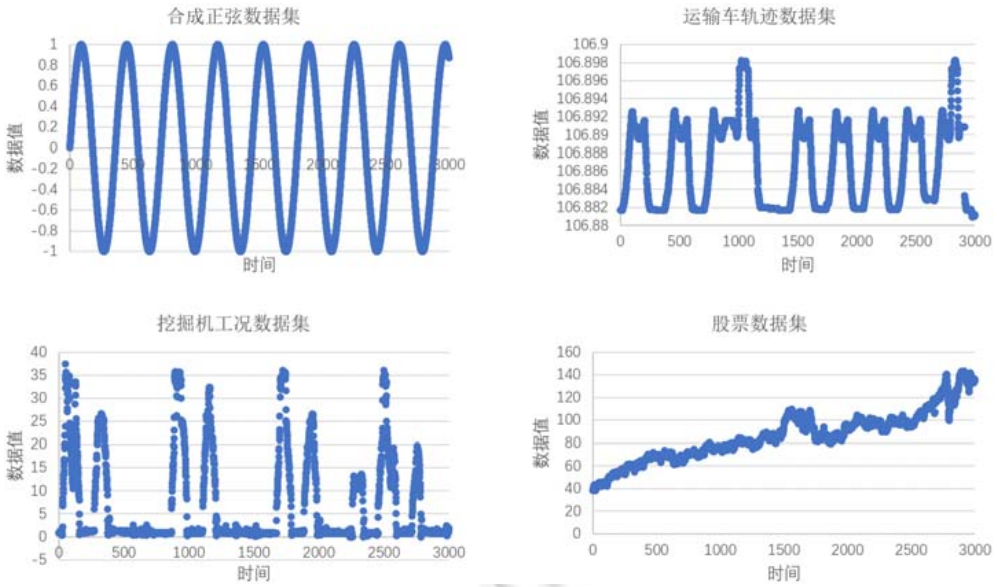


图 12　多数据集时间序列采样结果

结合图 11 和图 12 的采样结果可知, 无论是在具有周期性的正弦数据集, 还是具有较多噪声的挖掘机工况数据集, 抑或不具有稳定对称模式的股票数据集, 本文提出挖掘算法的误差率均远低于其他算法. 原因在于: 本文的对称模式挖掘算法采用的对称性度量方式为每个数据点选择了最佳的匹配点, 从而计算出的对称度较小; 同时, 根据自然断点确定合适对称度阈值, 保证能挖掘出与真实结果数量相当的对称模式. 此外还可以发现, 对称模式挖掘算法的误差率随着数据量增大而逐渐稳定在某一范围,证明了算法在大数据量下识别对称模式的稳定性和有效性.

图 13 通过在 4 个数据集上分别用不同的算法挖掘对称模式所消耗的时间, 来评价各个算法的时间性能.

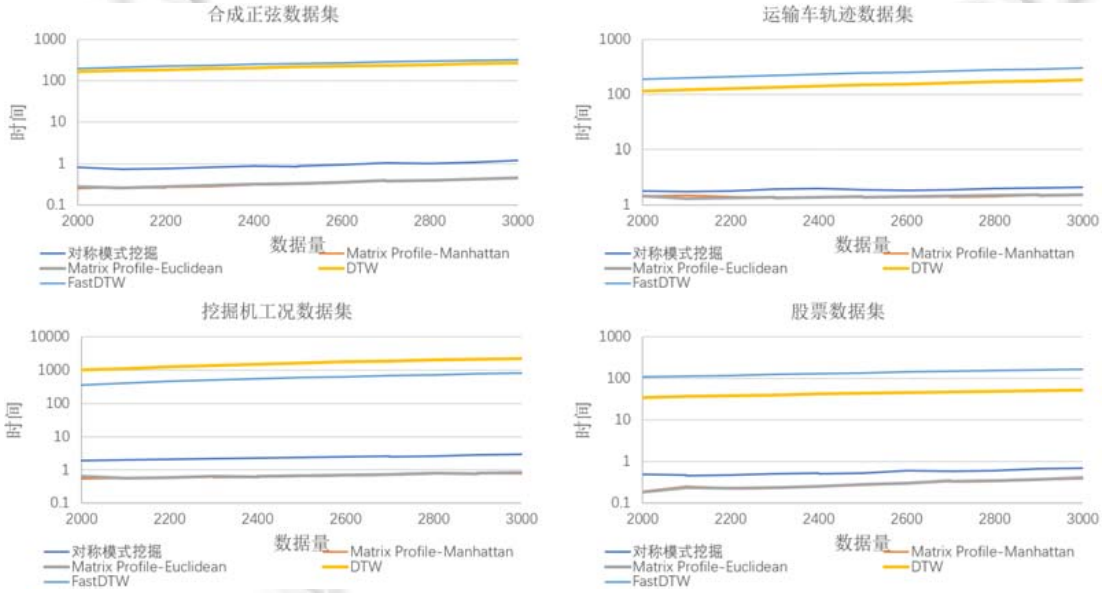


图 13　多种算法时间性能对比

由图 13 可以发现: 本文提出的对称模式挖掘算法与基于曼哈顿距离和欧几里德距离的 Matrix Profile 算

法的时间性能几乎一致，远高于全量时间子序列DTW距离度量和FastDTW算法，具有较高的可用性，在真实工业场景中具有较高的实用价值.

(2) 自适应窗口

时间序列的数据和模式均具有很强的随机性，同一个时间序列中很可能存在不同的模式. 为保证尽可能完整地挖掘出所有的对称模式，在第 5 节引入自适应窗口来动态地调整窗口大小，以增强挖掘出对称模式的完整性. 如图 14 所示，增加了自适应窗口的对称模式挖掘方法比原始方法和其他方法所挖掘出对称模式的完整性更高，其对称模式的信息更加全面.

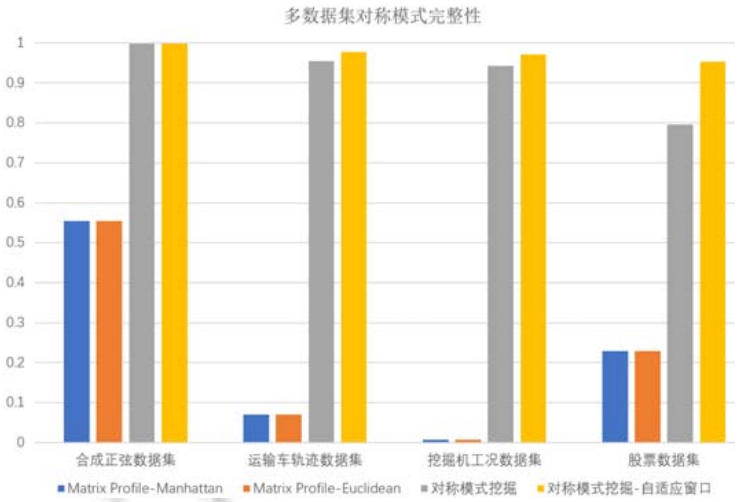


图 14 多数据集多算法对称模式完整性对比

7 相关工作

随着信息技术的普及和发展，各行各业，尤其是工业领域，通过相应的传感器和信息系统积累了大量的时间序列数据，对特定领域或者特定模式的时间序列数据，利用数学建模、机器学习等方法,进行建模和分析，从而得到特定类型的时间序列子模式，是一项非常有价值的研究课题. 本文主要研究时间序列中蕴含的对称模式，诸多业内研究者针对时间序列模式挖掘问题提出了多种计算和优化方法.

7.1 基于相似性检索的模式挖掘方法

文本领域中，相似性检索方法在包括社区发现、重复检测、聚类和查询细化都有应用[18,19]. Rakthanmanon 等人将相似性检索算法扩展到了时间序列领域，通过计算时间序列子序列的全对相似性，识别蕴含在时间序列中的所有模式[4]. 该算法使用滑动窗口截取了时间序列的全部子序列，使用归一化的欧几里德距离度量两个时间子序列的距离. 为了能够加速计算，提升算法的时间效率，该算法被设计成了迭代算法，每次随机抽取一个时间子序列，计算其和全部其他时间子序列的欧几里德距离，从而更新距离最小的时间子序列对的信息. 为了利用所有时间子序列中大量的重叠信息，该算法使用离散傅里叶变换算法对原始时间序列进行逆变换，进而计算选中子序列和完整时间序列的距离. 该方法具有较强的可扩展性，对于超大规模的数据集，可以随时计算近似结果[20].

基于相似性检索的模式挖掘方法在对称模式挖掘上具有较大的局限，该方法只能计算出任意两对时间子序列的相似度，却无法判断其是否属于对称子序列. 如需挖掘对称模式，当在 Matrix Profile 算法计算过程中加入子序列对称性的判断，降低算法的时间效率. 且 Matrix Profile 算法使用欧氏距离度量两个时间子序列的相似性，由第6节的实验可知，欧几里德距离度量时间序列的相似性忽略了子序列的位置信息，从而导致识别不准，在精确率上不如本文设计的对称模式挖掘算法.

7.2 基于时间序列分解的模式挖掘方法

在时间序列分析和预测领域, 时间序列分解是一种常用的方法. STL (seasonal-trend decomposition procedure based on loess)[19]时间序列分解方法基于 LOESS[21]将某时刻的数据分解为趋势分量、周期分量和余项, 从而挖掘时间序列中蕴含的周期信息和趋势信息. 与经典的时间序列分解算法[22]相比, STL 并不依赖于移动平均模型, 它允许对过程的属性进行分析, 对于大量的趋势和季节性的平滑也可以进行快速计算; 并且可以灵活地调整季节项的周期, 而不会被数据中的异常行为扭曲.

然而, 基于时间序列分解的模式挖掘方法仅可用于挖掘对称模式为季节模式的时间序列, 不能挖掘任意类型的对称模式. 真实工业场景中的时间序列数据类型复杂, 变化较多, 对称模式并不一定呈季节性出现, 更可能存在季节性对称模式和非季节性对称模式随机出现的情况. 对于该类型的时间序列数据, 使用 STL 算法进行计算, 往往就不能充分地挖掘出所有的对称模式, 或者挖掘出错误的对称模式. 比如第 6 节实验中的股票数据, 因为具有较强的趋势性, 其对称模式较少. 然而, STL 依旧挖掘出了大量的对称模式, 与正确结果存在较大差别.

8 结 论

考虑到一般情况下时间序列中数据采样的频率和持续时间的随机性, 本文根据动态时间规整的算法思想, 提出了给定时间窗口约束内的对称模式动态规划挖掘算法. 计算出所有时间子序列的对称度之后, 根据对称度的分布特征确定自然断点(即对称度阈值), 即可根据贪心算法求得所有的对称模式. 此外, 对于具有多种对称模式的时间序列, 只能选择最短对称模式的时间跨度为固定时间窗口, 该窗口会使得较长的对称模式信息不能被完整地挖掘. 因此, 本文设定了一种可自适应调整窗口的策略, 根据起止点的差值是否在正常的范围内确定窗口是否需要扩展, 进而较为完整地挖掘出所有的对称模式. 为对上述所提挖掘方法进行验证, 本文通过一个人工数据集、3 个真实数据集(运输车轨迹数据集、挖掘机工况数据集以及股票数据集)来对本文方法及其他现有方法进行实验. 由实验结果可知, 与现存其他对称模式挖掘方法相比, 本文所提出的基于 DTW 模型下的动态规划对称模式挖掘方法挖掘出的对称模式数量最接近正确值, 从而误差率最小; 且根据自适应窗口可应对较为复杂的数据状况, 挖掘出不同种类和长度的对称模式; 此外, 本方法在运行性能时间开销方面也较为理想, 远低于 FastDTW 等全局时间序列模式挖掘算法.

References:

[1] Mueen A, Keogh EJ, Zhu Q, Cash S, Westover MB. Exact discovery of time series motifs. In: Proc. of the SDM. 2009. 473−484.

[2] Yeh CCM, Zhu Y, Ulanova L, Begum N, Ding YF, Dau HA, Silva DF, Mueen A, Keogh EJ. Matrix profile I: All pairs similarity joins for time series: A unifying view that includes motifs, discords and shapelets. In: Proc. of the ICDM. 2016. 1317−1322.

[3] Ding H, Trajcevski G, Scheuermann P, Wang XY, Keogh EJ. Querying and mining of time series data: Experimental comparison of representations and distance measures. Proc. of the VLDB Endow, 2008, 1(2): 1542−1552.

[4] Rakthanmanon T, Campana BJL, Mueen A, Batista GEAPA, Westover MB, Zhu Q, Zakaria J, Keogh EJ. Searching and mining trillions of time series subsequences under dynamic time warping. In: Proc. of the KDD. 2012. 262−270.

[5] Song SX, Zhang AQ, Wang JM, Yu PS. SCREEN: Stream data cleaning under speed constraints. In: Proc. of the SIGMOD Conf. 2015. 827−841.

[6] Gao F, Song SX, Wang JM. Time series data cleaning under multi-speed constraints. Ruan Jian Xue Bao/Journal of Software, 2021, 32(3): 689−711 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/6176.htm [doi: 10.13328/j.cnki.jos.006176]

[7] Hishinuma T, Hasegawa H, Tanaka T. SIMD parallel sparse matrix-vector and transposed-matrix-vector multiplication in DD precision. In: Proc. of the VECPAR. 2016. 21−34.

[8] Wan SP. An efficient implementation of Manacher's algorithm. CoRR abs/2003.08211, 2020.

[9] Song SX, Zhu H, Wang JM. Constraint-variance tolerant data repairing. In: Proc. of the SIGMOD Conf. 2016. 877−892.

[10] Ester M, Kriegel HP, Sander J, Xu XW. A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proc. of the KDD. 1996. 226−231.

[11] Jenks GF. The Data Model Concept in Statistical Mapping. In: Int'l Yearbook of Cartography, Vol.7, 1967. 186−190.

[12] Cheng YZ. Mean shift, mode seeking, and clustering. IEEE Trans. on Pattern Analysis and Machine Intelligence, 1995, 17(8): 790−799.

[13] Butcher JC. Random sampling from the normal distribution. The Computer Journal, 1961, 3(4): 251−253.

[14] Chen ZW, Song SX, Wei ZH, Fang JY, Long J. Approximating median absolute deviation with bounded error. Proc. of the VLDB Endowment, 2021, 14(11): 2114−2126.

[15] Sammut C, Webb GI. Encyclopedia of Machine Learning and Data Mining. Boston: Springer, 2017.

[16] Salvador S, Chan P. Toward accurate dynamic time warping in linear time and space. Intelligent Data Analysis, 2007, 11(5): 561−580.

[17] Cleveland, Robert B, Cleveland WS, Terpenning I. STL: A seasonal-trend decomposition procedure based on loess. Journal of Official Statistics, 1990, 6(1): 3−73.

[18] Yoon CE, O'Reilly O, Bergen KJ, *et al*. Earthquake detection through computationally efficient similarity search. Science Advances, 2015, 1(11): Article No.e1501057.

[19] Bayardo RJ, Ma Y, Srikant R. Scaling up all pairs similarity search. In: Proc. of the WWW. 2007. 131−140.

[20] Zhu Y, Zimmerman Z, Senobari NS, Yeh CCM, Funning GJ, Mueen A, Brisk P, Keogh EJ. Matrix profile II: Exploiting a novel algorithm and GPUs to break the one hundred million barrier for time series motifs and joins. In: Proc. of the ICDM. 2016. 739−748.

[21] Hastie T, Tibshirani R, Friedman JH, *et al*. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. New York: Springer, 2009.

[22] Dagum EB, Bianconcini S. Seasonal Adjustment Methods and Real Time Trend-cycle Estimation. Berlin, Heidelberg: Springer, 2016.

附中文参考文献:

[6] 高菲, 宋韶旭, 王建民. 多区间速度约束下的时序数据清洗方法. 软件学报, 2021, 32(3): 689−711. http://www.jos.org.cn/1000-9825/6176.htm [doi: 10.13328/j.cnki.jos.006176]

李盼盼(1996－), 男, 硕士生, 主要研究领域为时间序列模式挖掘.

宋韶旭(1981－), 男, 博士, 副教授, 博士生导师, CCF 专业会员, 主要研究领域为数据库, 数据质量, 时序数据清理, 大数据集成.

王建民(1968－), 男, 博士, 教授, 博士生导师, CCF 高级会员, 主要研究领域为数据库, 工作流, 大数据与知识工程.