基于领域偏好的可变时间窗口时序数据主题模式识别算法

王一博[1,2] 彭广举[1,2] 何远舵[1,2] 王亚沙[1,3] 赵俊峰[1,2] 王江涛[1,2]

(高可信软件技术教育部重点实验室(北京大学) 北京 100871)[1]

(北京大学信息科学技术学院 北京 100871)[2] (北京大学软件工程国家工程研究中心 北京 100871)[3]

摘 要 随着传感器的普及,智慧城市、普适计算等领域应用不断涌现,对时序数据处理的需求也在不断增长。时序数据中反复出现的高度相似的模式被称为主题模式。时序数据的主题模式蕴含有了大量的信息,对主题模式的识别是时序数据处理的重要分支领域。现有主题模式识别算法无法根据特定应用或领域的知识来指定主题模式识别的偏好,从而难以发现对分析领域问题最具价值的模式。针对这一问题,文中给出了一种可以根据领域偏好定义子序列相似性的机制,并设计了一种针对上述相似性度量机制的可变时间窗口主题模式识别加速剪枝算法。实验证明,所提方法在多个公开数据集上,能高效且准确地发现具有领域偏好的主题模式。

关键词 时序数据,主题模式,领域偏好,可变时间窗口,主题模式实例

中图法分类号 TP274 **文献标识码** A **DOI** 10.11896/jsjkx.191100505C

Time Series Motif Discovery Algorithm of Variable Length Based on Domain Preference

WANG Yi-bo[1,2] PENG Guang-ju[1,2] HE Yuan-duo[1,2] WANG Ya-sha[1,3] ZHAO Jun-feng[1,2] WANG Jiang-tao[1,2]

(Key Lab of High Confidence Software Technologies(Peking University),Ministry of Education,Beijing 100871,China)[1]

(School of Electronics Engineering and Computer Science,Peking University,Beijing 100871,China)[2]

(National Engineering Research Center for Software Engineering,Peking University,Beijing 100871,China)[3]

Abstract With the development of ubiquitous computing,more and more sensors are installed in our daily applications. As a result,the demand for time series data processing is very high. The similar pattern which appears in time series data several times are called time series motif. Motif contains huge amounts of information in time series data. Motif discovery is one of the most important work in motif analysis. State-of-art motif discovery algorithm cannot find proper motif based on domain knowledge. As a result,such algorithm cannot find most valuable motif. Aiming at this problem,this paper used domain distance to evaluate the similarities of subsequences based on domain knowledge. By using the new distance,this paper developed a branching method to discovery motif with variable length. Several data from real life are used to test the performance of the algorithm. The results show that the proposed algorithm can find motif with domain knowledge accurately.

Keywords Time series data,Motif,Domain knowledge,Variable time window,Motif example

1 引言

数值按其发生的时间先后顺序排列而成的数据,被称为时间序列数据(即时序数据)[1]。例如,传感器数据就是非常典型的时序数据。随着物联网和智慧城市建设的深入,时序数据分析的需求快速增长。

时序数据中彼此高度相似的子序列构成的集合被称为一个主题模式集合,而该集合中最具代表性的子序列被称为时间序列主题模式[2-3](下文简称主题模式),同时该集合中的任意一个子序列都被称为该主题模式的一个实例。主题模式(Motif)是时序数据研究的重点之一。相同主题模式的实例彼此相似,不同主题模式的实例具有较大差异。不同主题模式按照其所属的主题模式集合内部各元素的相似程度排序,前 K 个被称为此时间序列的前 K 个主题模式。以某家用电器用电量的时序数据为例(见图 1)。假设红色部分的 3 个子序列分别对应家用电器在 3 个不同时间段上的工作状态,分别用 $T_{1,m}$,$T_{2,m}$,$T_{3,m}$ 表示,这 3 个子序列是此时间序列中最相似的 3 个子序列。则根据定义可知,集合 $\{T_{1,m},T_{2,m},T_{3,m}\}$ 是主题模式集

到稿日期:2018-10-03 返修日期:2018-12-25 本文受国家自然科学基金重点支持项目(91546203),国家电网公司总部科技项目(JS71-16-005)资助。

王一博(1993-),男,硕士生,主要研究领域为普适计算、机器学习、数据挖掘;**彭广举**(1995-),男,硕士生,主要研究领域为普适计算、机器学习、数据挖掘;**何远舵**(1992-),男,博士生,主要研究领域为普适计算,数据挖掘;**王亚沙**(1975-),男,博士,教授,CCF 高级会员,主要研究领域为城市计算、数据分析、软件工程,E-mail:wangyasha@pku.edu.cn(通信作者);**赵俊峰**(1974-),女,博士,副教授,主要研究领域为软件工程、知识工程、大数据分析等;**王江涛**(1987-),男,博士,助理研究员,CCF 会员,主要研究领域为移动计算、群智感知、社会计算。

合，由于 $T_{1,m}$ 与 $T_{2,m}$，$T_{3,m}$ 都十分相似，可以取其中任意一个作为具有代表性的实例，即为此时序数据的一个主题模式。

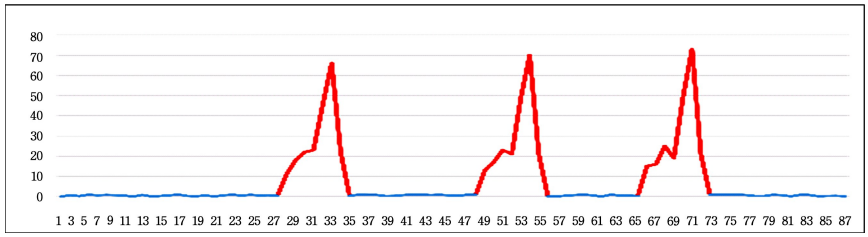

图 1 家用电器用电量时序图

Fig. 1 Time series diagram of electric consumption of a household appliance

主题模式蕴含了时序数据中最具分析价值的信息，在不同领域均有重要的应用[4-8]。例如：心电时序数据的主题模式可以帮助医生发现重复出现的心动模式，并检测出心动异常[6]；用电量时序数据的主题模式标识着电器的不同工作状态，可以帮助研究者了解不同电器的电能消耗情况，为节能提供了决策支持[9]。同时，主题模式也是很多时序数据分析工作的基础：如时序数据的规则发现[9]、时序数据异常检测[6]、时序数据降维等。因此，从时序数据中发现主题模式算法具有十分重要的研究价值和实用价值。

然而，现有的主题模式发现算法仍存在一些不足。固定时间窗口主题模式识别算法要求对主题模式的长度有先验知识，难以适应很多实用场景，应用较少。现有主流可变时间窗口主题模式识别算法，一般采用标准化欧氏距离定义模式之间的差异，无法根据特定应用或领域的知识，来指定主题模式识别的偏好。例如：在活动识别应用中，只有具有明显波动的加速度传感器数据子序列才表示用户在执行某种动作，而波动较小的子序列往往是由噪声引起的。活动识别应用要求主题模式识别算法发现那些具有明显波动的相似子序列，而不是波形相似但波动性不大的子序列。传统算法由于无法根据应用需求调整对子序列相似性的度量方法，因此会导致在一些应用场景中效果不佳。另外，大多数现有的可变时间窗口主题模式识别算法并不区分主题模式和主题模式实例这两个概念，这会使得算法返回的主题模式在很多情况下是同一个主题模式的不同实例，对大量需要分析不同形状的多个主题模式的应用场景支持不足。

针对这些问题，本文给出了一种可以根据领域偏好定义子序列相似性的机制，并设计了一种针对上述相似性度量机制的可变时间窗口主题模式识别加速剪枝算法。具体包括 3 方面工作：

1）定义包含领域偏好的子序列之间距离的度量方法。在子序列的距离度量中考虑领域偏好，可使得符合偏好的主题模式更容易被算法识别出来。例如：在活动识别领域，设置偏好可以增加波动较小的子序列之间的距离，从而使得具有明显波动且重复出现的子序列更容易被算法识别为主题模式。

2）针对上述包含领域偏好的子序列之间距离的度量方法，给出可变时间窗口主题模式发现剪枝方法，以提高算法的执行效率。

3）设计基于实例聚合的主题模式识别算法，建立相似主题模式之间的关联关系和统一抽象表示，从而方便发现形状不同的多个主题模式。

2 相关工作

2.1 相关概念

本节主要介绍时序数据主题模式识别常见的概念，包括时序数据、主题模式、主题模式实例、子序列等的准确定义。

定义 1 时间序列 T 是指将关于时间的实数值 t_i 按照时间顺序依次排列，$T = t_1, t_2, \cdots, t_n$。其中 n 为时间序列的长度。

定义 2 时间序列 T 的子序列 $T_{i,m}$ 由从第 i 个位置开始的连续 m 个实数值构成。其中 m 为时间序列子序列的时间窗口长度。

在关于子序列相似性的度量标准上，文献[10]提出相似性度量需要在欧氏距离的基础上进行 Z-score 标准化(z-normalization)：分别对时间序列 X 和 Y 进行 Z-score 标准化，然后再计算标准化以后的距离，这个距离被称为标准化欧氏距离。

定义 3 对于时间序列 T 在时间窗口长度为 m 下的子序列 $T_{i,m}$，其最近邻距离为时间序列 T 中所有时间窗口长度为 m，并且与子序列 $T_{i,m}$ 不重叠的子序列到子序列 $T_{i,m}$ 的距离的最小值。

定义 4 时间序列 T 的时间窗口长度为 m 的彼此高度相似的子序列组成的集合被称为时间序列 T 的一个主题模式集合。主题模式集合中最近邻距离最小的子序列为时序数据的主题模式。主题模式集合中每一个元素为主题模式的一个实例。相同的主题模式实例两两相似，不同的主题模式实例两两具有较大差异。不同的主题模式集合按照主题模式集合内部元素的相似程度，即集合元素最小最近邻距离排序。时序数据前 K 个主题模式为前 K 个代表集合内部相似程度最高的主题模式集合的子序列。

按照主题模式时间窗口长度是否固定，主题模式发现算法可分为固定时间窗口主题模式发现算法和可变时间窗口主题模式发现算法。

2.2 固定时间窗口长度的主题模式发现算法

固定时间窗口长度的主题模式识别算法，假定主题模式的时间窗口长度是相同的并且已知且固定。该算法的基本思想是：为了寻找时间窗口长度为 m 的主题模式，计算长度为 n 的时间序列 T 中所有长度为 m 的子序列间的相似性，其中相似性最高的子序列对被称为时间序列 T 的主题模式。之后的研究主要集中在如何对算法的效率进行优化，例如：文献[11]利用剪枝来提高效率；文献[12]利用矩阵轮廓来提高效率等。

然而，基于固定时间窗口长度的时序数据主题模式识别算法要求主题模式的时间窗口长度是固定且已知的，这一假设在现实中几乎不成立。一方面，人们很难提前准确地知道主题模式时间窗口长度的大小；另一方面，不同主题模式的时间窗口长度也不一定一致。因此，固定时间窗口长度的主题模式算法在现实中依然很难被直接使用。

2.3　可变时间窗口长度的主题模式发现算法

可变时间窗口长度的主题模式识别算法，允许主题模式时间窗口变化。一个基本算法是用固定时间窗口长度的主题模式识别算法来计算每一个时间窗口长度下的主题模式。然而，这种算法的时间复杂度过高，在数据量比较大的情况下耗时过长。

文献[14]在此基础上提出了一种基于启发式剪枝的改进算法，称为 MOEN 算法。该算法基于一个观察：对于同一个时序数据，在时间窗口长度差别不大时，主题模式的位置比较接近。因此剪枝判断的核心就是检查时间序列中最优的那部分子序列在下一个时间窗口长度下是否仍然是最优的。

具体来说，文献[14]推导出了一个关于时间窗口长度的距离下界公式。

假设时序数据 T 的两个长度为 $m-1$ 的子序列 $T_{i,m-1}$，$T_{j,m-1}$ 之间的标准化欧氏距离为 d，那么在时间窗口长度为 m 时，两个子序列 $T_{i,m}$，$T_{j,m}$ 的距离下界可用下式计算：

$$d_{LB}=\left(\frac{m-1}{m}+\frac{m-1}{m^2}z^2\right)^{-1}d^2 \tag{1}$$

其中，z 的计算式如下：

$$z=\frac{\max\limits_i(t_i-\mu_{i-m+1,m-1})}{\sigma_{i-m+1,m-1}} \tag{2}$$

从距离下界公式可以看出，该公式是关于距离 d 的单调函数。因此距离下界的单调性可以帮助我们进行剪枝。因为篇幅原因，更多的细节请参考文献[14]。MOEN 算法能够加速计算的本质原因是：其利用前一个时间窗口的信息避免了不必要的遍历搜索，很大程度上解决了可变时间窗口长度的主题模式识别算法的效率问题。但该算法仍然存在 3 个方面的问题：

1）算法找到的主题模式具有极强的“简易化倾向”，也就是算法倾向于寻找变化较小的子序列作为主题模式。然而在很多时候，变化较小的子序列是由噪音引起的，没有实际意义，而变化相对较大的子序列往往是具有实际意义的主题模式。

2）算法无法将一些简单的上下文信息加入到时间序列主题模式的寻找中，即很难结合领域知识对时间序列主题模式进行寻找。例如：在石油压力的时序数据中，具有上升趋势的子序列更有研究意义，因此石油研究者期待找到具有上升趋势的主题模式。然而，目前的时序数据主题模式识别算法只能找到通用的主题模式，无法根据该领域知识定向地寻找相关主题模式。

3）对于主题模式定义带来的问题，算法很有可能找到的是同一个主题模式的不同实例。文献[14]采用的主题模式的定义是主题模式对，算法会将这些不同的实例作为不同的主题模式返回。例如：图 2 为在某家用电器用电量的时序图中寻找主题模式时，采用 MOEN 算法找到的前 30 个主题模式对的结果。可以看到，返回结果其实仅仅是同一个主题模式在不同位置的主题模式实例，而不是不同的主题模式。

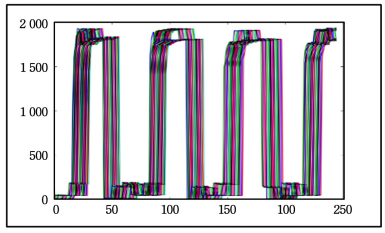

图 2　MOEN 算法找到的某家用电器前 30 的主题模式图

Fig. 2　Top 30 motifs of household appliance found by MOEN algorithm

针对 MOEN 算法存在的问题，本文设计了一种全新的基于领域偏好的主题模式距离度量方式，给出了基于领域偏好主题模式距离下界的计算方法，并根据新的距离下界重新设计了剪枝方案，最后给出了根据主题模式的相似程度进行主题模式实例聚合的算法，最终实现了可定义领域偏好的变长时间窗口主题模式识别算法。

3　算法设计

本文在 MOEN 算法的基础上，设计了基于领域偏好的可变时间窗口长度下的主题模式识别算法。该算法的整体流程如图 3 所示。该算法须给定一个时间长度搜索范围，如（m，mx）。首先遍历计算时间窗口长度为 m 时，每个子序列之间的距离以及基于领域偏好的距离下界。在时间窗口长度加 1 的情况下，算法优先根据距离下界判断在该时间窗口长度下寻找主题模式的过程是否能够剪枝。如果能够剪枝，则更新距离下界，并且保存结果；如果不能够剪枝，则重新遍历计算该时间窗口长度下基于领域偏好的距离。

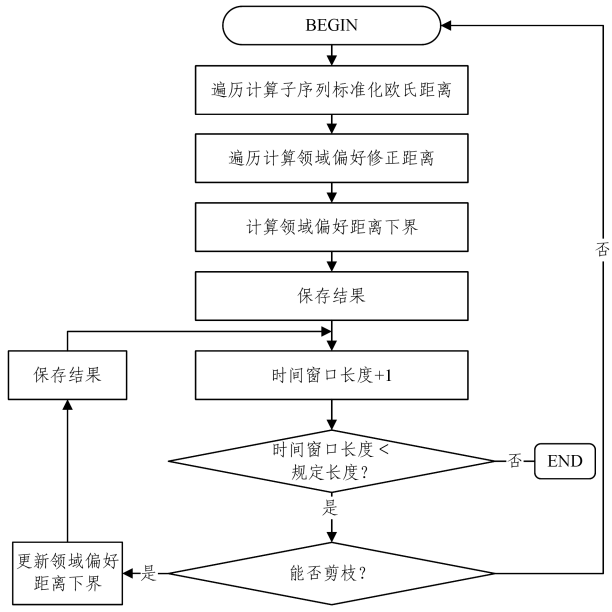

图 3　基于领域偏好的可变时间窗口长度下的主题模式识别算法流程

Fig. 3　Flowchart of time series motif discovery algorithm of variable length based on domain preference

从算法的整体流程可以看出,算法的难点是解决以下 3 个问题:

1)如何定义基于领域偏好的主题模式距离度量。更重要的是,如何保证新的距离能够有效地将领域偏好反映到距离度量中,并且不破坏原始算法的复杂度。

2)如何将基于领域偏好的主题模式距离度量应用到距离下界公式中。因为现有的对于距离下界的一切推导都是基于标准化的欧氏距离的,而且距离下界公式并不能直接应用到新的距离度量中,因此需要给出基于领域偏好距离下界的计算方法,并设计基于领域偏好距离下界的剪枝方案。

3)如何进行主题模式实例的筛选以保证得到不同的主题模式。目前大部分主题模式识别算法都是用主题模式实例来代替主题模式,避开了对找到的主题模式高度相似的问题的讨论。但是本文希望能够通过算法对找到的主题模式实例进行相似性判断,从而找出不同的主题模式。

3.1 基于领域偏好的主题模式距离度量

基于领域偏好的主题模式距离,在固定时间窗口长度下,Dau 在文献[15]中给出了一个总体框架。该框架是基于矩阵轮廓的[12]。矩阵轮廓是为解决固定时间窗口长度下的主题模式识别问题而开发的一种算法,其核心思路是通过快速傅里叶变换的方式来加速计算不同子序列之间标准化欧氏距离的过程,最终返回每一个子序列的最近邻距离。

定义 5 对于时序数据 T 的任意一个时间窗口长度为 m 的子序列 $T_{i,m}$。假设通过算法已经求得其最近邻距离为 $d_{i,m}$。本文定义基于领域偏好的最近邻距离表达式如下:

$$CONTEXT(d_{i,m})=d_{i,m}+(1-AD_i)*\max_j(d_{j,m}) \tag{3}$$

其中,$CONTEXT(d_{i,m})$为子序列 $T_{i,m}$ 基于领域偏好的最近邻距离,又称为最近邻领域偏好距离。$d_{i,m}$为子序列 $T_{i,m}$ 的最近邻距离,AD_i 为子序列 $T_{i,m}$ 的一个最近邻领域偏好系数。$AD_i\in(0,1)$,AD_i 越接近 1,说明子序列 $T_{i,m}$ 和领域偏好所期待的子序列特征越接近。AD_i 越接近 0,则说明子序列 $T_{i,m}$ 和领域偏好所期待的子序列特征越远。$\max_j(d_{j,m})$为所有子序列最近邻距离中的最大距离。

可以看到,基于领域偏好的最近邻距离的本质就是在最近邻距离的基础上加了一个惩罚项:$(1-AD_i)*\max_j(d_{j,m})$。本文称这个惩罚项为惩罚距离。惩罚距离表示了子序列自身的特征与领域知识的契合程度。当子序列的特征与领域知识极为接近时,$1-AD_i=0$,即完全不惩罚该子序列;当子序列的特征与领域知识相差极大时,$1-AD_i=1$,即给原来的子序列一个相对较大的惩罚。通过这个方式,本文可以将领域知识量化成基于领域知识的惩罚项,从而实现距离计算。

对于子序列 $T_{i,m}$ 的最近邻领域偏好系数 AD_i 的计算方法,总体上分为 2 步:1)计算子序列 $T_{i,m}$ 的领域特征值;2)对同一时间窗口所有领域特征值进行归一化,从而得到该时间窗口的领域偏好系数。

3.2 基于领域偏好的主题模式距离度量

在采用了基于领域偏好的最近邻距离度量方式后,本文进一步探讨了如何计算基于领域偏好的最近邻距离下界和基于新的距离下界的剪枝方案。

3.2.1 基于领域偏好的子序列最近邻距离的概念

下面给出基于领域偏好的最近邻距离下界的概念和计算方法,并证明这种定义的正确性。

定义 6 对于时序数据 T 的任意一个时间窗口长度为 m 的子序列 $T_{i,m}$,假设通过算法已经求得其最近邻距离为 $d_{i,m}$,其对应的距离下界 $d_{LB(i,m)}$ 根据式(1)计算。本文定义基于领域偏好的最近邻距离下界的表达式如下:

$$D_{LB(i,m)}=d_{LB(i,m)}+(1-AD_i)*\max_j(d_{j,m}) \tag{4}$$

其中。$D_{LB(i,m)}$为子序列 $T_{i,m}$ 基于领域偏好的最近邻距离下界,$d_{j,m}$为子序列 $T_{j,m}$ 的最近邻距离,AD_i 为子序列 $T_{i,m}$ 的一个最近邻领域偏好系数,$AD_i\in(0,1)$。AD_i 越接近 1,说明子序列 $T_{i,m}$ 与领域偏好所期待的子序列特征越接近;AD_i 越接近 0,说明子序列 $T_{i,m}$ 与领域偏好所期待的子序列特征越远。$\max_j(d_{j,m})$为所有子序列最近邻距离中的最大距离。

同时,为了区分,本文称式(4)右边的第一项 $d_{LB(i,m)}$ 为最近邻欧氏距离下界,第二项 $(1-AD_i)*\max_j(d_{j,m})$ 为惩罚距离。

下面证明为何式(4)给出的距离下界能够成为基于领域偏好的最近邻距离下界。

首先,根据最近邻距离的定义,对于长度为 n 的时序数据 T 的任意一个子序列 $T_{i,m}$,假设已经求得其最近邻距离为 $d_{i,m}$。那么,对于除 $T_{i,m}$ 外的任意一个子序列,假设其到 $T_{i,m}$ 的距离为 d(标准化欧氏距离),根据最近邻距离的定义有:

$$d\geqslant d_{i,m} \tag{5a}$$

在时序数据和时间窗口长度都不变的情况下,式(1)具有单调性。因此,假设距离 d 对应的距离下界为 d_{LB},$d_{i,m}$ 对应的距离下界为 $d_{LB(i,m)}$,有:

$$d_{LB}\geqslant d_{LB(i,m)} \tag{5b}$$

根据定义(5)中对基于领域偏好的距离的定义,对于同一个子序列,其领域偏好的惩罚距离是相同的。因为惩罚距离由领域特征值来确定,而领域特征值的计算只与子序列本身的特征相关,所以对于同一个子序列的不同子序列对,其惩罚距离是相同的。在式(5b)的两边加相同的惩罚距离,可以得到:

$$d_{LB}+(1-AD_i)*\max_j(d_{j,m})\geqslant d_{LB(i,m)}+(1-AD_i)*\max_j(d_{j,m}) \tag{5c}$$

将式(3)代入式(5c)左侧,将式(4)代入式(5c)右侧,则得到:

$$CONTEXT(d_{i,m})\geqslant D_{LB(i,m)} \tag{6}$$

将推导出的式(6)用语言来描述:

对于长度为 n 的时间序列 T 的任何一个子序列 $T_{i,m}$,其到任何时间窗口长度为 m 的子序列的基于领域偏好的距离均不小于按照定义(6)给出的该子序列的基于领域偏好的最近邻距离下界。因此,基于领域偏好的最近邻距离下界的定义和计算是正确的。

3.2.2 基于子序列最近邻领域偏好距离下界的剪枝方案

基于领域偏好的最近邻距离下界,是该子序列在下一个时间窗口长度下所有包含该子序列的子序列对的领域偏好距离的下界。因此该下界虽然没有式(1)那样优秀的单调性,但

已经能够让本文利用此下界设计一个优良的剪枝方案。下面介绍本文的剪枝方案。

对于长度为 n 的时间序列 T,剪枝算法总体上分为以下 5 步。

1)遍历求得时间窗口长度为 $m-1$ 时,每一个子序列的最近邻距离,并求得最近邻距离下界。

2)计算每一个子序列在下一个时间窗口的惩罚距离,并且将其与最近邻欧氏距离下界组合成基于领域偏好的最近邻距离下界。取第 K 大的距离下界作为临界值。

3)依次计算时间窗口长度为 m 时,距离下界最小的前 K 个子序列的实际最近邻距离,并由此计算该子序列基于领域偏好的最近邻距离。若该距离小于临界值,则将其放入集合 S 中。

4)若集合 S 中元素的数量大于规定值,则说明剪枝成功。采用式(4)更新最近邻距离下界,时间窗口长度 m 加 1。转步骤 2)。

5)若 S 中元素的数量小于规定值,则剪枝失败。时间窗口长度 m 加 1。转步骤 1)。

下面具体解释这个算法。

算法的步骤 1)和步骤 2),就是根据式(4)来计算基于领域偏好的最近邻距离下界。这里的难点是理解步骤 1)的最后一句话,即为何第 K 大的最近邻领域偏好距离下界能够作为临界值。

假定时序数据 T 有 2 个时间窗口长度为 m 的子序列 $T_{i,m}$ 和 $T_{j,m}$,并且它们基于领域偏好的最近邻距离下界分别为 $D_{LB(i,m)}$,$D_{LB(j,m)}$。进一步假设:

$D_{LB(i,m)}<D_{LB(j,m)}<$其他子序列的领域偏好最近邻距离下界

如果取 $D_{LB(j,m)}$ 作为临界值,那么由 3.2.1 节中的证明可知,对于任意一个子序列对($T_{i,m}$,$T_{j,m}$),其关于 $T_{j,m}$ 的领域偏好距离一定大于 $D_{LB(j,m)}$。对子序列 $T_{i,m}$,$T_{j,m}$ 外的任意一个子序列对($T_{g,m}$,$T_{k,m}$),其领域偏好距离大于对应的领域偏好距离下界,也必然大于 $D_{LB(j,m)}$。因此,所有领域偏好距离小于 $D_{LB(j,m)}$ 的子序列对必然包含 $T_{i,m}$。

假定找到了一个子序列对($T_{i,m}$,$T_{g,m}$),其领域偏好距离小于 $D_{LB(j,m)}$,那么($T_{i,m}$,$T_{g,m}$)定然也小于被 $D_{LB(j,m)}$ 排除在外的元素。

同理,当选择第 K 大的最近邻领域偏好下界作为临界值时,对于所有更大的最近邻领域偏好下界的子序列,其构成的子序列的领域偏好距离定然大于这个临界值,因此被排除在外。而对于最近邻领域偏好下界小于临界值的子序列,由于不确定其最近邻领域偏好距离是否小于临界值,因此需要通过计算进一步判断。如果该子序列的最近邻领域偏好距离小于临界值,那么它就小于所有被排除在外的子序列。因此以最近邻领域偏好下界作为临界值非常好的判断条件。

算法步骤 3),即根据临界值筛选最优的基于领域偏好的最近邻距离的子序列:首先计算最近邻领域偏好距离下界最好的 K 个子序列的实际最近邻距离。然后将实际领域最近邻距离与临界值进行比较,若实际最近邻距离比临界值小,则说明该领域最近邻距离小于最近邻领域距离下界,从而也小于比临界值大的所有子序列对的领域最近邻距离,则可以将该子序列放入主题模式候选集合中。

这里需要额外说明的是,对于寻找前 K 个子序列的最近邻距离,由于实际中 K 远小于时间序列长度 n,因此采用经典的固定时间窗口长度求某子序列最近邻距离的算法,该算法的时间复杂度也仅仅是 $O(mn)$,m 为时间窗口长度。该算法的时间复杂度与采用子序列对的 MOEN 算法的时间复杂度是相同的。

然而,实际上算法的时间复杂度会远低于预期。因为对于某个子序列,在时间窗口长度变化不大的条件下,其最近邻距离对应的子序列的位置基本是不变的。因为对于同一个子序列、同一个时间窗口长度,其惩罚距离是固定的,所以寻找最近邻距离不需要考虑惩罚距离,则与 MOEN 算法相同,对于前 K 个子序列,每一个子序列都维护一个基于式(1)的距离下界,从而快速判断之前的最近邻距离的位置是否仍然是最近邻距离,以加速计算。

算法的步骤 4)和步骤 5)分别是剪枝成功和剪枝失败的做法。剪枝成功则更新距离下界,剪枝失败则在下一个时间窗口重新遍历计算。

关于步骤 4),下面给出更新距离下界的方法。定义 5 中对于基于领域偏好的最近邻距离的定义,将距离分成了最近邻欧氏距离和惩罚距离两个部分。而定义 6 沿用了这个分割方式,并将距离下界分成了最近邻欧氏距离下界和惩罚距离两个部分。

关于惩罚距离,从步骤 2)可以看出,每一个时间窗口都要计算,因此不需要考虑其更新方式。而最近邻欧氏距离下界是采用式(1)进行计算,假设在上一个时间窗口,根据式(1)计算的最近邻距离为 d_{LB},那么在当前时间窗口,不具体计算最近邻距离,而是直接用之前的 d_{LB} 替代式(1)中的实际距离 d,即:

$$d_{LB\_NEW}=\left(\frac{m-1}{m}+\frac{m-1}{m^2}z^2\right)^{-1}d_{LB}^2 \tag{7}$$

这个距离能作为时间窗口为 $m+1$ 时的距离下界,这是因为将 $d>d_{LB}$ 代入到式(1)中,可以得到:

$$\begin{aligned}d_{LB\_d}&=\left(\frac{m-1}{m}+\frac{m-1}{m^2}z^2\right)^{-1}d^2\\&>\left(\frac{m-1}{m}+\frac{m-1}{m^2}z^2\right)^{-1}d_{LB}^2=d_{LB\_NEW}\end{aligned} \tag{8}$$

这将距离下界进一步缩小了。因为本文的目标是求距离下界,所以一个距离更小的下界仍然可以作为剪枝的条件。因此采用式(7)的方法来更新距离下界是合理的。

3.3　主题模式实例聚合

本节介绍如何有效地发现同一主题模式的不同实例,同时寻找到不同的主题模式。本节设计了基于聚类的主题模式实例聚合算法。

首先讨论固定时间窗口长度下的主题模式聚合算法。这里假设已经求得每个子序列的最近邻距离,本文进行主题模式实例的聚合,分为以下几个步骤:

1)提前设定一个阈值 c 作为判断不可能是主题模式的临界值。将所有子序列的基于领域偏好的最近邻距离与阈值 c 比较,挑选出最近邻距离小于阈值 c 的子序列放入候选集合中,并将候选集合里面的子序列按照基于领域偏好的最近邻距离从小到大排序。同时,根据实际需要,设置 K 个集合。

K 为实际中需要的主题模式个数。

2)选择候选集合剩余元素里面的第一个元素,按照顺序依次计算它与 K 个集合的距离。如果该元素与某个集合 S 的最小距离小于 c 的某个比例(如 10%),那么说明它与集合 S 中的元素高度相似,则认为它是主题模式 S 下面的一个实例,从而将该元素加入集合 S,停止计算。如果所有有元素的集合均不符合此条件,则将该元素放入一个空集合中。将该元素从候选集中删掉。

3)重复步骤 2),直到所有 K 个集合都有元素,或者候选集中没有元素,算法停止。

下面来解释这个算法。

首先进行步骤 1),即设置一个阈值 c。该阈值用于判断子序列是否能够进入主题模式候选集合。因为子序列的基于领域偏好的最近邻距离代表了与子序列最相似的序列和它的相似性。如果这个相似度依然不高,那么有足够理由说明这个子序列不是主题模式。c 的设定可以通过经验判断,即作为超参数给出,也可以通过数据本身来获得。例如可以将 c 设置成所有子序列最小最近邻距离的 3 倍,或者设置成所有子序列的前 5%的最近邻距离等。

将候选集合的子序列按照从小到大的顺序排序,并且之后一直按照这个顺序搜索。虽然本文扩展了主题模式的定义,但是关于主题模式的优先级,本文仍然使用了定义 4 中的概念,即基于领域偏好的最近邻距离越小的子序列,其主题模式的优先级越高。而设计 K 个集合的本质就是设置 K 个主题模式。

算法步骤 2)是将子序列按照相似性进行聚类。因为之前已经按照基于领域偏好的最近邻距离从小到大排序,所以每次集合中的第一个元素就是没有被匹配主题模式的子序列中最应该优先被匹配的元素。对于这个子序列,首先依次计算这个子序列和已有的主题模式之间的距离。若子序列和某个主题模式距离很近,则该子序列就是这个主题模式的一个实例。若子序列和所有已经找到的主题模式之间的距离都很远,则说明该子序列是一个还未发现的主题模式的一个实例。并且无论是什么情况,都应该将该子序列从集合中删除。

计算子序列和主题模式之间的距离,最简单的方法是计算子序列和主题模式所在的集合内部所有元素距离的最小值。该算法的时间复杂度较高。因此可以考虑将每个集合中的基于领域偏好的最近邻距离最小的元素作为该集合的代表来计算距离。考虑到这些操作都在一个很小的候选集里,而且实际情况中 K 也很小,因此不必考虑时间复杂度的因素。

算法步骤 3)是循环步骤 2),并且在 K 个主题模式都被发现后停止算法。如果把所有候选集都检查完,找到的主题模式个数仍然小于 K,那么就返回找到的主题模式。这里需要说明的是,本文算法虽然找到了 K 个主题模式,但是并没有找到 K 个主题模式的所有实例。当然本文可以通过简单的修改,让算法找到所有实例:即让循环终止条件变为遍历所有候选集合里面的元素后停止循环,而不是找到第 K 个才停止。但这样做既浪费时间,也没有必要。因为正如前面所说,对主题模式实例进行聚合的核心目标是为了保证算法能够找到 K 个不同的主题模式,而不是为了找全所有主题模式的实例。算法发现第 K 个主题模式时就已经达到目标,此时没必要浪费额外的计算资源进行不必要的操作。而且这样也保证了整个主题模式聚合操作都在一个很小的数据量上,提高了程序效率。

可变时间窗口长度的主题模式识别算法的本质还是转换成固定时间窗口长度主题模式识别算法来求解。因此,对于主题模式实例的聚合,其本质上就是找到子序列后,多次调用固定时间窗口长度的主题模式实例聚合算法来解决。限于篇幅,这里不再赘述。

最后要强调,因为本节说明的内容是基于领域偏好的可变时间窗口长度主题模式识别的一个部分,因此主题模式实例聚合算法全部按照基于领域偏好的最近邻距离设计。但是本算法并不仅仅局限于基于领域偏好的最近邻距离,对于标准化欧氏距离,本算法同样适用,只需要将算法中基于领域偏好的最近邻距离替换为基于标准化欧氏距离的最近邻距离即可。

4 实验验证

本文用真实数据集对本文设计的基于领域偏好的可变时间序列主题模式识别算法进行实验验证,主要从以下 3 个方面对算法效果进行验证。

1)在给定领域偏好的基础上,算法能否成功找到具有领域偏好的主题模式。这是算法经过一系列变换将标准化欧氏距离变成基于领域偏好的距离的核心目的。在后面的实验中,将本算法找到的主题模式和 MOEN 算法找到的主题模式进行对比,以证明本算法确实能够成功找到有意义的具有领域偏好的主题模式。

2)相比于遍历搜索合适的主题模式,算法是否能够保证较高的效率。相比于 MOEN 算法,本算法加入了较多的功能,因此效率难免会有一定程度的下降。需要通过实验证实,相比于 MOEN 算法,本文算法的效率只有些许下降,相比于直接暴力搜索,其依然具有非常高的效率。

3)算法能否成功地对主题模式实例进行聚合。相比于 MOEN 算法,本算法对主题模式和主题模式实例进行了区分。为强调算法应该返回不同的主题模式,本文设计了主题模式实例聚合的算法来解决 MOEN 算法返回同一主题模式不同实例的问题。通过实验证实,主题实例聚合是非常必要的,并且本文算法能成功解决这个问题。

下面以两个真实数据集来验证算法的实际效果。

第一个数据集是家用电器功率数据集[16],该数据集记录了某个家用电器的工作情况。该数据集主要测试算法的效率和主题模式实例聚合效果。

第二个数据集是测试手指运动的数据集[17],该数据集记录手指弯曲情况。该数据集主要测试算法是否能够准确找到基于领域偏好的主题模式。

下面依次介绍算法在这两个数据集中的表现。本文主要对比了本文提出的算法 CTCL(Context Class model)、遍历搜索算法 BRUTE(Brute Force)[12] 和 MOEN 算法。

4.1 家用电器功率数据集

该数据集主要包括某一个家庭的常见用电器每分钟的用

电功率情况。这里使用该数据集是因为其数据量较大，方便对算法的时间效率进行测试。算法的时间效率测试总体上分为以下几步：

1）算法整体效果测试，即统计从算法开始，到按要求找到相应时间窗口长度范围内的主题模式为止，算法的整体运行时间。

2）算法单步遍历搜索用时，即假设算法剪枝失败，算法需要进行一次遍历搜索的时间消耗。

3）算法剪枝成功单步用时，即假设算法剪枝成功，算法在计算和更新距离下界的时间消耗。

4）算法剪枝失败的次数，反映算法剪枝的成功率。

为了防止偶然性的结论，随机从数据集里截取长度为 1 000，3 000，10 000，20 000，30 000 的 5 个长度的时间序列，最后将 CTCL 算法的时间效率与 MOEN 算法、BRUTE 算法的时间效率相对比。本文的实验环境为 CPU 为 Intel® Core™ i7-4790，内存为 8 GB 的电脑。

算法搜索的时间窗口长度范围为 30～100。其中对于每一个时间窗口长度，算法要求发现 5 个主题模式。由于 MOEN 算法只能返回主题模式实例，并且主题模式实例中有大量相似的主题模式。因此为了比较的公平性，在计算时间时，让 MOEN 算法返回前 50 个主题模式实例。由于本文算法加入了主题模式实例聚合，同一个主题模式下的不同实例只返回其中最近邻距离最小的实例，因此本文算法需要更多的主题模式实例候选集。按照 1 个主题模式有 10 个实例进行估计，MOEN 算法取前 50 个实例，基本保持了与本文算法相似的候选集水平，比较相对公平。实际上，在大多数情况下，1 个主题模式中的实例是超过 10 个的，即 MOEN 算法需要返回更多主题模式实例才与本算法有可比性。

图 4 为不同时间窗口长度下，3 个算法的整体用时对比。可以得到如下结论：

1）相比于传统 BRUTE 算法，CTCL 算法和 MOEN 算法在时间效率上均有很大提升。这一点随着数据规模的增大，效果提升越来越明显。

2）CTCL 算法的时间消耗大约为 MOEN 算法耗时的 1.6 倍。也就是说，CTCL 算法继承了 MOEN 算法高效的特点，其时间消耗仍然是很短的。

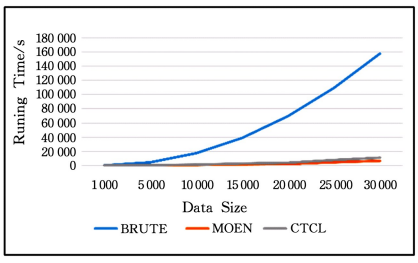

图 4　不同主题模式识别算法随数据规模变化的时间消耗

Fig. 4　Time consumption of three different motif discovery algorithms under different data sizes

CTCL 算法的时间消耗大约为 MOEN 算法的 1.6 倍，下面进一步研究时间主要消耗在什么环节。图 5 给出了 3 种算法进行一次固定时间窗口长度暴力搜索用时。对于传统 BRUTE 算法，遍历搜索是每一个时间窗口都需要进行的。对于 MOEN 算法和 CTCL 算法，遍历搜索对应算法剪枝失败后的操作。图 6 给出了 CTCL 算法和 MOEN 算法在剪枝成功时的时间消耗。

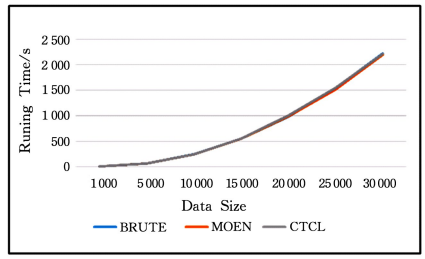

图 5　不同主题模式识别算法进行一次暴力搜索的时间消耗

Fig. 5　Time consumption of a brute force search by three different motif discovery algorithms under different data sizes

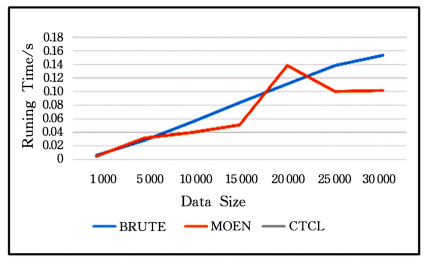

图 6　不同主题模式识别算法在一次剪枝成功时的搜索时间

Fig. 6　Search Time of a success pruning by two different motif discovery algorithms under different data sizes

从图 5、图 6 可以得到以下结论：

1）不同算法在进行一次暴力搜索的时间消耗基本相同，不同算法在进行一次剪枝成功后的搜索时间消耗也基本相同。

2）对于任何一个算法，在相同的数据规模下，算法进行一次暴力搜索的时间消耗要远远高于算法进行一次剪枝成功后的搜索的时间消耗。因此影响算法时间效率的最主要因素是暴力搜索时间和算法剪枝失败的次数。

图 7 给出了在不同数据规模情况下，MOEN 算法剪枝失败的次数和 CTCL 算法剪枝失败的次数。

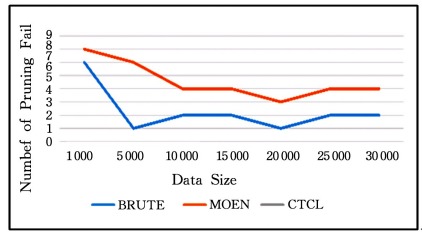

图 7　不同算法剪枝失败的次数

Fig. 7　Number of failed pruning by two different motif discovery algorithms under different data sizes

CTCL 算法剪枝失败的次数大约是 MOEN 算法的 1.6 倍，因此 CTCL 算法的整体耗时大约为 MOEN 算法的 1.6 倍。CTCL 算法之所以产生更多剪枝失败的情况，有以下 2 个原因：

1）CTCL 算法的距离下界是基于领域偏好的最近邻距离

下界。相比于 MOEN 算法基于子序列对的下界，CTCL 算法的下界在一定程度上被放宽了。因此其剪枝失败次数会略高于 MOEN 算法。

2)对于每个时间窗口，CTCL 算法最后返回的是 5 个不同的主题模式。相比之下，MOEN 算法虽然返回了 50 个实例，但在实验中发现，这些实例几乎来自同一个主题模式。因此，CTCL 算法维护的候选集合的数量远高于 MOEN 算法，这会导致更高的剪枝失败概率。

这里要强调的是，虽然 CTCL 算法剪枝失败的次数大约是 MOEN 算法的 1.6 倍，但是相比于传统的搜索算法，CTCL 算法依然保持了非常高的剪枝成功率。实验中，搜索的时间窗口长度范围是 30～100，也就是 71 个时间窗口。CTCL 算法在最坏情况下，也能保持 88.7%的剪枝成功率。而在平均情况下能保持 92.1%的成功率。对应地，MOEN 算法在最坏情况下能保持 90%的剪枝成功率，在平均情况下能保持 95.4%的剪枝成功率。因此 CTCL 算法依然能够保持算法的高效性。

4.2 手指运动数据集

上一个实验主要验证 CTCL 算法在时间效率、主题实例聚合方面的效果。虽然从一定程度上可以看出 CTCL 算法找到了对应领域偏好的主题模式，但是其实际意义不是特别显著。因此本文第二个实验选择手指运动数据集，集中观察 CTCL 算法在基于领域偏好的主题模式识别中的效果。该数据集主要记录手指的弯曲幅度。图 8 为该数据集的部分截图，其数据特点是：大部分数据波动较小，表示手指处于静止状态；但表示手指弯曲的数据一般具有较大波动。本文希望通过算法能够自发地找到手指弯曲幅度的主题模式，以方便进一步研究。因此该数据的领域知识为：希望能够找到波动较大的主题模式。下面通过这个实验来验证本文算法的优势。

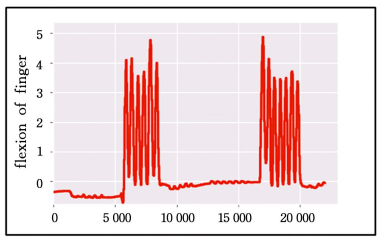

图 8 手指运动数据集的片段

Fig. 8 Part of finger motion dataset

由于领域偏好是希望找到波动性较大的主题模式，本文算法采用方差的方式来计算特征值。图 9 和图 10 分别是 CTCL 算法和 MOEN 算法找到的主题模式：

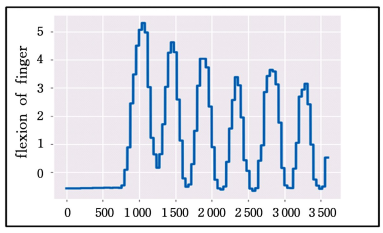

图 9 CTCL 算法找到的主题模式

Fig. 9 Motif found by CTCL algorithm

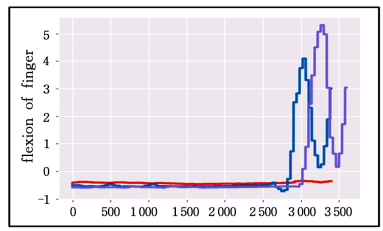

图 10 MOEN 算法找到的主题模式

Fig. 10 Motifs found by MOEN algorithm

可以看到，本节实验所用的手指运动数据集的数据有如下 2 个特点：1)数据集本身存在较多相对“稳定”的片段；2)基于标准化欧氏距离的度量存在简易化倾向。这两个原因的共同作用使得 MOEN 算法找到的主题模式大多为平缓片段，或者平缓片段加一个波动的结尾。这样的数据对于研究手指运动几乎是没有意义的。本文设计的 CTCL 算法利用基于领域偏好的最近邻距离成功解决了这个问题。在这个例子中，CTCL 算法在衡量距离时对复杂性进行了相应的修正，使得算法成功发现了记录手指弯曲的主题模式。在这个基于领域偏好的主题模式的基础上，研究者可以更好地进行手势识别等研究工作。因此本文设计的 CTCL 算法能够有效地将领域偏好加入到主题模式寻找中，可以发现基于领域偏好的可变时间窗口长度的主题模式。

结束语 本文算法首次成功地将简单领域知识加入到可变时间窗口长度下主题模式的寻找中，并且能够非常准确、高效地查找到带有简单领域知识的主题模式。根据需要，本文算法能够处理主题模式寻找中的“简易化”倾向。因此该算法非常方便领域专家使用。同时针对现有算法返回大量相同主题模式不同实例的问题，本文加入了主题模式聚合算法，成功地将子序列按照主题模式进行区分，保证了算法返回不同主题模式的实例。最后本文用实际数据对算法效果进行了检验，验证了算法在保持高效性的同时，能够准确地找到基于领域偏好的主题模式。

虽然本文设计的基于领域偏好的时序数据主题模式识别算法和时序数据主题模式规则发现算法已经在实验中取得了很好的效果，但是算法仍然有改进的空间。

基于领域偏好的时序数据主题模式识别算法目前依然采用基于剪枝的思路，因此该算法在时间复杂度上并没有进行任何优化。能否直接使用某种算法，在时间复杂度上对可变时间窗口主题模式识别问题进行优化，是未来非常重要的研究方向之一。另一方面，本文算法虽然可以得到基于领域偏好的各种时间窗口长度的主题模式，但是并没有讨论不同时间窗口长度的主题模式的取舍问题，在 MOEN 算法中采用了最大化主题模式来简化该问题，即当两个主题模式重合度较大时保留长度更长的主题模式，然而更合理的方式应该是综合考虑主题模式的时间窗口长度、主题模式实例在时间序列数据中出现的频率和实例之间的相似性以作出判断。因此，如何更好的筛选不同时间窗口长度的主题模式，也是未来非常重要的研究方向之一。

致谢 感谢英伟达公司为本研究提供 Titan X Pascal GPU 计算支持。

参 考 文 献

[1] BOX G E P,JENKINS G M,REINSEL G C,et al. Time series analysis:forecasting and control[M]. New Jersey:John Wiley & Sons,2015.

[2] PATEL P,KEOGH E,LIN J,et al. Mining motifs in massive time series databases[C]// Proceedings of 2002 IEEE International Conference on Data Mining. IEEE,2002:370-377.

[3] LONARDI J,PATEL P. Finding motifs in time series[C]// Proceedings of the 2nd Workshop on Temporal Data Mining. 2002:53-68.

[4] WANG H,ZHANG D,WANG Y,et al. RT-Fall:A real-time and contactless fall detection system with commodity WiFi devices[J]. IEEE Transactions on Mobile Computing,2017,16(2):511-526.

[5] BROWN A E X,YEMINI E I,GRUNDY L J,et al. A dictionary of behavioral motifs reveals clusters of genes affecting Caenorhabditis elegans locomotion[J]. Proceedings of the National Academy of Sciences,2013,110(2):791-796.

[6] LIN J,KEOGH E,FU A,et al. Approximations to magic:Finding unusual medical time series[C]// 18th IEEE Symposium on Computer-Based Medical Systems (CBMS'05). IEEE,2005:329-334.

[7] BARRENETXEA G,INGELREST F,SCHAEFER G,et al. Sensorscope:Out-of-the-box environmental monitoring[C]// Proceedings of the 7th International Conference on Information Processing in Sensor Networks. IEEE Computer Society,2008:332-343.

[8] MCGOVERN A,ROSENDAHL D H,BROWN R A,et al. Identifying predictive multi-dimensional time series motifs:an application to severe weather prediction[J]. Data Mining and Knowledge Discovery,2011,22(1-2):232-258.

[9] SHOKOOHI-YEKTA M,CHEN Y,CAMPANA B,et al. Discovery of meaningful rules in time series[C]// Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM,2015:1085-1094.

[10] KEOGH E,KASETTY S. On the need for time series data mining benchmarks:a survey and empirical demonstration[J]. Data Mining and Knowledge Discovery,2003,7(4):349-371.

[11] MUEEN A,KEOGH E,ZHU Q,et al. Exact discovery of time series motifs[C]// Proceedings of the 2009 SIAM International Conference on Data Mining. Society for Industrial and Applied Mathematics,2009:473-484.

[12] YEH C C M,ZHU Y,ULANOVA L,et al. Matrix profile I:all pairs similarity joins for time series:a unifying view that includes motifs,discords and shapelets[C]// 2016 IEEE 16th international conference on data mining (ICDM). IEEE,2016:1317-1322.

[13] ZHU Y,ZIMMERMAN Z,SENOBARI N S,et al. Matrix profile ii:Exploiting a novel algorithm and gpus to break the one hundred million barrier for time series motifs and joins[C]// 2016 IEEE 16th International Conference on Data Mining (ICDM). IEEE,2016:739-748.

[14] MUEEN A,CHAVOSHI N. Enumeration of time series motifs of all lengths[J]. Knowledge and Information Systems,2015,45(1):105-132.

[15] DAU H A,KEOGH E. Matrix Profile V:A Generic Technique to Incorporate Domain Knowledge into Motif Discovery[C]// Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM,2017:125-134.

[16] MAKONIN S,ELLERT B,BAJIĆ I V,et al. Electricity,water,and natural gas consumption of a residential house in Canada from 2012 to 2014[J]. Scientific Ata,2016,3:160037-160037.

[17] KUBÁNEK J,MILLER K J,OJEMANN J G,et al. Decoding Flexion Of Individual Fingers Using electrocorticographic signals in humans[J]. Journal of Neural Engineering,2009,6(6):066001-066001.