Journal of Computer Applications
计算机应用,2019,39(2):414-420
ISSN 1001-9081
CODEN JYIIDU
2019-02-10
http://www.joca.cn

文章编号:1001-9081(2019)02-0414-07
DOI:10.11772/j.issn.1001-9081.2018061326

基于子序列全连接和最大团的时间序列模体发现算法

朱跃龙,朱晓晓*,王继民
(河海大学 计算机与信息学院 南京 211100)
(*通信作者电子邮箱 1099084032@qq.com)

摘　要:针对时间序列模体发现算法计算复杂,并且无法发现多实例模体的问题,提出基于子序列全连接和最大团的时间序列模体发现(TSSJMC)算法。首先,使用快速时间序列子序列全连接算法求得所有子序列之间的距离,生成距离矩阵;然后,设置相似性阈值,将距离矩阵转化为邻接矩阵,构造子序列相似图;最后采用最大团搜索算法从相似图中搜索最大团,最大团的顶点对应的时间序列为包含最多实例的模体。在公开的时间序列数据集上进行实验,选用已有的能够发现多实例模体的 Brute Force 和 Random Projection 算法作为对比对象,分别从准确性、效率、可扩展性和鲁棒性对 TSSJMC 算法进行分析并获得了客观的评判结果。实验结果表明,与 Random Projection 算法相比,TSSJMC 算法在效率、可扩展性和鲁棒性法方面均有明显优势;与 Brute Force 算法相比,TSSJMC 算法发现的模体实例数量虽略低,但其效率和可扩展性都优于 Brute Force 算法。因此,TSSJMC 是质量和效率相平衡的算法。

关键词:时间序列;时间序列子序列;子序列连接;最大团;模体发现
中图分类号:TP311.13　　**文献标志码**:A

Time series motif discovery algorithm based on subsequence full join and maximum clique

ZHU Yuelong, ZHU Xiaoxiao*, WANG Jimin
(*College of Computer and Information, Hohai University, Nanjing Jiangsu* 211100)

Abstract: Existing time series motif discovery algorithms have high computational complexity and cannot find multi-instance motifs. To overcome these defects, a Time Series motif discovery algorithm based on Subsequence full Joins and Maximum Clique (TSSJMC) was proposed. Firstly, the fast time series subsequence full join algorithm was used to obtain the distance between all subsequences and generate the distance matrix. Then, the similarity threshold was set, the distance matrix was transformed into the adjacency matrix, and the sub-sequence similarity graph was constructed. Finally, the maximum clique in the similarity graph was extracted by the maximum clique search algorithm, and the time series corresponding to the vertices of the maximum clique were the motifs containing most instances. In the experiments on public time series datasets, TSSJMC algorithm was compared with Brute Force algorithm and Random Projection algorithm which also could find multi-instance motifs in accuracy, efficiency, scalability and robustness. The experimental results demonstrate that compared with Random Projection algorithm, TSSJMC algorithm has obvious advantages in terms of efficiency, scalability and robustness; compared with Bruce Force algorithm, TSSJMC algorithm finds slightly less motif instances, but its efficiency and scalability are better. Therefore, TSSJMC is an algorithm that balances quality and efficiency.

Key words: time series; time series subsequence; subsequence join; maximum clique; motif discovery

0　引言

时间序列是按时间顺序排列的、具有相等时间间隔的一系列数据的集合[1]。时间序列无处不在,使其在各个行业获得普遍的应用。例如金融领域的证券交易数据[2]、气象领域的气温气压数据[3]、工业领域的用电数据[4]、医学领域的脑电波和心电图数据[5-6]等。在时间序列数据挖掘的诸多问题中,时间序列模式发现是一个基础性问题。时间序列模式发现包括查找事先指定模式和预先未知的模式。查找事先指定模式的问题(即按内容查询)已有诸多解决方法[7-10]。然而,查找预先未知、重复出现的模式即时间序列模体发现(也称为时间序列的序列主题发现)问题则面临更多挑战[11]。模体发现对于时间序列挖掘具有重要意义,可以用于解决时间序列聚类、分类、关联规则发现等问题[12]。

模体发现源于生物信息学,用于寻找脱氧核糖核酸(DeoxyriboNucleic Acid, DNA)序列中具有相似排列和功能的短核苷酸片段。2002 年,Lin 等[11]首次将“模体”一词引入时间序列,并提出时间序列模体发现的概念,此后出现了许多模体发现方法。第一类常见的时间序列模体发现算法采用近似离散化方法,该类算法先采用字符串聚集近似(Symbolic Aggregate Approximation, SAX)算法将时间序列进行离散并符号化,然后在压缩后的数据中寻找相似片段或者提取规则[13-14]。此类算法虽表现出良好的性能,但是 SAX 会对数据进行平均处理,可能丢失数据集中具有重要意义的峰、谷等信息。第二类常见的模体发现方法是采用聚类发现时间序列模体,先将时间序列进行分段,然后使用聚类算法进行模体发现[15-16]。Eamonn 等[17]指出利用滑动窗口提取时间序列进行聚类挖掘是完全无意义的,因为通过滑动窗口平移提取的

收稿日期:2018-06-25;**修回日期**:2018-07-31;**录用日期**:2018-09-19。
作者简介:朱跃龙(1959—),男,江苏建湖人,教授,博士,主要研究方向:智能信息处理、数据挖掘;　朱晓晓(1994—),女,山东滕州人,硕士研究生,主要研究方向:数据挖掘;　王继民(1976—),男,安徽全椒人,副教授,硕士,主要研究方向:数据挖掘、智能信息处理。

每个数据点对于整体贡献为一条直线。第三类时间序列模体发现算法基于概率方法,该类算法主要采用滑动窗口提取子序列,然后将投影算法作用于子序列得到候选模体,再将候选模体作为基准从原始序列中查找多条模体[18-19]。此类算法虽效率高,但涉及参数过多并且计算复杂。第四类算法是子序列连接的时间序列模体发现算法,该类算法通过子序列连接得到所有子序列的最近邻后,构建子序列相似图,再采用最大团或最近邻算法计算相似序列作为模体[20-21]。该类算法在子序列连接和最大团搜索中计算成本很高。

2017 年 Yeh 等[1]使用快速傅里叶变换提高子序列全连接的速度,然后寻找每个子序列的 1-最近邻(1-Nearest Neighbor, 1-NN),彼此间相似度最高的两子序列即为模体,该算法无法发现多实例模体。结合文献[1]中算法计算简单、速度快的优势,本文提出基于子序列全连接和最大团的时间序列模体发现(Time Series motif discovery based on Subsequence full Joins and Maximum Cliques, TSSJMC)算法以高效地发现等长的多实例模体。实验结果表明,本文算法能够快速、准确地发现多实例模体,同时对噪声具有较好的鲁棒性。

1　相关工作

1.1　相关定义

定义 1　时间序列子序列[1]。给定一条时间序列 $\boldsymbol{T}=t_1,t_2,\cdots,t_n$,子序列 $\boldsymbol{S}_{(i,m)}=t_i,t_{(i+1)},\cdots,t_{(i+m-1)}$ 是 $\boldsymbol{T}$ 中一条从 i 开始,长度为 $m(m\leqslant n)$ 的连续序列,此处 $1\leqslant i\leqslant n-m+1$。

定义 2　子序列集[1]。时间序列 $\boldsymbol{T}$ 的一个子序列集 $\boldsymbol{A}$ 是长度为 m 的滑动窗口在时间序列 $\boldsymbol{T}$ 上顺序滑动得到的所有可能子序列的有序集合,可表示为 $\boldsymbol{A}=\{\boldsymbol{T}_{(1,m)},\boldsymbol{T}_{(2,m)},\cdots,\boldsymbol{T}_{(n-m+1,m)}\}$,$m$ 为用户定义的子序列长度。可用 $\boldsymbol{A}[i]$ 来表示 $\boldsymbol{T}_{(i,m)}$。

定义 3　距离矩阵。时间序列 $\boldsymbol{T}$ 的子序列集 $\boldsymbol{A}$ 中所有子序列之间距离组成的矩阵,可表示为 $\boldsymbol{M}$。

$$\boldsymbol{M}=\begin{bmatrix} m_{11} & m_{12} & \cdots & m_{1n} \\ m_{21} & m_{22} & \cdots & m_{2n} \\ \vdots & \vdots & & \vdots \\ m_{m1} & m_{m2} & \cdots & m_{mn} \end{bmatrix}$$

其中,m_{ij} 表示子序列 $\boldsymbol{A}[i]$ 和 $\boldsymbol{A}[j]$ 之间的距离。距离矩阵是对称矩阵。

定义 4　平凡匹配(Trivial Matching)。在一个时间序列里,存在两个片段 $\boldsymbol{C}$、$\boldsymbol{F}$,如果这两个片段起点相同,那么它们是平凡匹配;如果起点不同,并且在这两个序列之间不存在任何一个序列 $\boldsymbol{S}$ 满足 $D(\boldsymbol{C},\boldsymbol{S})>r$,那么它们是平凡匹配。$D$ 函数是距离函数,负责计算两对序列之间的距离,r 是事先定义的距离阈值。

定义 5　最大团。对于给定图 $G=(V,E)$。其中,$V=\{1,2,\cdots,n\}$ 是图 G 的顶点集,E 是图 G 的边集。图 G 的团是一个两两之间有边的顶点集合。团是 G 的一个完全子图。如果一个团不被其他任一团所包含,即它不是其他任一团的真子集,则称该团为图 G 的极大团(maximal clique)。顶点最多的极大团,称为图 G 的最大团(maximum clique)。如图 1 所示的无向图,{1,2,3} 为无向图的最大团。

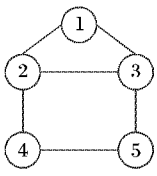
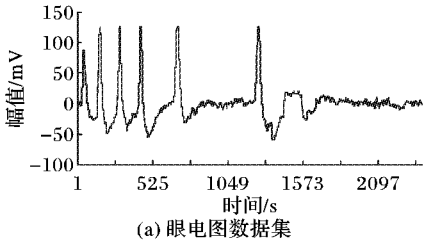


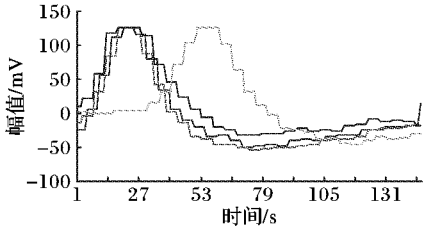
图 1　无向图

Fig. 1　Undirected graph

定义 6　时间序列模体。给定一条时间序列 $\boldsymbol{T}$,一个子序列 $\boldsymbol{C}$,如果与子序列 $\boldsymbol{C}$ 相似的子序列数量最多,那么 $\boldsymbol{C}$ 称为时间序列 $\boldsymbol{T}$ 的模体,所有与 $\boldsymbol{C}$ 相似的子序列称为模体的实例。具体示例如图 2 所示,图(a)为眼电图数据集,图(b)为其中的模体实例。



(a) 眼电图数据集



(b) 眼电图数据集中的模体实例

图 2　模体示例

Fig. 2　Motif example

1.2　Brute Force 算法

2002 年 Lin 等[11]提出 Brute Force(BF)算法,采用长度为 m 的滑动窗口在时间序列中提取子序列,依次将每个子序列作为查询序列,进行 r 范围查询,每个子序列及其相似子序列构成一个相似子集,包含子序列最多的相似子集中的子序列视为该时间序列的模体。该算法具有较高的时间复杂度,同时被广泛用作基准算法以评价其他算法的准确性[13,18,22-26]。BF 算法伪代码如算法 1 所示。

算法 1　BF 算法。

输入　时间序列 $\boldsymbol{T}$,子序列长度 n,相似性阈值 r;

输出　时间序列 $\boldsymbol{T}$ 的 1-motif 实例。

```
best_motif_count_so_far = 0                                    //初始化
best_motif_location_so_far = null
for i = 1 to length(T) - n + 1
  count = 0
  pointers = null
  for j = 1 to length(T) - n + 1
    if non_trival_match(C[i:i + n - 1], C[j:j + n - 1], R)
                                                    //去除平凡匹配
      count = count + 1
      pointers = append(pointers, j)
    endif
  endfor                                   /* 找到数量最多的子序列集合 */
  if count > best_motif_count_so_far
    best_motif_count_so_far = count
    best_motif_location_so_far = i
    motif_matches = pointers
  endif
endfor
```

1.3　Random Projection 算法

Chiu 等[18]提出了基于概率思想的 Random Projection(RP)算法,首先使用分段聚集近似(Piecewise Aggregate Approximation, PAA)算法对时间序列进行离散化,然后采用 SAX 算法对时间序列进行符号化,最后使用随机投影在符号序列中发现模体。随机投影使用碰撞矩阵作为基础结构,通

过滑动窗口提取原序列的子序列,并将其符号化后放入碰撞矩阵中。碰撞过程中,随机选择 *mask* 列作为掩码,依据选择列的值将矩阵投影到桶中,并通过增加碰撞矩阵中对应位置的值记录碰撞。循环地随机选择 *mask* 列作为掩码进行碰撞迭代,迭代适当的次数之后检查碰撞矩阵。碰撞矩阵中碰撞次数最多的子序列选为候选模体,并作为查询序列,寻找原序列中与其距离在相似性阈值之内的子序列,找到的子序列即为模体。

1.4 基于 Matrix Profile 的时间序列模体发现算法

2017 年,Yeh 等[1]提出基于 Matrix Profile 的时间序列模体发现算法,采用长度为 m 的滑动窗口在时间序列 $\boldsymbol{T}$ 上提取子序列,然后采用超快速相似性搜索算法(Mueen's ultra fast Algorithm for Similarity Search, MASS)[27]计算各子序列与其他子序列之间的距离,得到距离矩阵。将矩阵中每行的最小值作为 Matrix Profile 对应行中的元素,Matrix Profile 是记录时间序列中所有子序列与其最近邻子序列之间欧氏距离的向量。Matrix Profile 中提取的最小值对应的子序列为彼此间相似度最高的时间序列子序列,即为模体。

MASS 采用基于 z-归一化后的欧氏距离度量序列相似性,其核心是采用快速傅里叶逆变换替代计算复杂的卷积操作,从而提高算法的速度。该算法计算高效,无需复杂的参数设置,可扩展性强,但是仅能发现一对彼此最相似的时间序列子序列,无法发现多实例模体。MASS 实现的伪代码如算法 2 所示。

算法 2 MASS。

输入 查询序列 $\boldsymbol{Q}$, 时间序列 $\boldsymbol{T}$;

输出 查询序列 $\boldsymbol{Q}$ 的 distance profile。

1) $\boldsymbol{QT} \leftarrow$ SlidingDotProducts($\boldsymbol{Q},\boldsymbol{T}$)

2) $\mu_Q,\sigma_Q,M_T,\Sigma_T \leftarrow$ ComputeMeanStd($\boldsymbol{Q},\boldsymbol{T}$)

/* 计算 $\boldsymbol{Q}$ 和 $\boldsymbol{T}$ 的均值和方差,μ_Q 为 $\boldsymbol{Q}$ 的平均值,σ_Q 为 $\boldsymbol{Q}$ 的标准差,M_T 是时间序列 $\boldsymbol{T}$ 长度为 m 的子序列的平均值,Σ_T 是时间序列 $\boldsymbol{T}$ 的长度为 m 的子序列的标准差 */

3) $D \leftarrow$ CalculateDistanceProfile ($\boldsymbol{Q},\boldsymbol{T},\boldsymbol{QT},\mu_Q,\sigma_Q,M_T,\Sigma_T$)

//计算基于 z-归一化的欧氏距离

4) return D

算法 2 的第 1)行调用了 SlidingDotProducts 算法,其伪代码如算法 3 所示,主要功能是计算 $\boldsymbol{QT}[i]$ 的值。算法 3 的第 5)~6)行是对两个向量的经典卷积运算,算法将采用快速傅里叶变换和逆快速傅里叶变换替代计算复杂的卷积操作,从而提高 MASS 的速度。卷积操作提供了子序列和时间序列之间的滑动点积,换句话说,在自连接的情况下,卷积提供了所分析的子序列与来自其他时间序列或所属时间序列相同长度的子序列之间的点积。

算法 3 SlidingDotProducts 算法。

输入 查询序列 $\boldsymbol{Q}$, 时间序列 $\boldsymbol{T}$;

输出 查询序列 $\boldsymbol{Q}$ 和时间序列 $\boldsymbol{T}$ 所有子序列之间的点积。

1) $n \leftarrow$ Length ($\boldsymbol{T}$), $m \leftarrow$ Length ($\boldsymbol{Q}$)

//n 为时间序列 $\boldsymbol{T}$ 的长度, m 为查询序列 $\boldsymbol{Q}$ 的长度

2) $\boldsymbol{T}_\alpha \leftarrow$ Append $\boldsymbol{T}$ with zero

//后续点积计算要求两个向量具有相同的长度

3) $\boldsymbol{Q}_r \leftarrow$ Reverse($\boldsymbol{Q}$) //将 $\boldsymbol{Q}$ 倒序

4) $\boldsymbol{Q}_{r\alpha} \leftarrow$ Append $\boldsymbol{Q}_r$ with $2n-m$ zero

//为倒序后的 $\boldsymbol{Q}_r$ 填入 $2n-m$ 个零,

//后续点积计算要求两个向量具有相同的长度

5) $\boldsymbol{Q}_{raf} \leftarrow$ FFT($\boldsymbol{Q}_{r\alpha}$), $\boldsymbol{T}_{af} \leftarrow$ FFT($\boldsymbol{T}_\alpha$)

//傅里叶变换, $\boldsymbol{Q}_{raf}$ 和 $\boldsymbol{T}_{af}$ 表示两个时间序列频率分量的复数向量

6) $\boldsymbol{QT} \leftarrow$ InverseFFT(ElementwiseMultiplication($\boldsymbol{Q}_{raf},\boldsymbol{T}_{af}$))

//计算两个复矢量元素的乘积,并对乘积执行逆 FFT

7) return $\boldsymbol{QT}$

因为 MASS 采用 z-归一化的欧氏距离 $Dist[i]$ 作为时间序列子序列 $\boldsymbol{Q}$ 与 $\boldsymbol{T}_{i,m}$ 之间的距离度量,因此需要进行时间序列子序列 $\boldsymbol{Q}$ 与 $\boldsymbol{T}_{i,m}$ 之间的点积计算 $\boldsymbol{QT}[i]$。z-归一化的欧氏距离即在计算欧氏距离之前对参与计算的序列进行 z-归一化,公式为 $Dist[i] = \sqrt{\sum_{j=1}^{m}(\overline{Q}_j - \hat{T}_j[i])^2}$, 其中 $\overline{Q}_j = \frac{1}{\sigma_Q}(Q_j - \mu_Q)$, $\hat{T}_j[i] = \frac{1}{\Sigma_T}(T_j[i] - M_T)$。由正相关系数和 z-归一化的欧氏距离之间的关系[28]可得公式: $C(\boldsymbol{Q},\boldsymbol{T}[i]) = 1 - \frac{Dist^2[i]}{2m}$, 又由于相关系数公式[29]: $C(\boldsymbol{Q},\boldsymbol{T}[i]) = \frac{\boldsymbol{QT}[i] - m\mu_Q M_T[i]}{m\sigma_Q \Sigma_T[i]}$, 则 z-归一化的欧氏距离计算公式(详细推导过程参见文献[27])为:

$$Dist[i] = \sqrt{2m\left(1 - \frac{\boldsymbol{QT}[i] - m\mu_Q M_T[i]}{m\sigma_Q \Sigma_T[i]}\right)}$$

其中: m 为子序列的长度; μ_Q 为时间序列子序列 $\boldsymbol{Q}$ 的平均值, σ_Q 为 $\boldsymbol{Q}$ 的标准差; M_T 为 $\boldsymbol{T}_{i,m}$ 的平均值, Σ_T 为 $\boldsymbol{T}_{i,m}$ 的标准差。

通常情况下,计算长时间序列中每个子序列的平均值和标准差的时间复杂度为 $O(m)$。因此,算法使用了文献[30]中提出的方法:缓存时间序列值的累积和与累积平方和,在任何阶段两个累积和向量足以计算任意长度子序列的均值和方差。与 K 最近邻(K-Nearest Neighbors, KNN)相似性搜索方法不同,该算法计算的是查询序列与时间序列中所有子序列之间的距离,即时间序列 $\boldsymbol{T}$ 的距离分布(Distance Profile)。

2 TSSJMC 算法

2.1 算法思想

本文在 Lin 等[20]提出的基于子序列连接和最大团的模体发现框架基础上,结合基于 Matrix Profile 的时间序列模体发现算法,提出了计算简单并且能够发现多条模体的基于子序列全连接和最大团的时间序列模体发现算法——TSSJMC。首先使用 MASS[27]进行子序列全连接,得到子序列的距离矩阵,并剔除平凡匹配;然后设定相似性阈值,将距离矩阵转化为邻接矩阵,构造子序列相似图,图中顶点代表子序列,每条边代表满足距离小于设定阈值的相似关系;最后,寻找图中的最大团,最大团中的顶点对应的子序列即为一个模体。

2.2 算法实现

TSSJMC 算法流程如图 3 所示,包括子序列全连接、构建子序列相似图以及寻找最大团三个步骤,伪代码如算法 4 所示。

算法 4 TSSJMC 算法。

输入 时间序列 $\boldsymbol{T}$, 子序列长度 m, 相似性阈值 r;

输出 时间序列 $\boldsymbol{T}$ 的多条模体实例。

1) $n \leftarrow$ Length($\boldsymbol{T}$) //n 为时间序列 $\boldsymbol{T}$ 的长度

2) $idxes \leftarrow 1:n-m+1$ //初始化

3) $excZoneLen$ = round ($subLen * 0.5$);

//设置平凡匹配排除区域

4) for each idx in $idxes$

5) $D \leftarrow$ MASS($\boldsymbol{T}[idx],\boldsymbol{T}$) // 计算所有子序列之间的距离

6) D = abs(D);

7) $D1$ = sqrt(D);

```
8)   excZoneStart = max(1,idx - excZoneLen);
                                        //去除平凡匹配
9)   excZoneEnd = min(proLen, idx + excZoneLen);
                                        //去除平凡匹配
10)  D1(excZoneStart:excZoneEnd) = inf;
                                        //去除平凡匹配
11) endfor
    /*将 Distance Matrix 转化为邻接矩阵*/
12) n = size(D1);
13) for i = 1:n
14)   if D1(i) ≤ r
15)     D1(i) = 1;
16)   else
17)     D1(i) = 0;
18)   endif
19)   D(idx,:) = D1;
20) endfor
21) maxcliques = FindMaxCliques(D);          //找到最大团
22) derive the motif from maxClique;         //模体来源于最大团
```

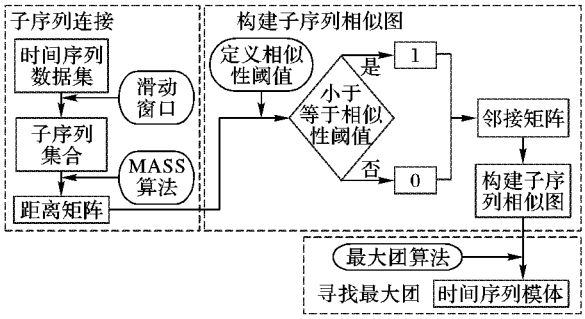


图 3　TSSJMC 算法流程

Fig. 3　Flow chat of TSSJMC algorithm

2.2.1　子序列全连接

使用长度为 m 的滑动窗口应用嵌套循环计算时间序列 $\boldsymbol{T}$ 中所有子序列之间的距离(即时间序列 $\boldsymbol{T}$ 的自连接),此过程使用“超快速”的 MASS 得到距离矩阵。该算法的“超快速”在于它先对数据进行快速傅里叶变换,然后执行点积操作,将其结果再进行逆傅里叶变换。该操作代替了计算复杂度较高的卷积操作。最后将此结果用于计算基于 z-归一化的欧氏距离,得到距离矩阵(Distance Matrix)。

2.2.2　构建子序列相似图

为了构建子序列相似图,需要将距离矩阵转化为对应的邻接矩阵。因此,定义相似性阈值,小于等于相似性阈值的距离用 1 表示,其他距离值用 0 表示。邻接矩阵的结构和转换过程如图 4 所示。假设图 4 的距离矩阵中存在平凡匹配的子序列的距离值设置为 inf,相似性阈值 $r=6$,距离值小于等于 r 时,距离元素用 1 表示,大于相似性阈值或为 inf 时用 0 表示,最终得到如图 4 右侧所示的相似邻接矩阵。

距离矩阵　　邻接矩阵

$$\begin{bmatrix} inf & 4.3 & 9.7 & 11.5 & 5.7 \\ 7.2 & inf & 2.3 & 5.8 & 1.3 \\ 9.4 & 2.2 & inf & 3.4 & 0.9 \\ 11.3 & 5.7 & 3.6 & inf & 0.7 \\ 5.5 & 1.2 & 0.5 & 0.8 & inf \end{bmatrix} \xrightarrow{r=6} \begin{bmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \end{bmatrix}$$

图 4　距离矩阵转换为邻接矩阵

Fig. 4　Transformation of distance matrix to adjacency matrix

2.2.3　寻找最大团

寻找图的最大团是一个复杂的组合优化问题,本文使用了文献[31]中求解最大团的算法。该算法提出了一个新的目标函数 $R1N_dM$:$\min\limits_{u\in\mathbf{R}^n} F(\boldsymbol{u}) = \|\boldsymbol{M}_d - \boldsymbol{u}\boldsymbol{u}^{\mathrm{T}}\|_F^2 (d \geqslant 0)$。假设图 G 的邻接矩阵 $\boldsymbol{A} \in \{0,1\}^{n\times n}$,定义相关的改进邻接矩阵 $\boldsymbol{B} = \boldsymbol{A} + \boldsymbol{I}_n$,$\boldsymbol{M}_d = (1+d)\boldsymbol{B} - d\boldsymbol{1}_{n\times n}$,其中 $\boldsymbol{1}_{n\times n}$ 为元素全 1 的 $n\times n$ 矩阵。目标函数的局部最小值对应图 G 的极大团,全局最小值对应图 G 的最大团。使用梯度下降算法作为迭代算法,利用 Armijo 准则调节步长,求解目标函数的最优解,该最优解对应图 G 的最大团(具体证明及步骤请参考文献[31])。该算法准确率较高并且计算速度非常快。

3　实验结果与分析

3.1　算法思想

实验采用文献[32]公开的数据集 EEG、EOG、ECG、Insect Behavior,数据集信息如表 1 所示。

表 1　数据集信息

Tab. 1　Information of datasets

序号	数据集	长度	备注
1	EEG	4 000	脑电图数据集
2	EOG	4 000	眼电图数据集
3	ECG	3 001	心电图数据集
4	Insect Behavior	4 000	昆虫行为数据集

3.2　实验方法

实验采用 Intel Core i5 CPU,8 GB 内存,操作系统为 Windows 7,软件为 Matlab R2012b。

实验将从算法的准确性、高效性、可扩展性和鲁棒性四个方面,将本文算法与 BF 算法[11]和 RP 算法[18]进行对比。将 BF 算法作为基准算法,将 TSSJMC 算法与 BF 算法在四个数据集上所发现的模体及其数量进行对比,以验证本文算法的准确性。为了评估 TSSJMC 算法的效率,将它与 BF 和 RP 算法在四个数据集上发现模体的运行时间(以 s 为单位)进行统计对比。

针对可扩展性,与 BF 和 RP 算法的对比实验分为两部分:第一部分基于四个数据集,分别在模体长度取值为 64、128、256、512 和 1 024 条件下,分析对比三个算法发现模体所需的运行时间;第二部分则基于 EEG、EOG 数据集,使用固定的模体长度,以 1 000 的步长增加数据集的长度,对比三个算法发现模体所需的运行时间。

针对鲁棒性,在 Insect Behavior 数据集上使用固定的模体长度,利用 Matlab 中的 awgn 函数对数据集进行加性高斯噪声处理:x = awgn($data$, SNR)。其中,SNR 为信噪比,以 dB 为单位,SNR 值越大,表示含有的噪声越少。在 0 ~ 80 dB 的信噪比范围内,对比分析各算法能够发现的模体数量和所需运行时间。

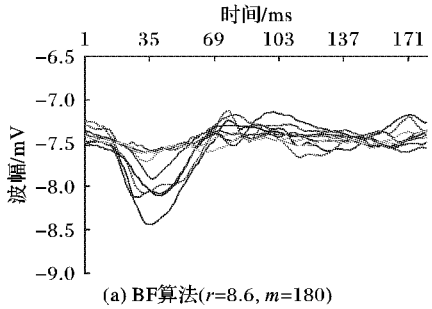
3.2.1　准确性分析

本实验中 TSSJMC 算法所使用的最大团算法的参数 γ、σ、β 都将使用文献[31]建议的最优值。BF 算法和 TSSJMC 算法找到的模体实例数量如表 2 所示,找到的具体模体实例如图 5 ~ 8 所示。

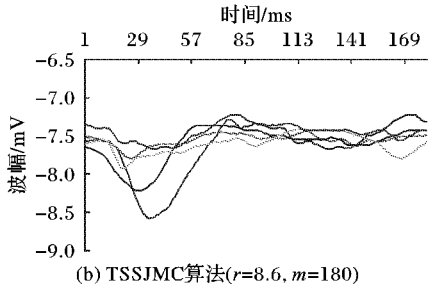
表 2　两种算法找到的模体实例数量对比

Tab. 2　Found motif instances comparison of two algorithms

数据集	BF	TSSJMC	数据集	BF	TSSJMC
EEG	7	5	ECG	14	14
EOG	5	5	Insect Behavior	10	6



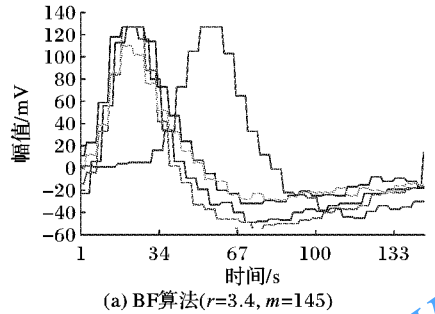
(a) BF算法(r=8.6, m=180)



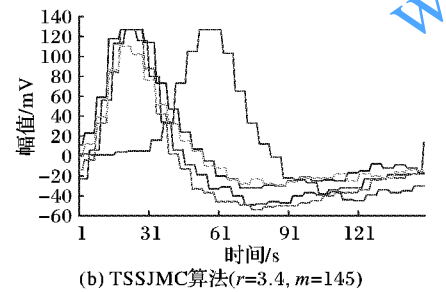
(b) TSSJMC算法(r=8.6, m=180)

图 5 BF 和 TSSJMC 算法在 EEG 数据集上发现的模体

Fig. 5 Found motifs by BF and TSSJMC from dataset EEG



(a) BF算法(r=3.4, m=145)



(b) TSSJMC算法(r=3.4, m=145)

图 6 BF 和 TSSJMC 算法在 EOG 数据集上发现的模体

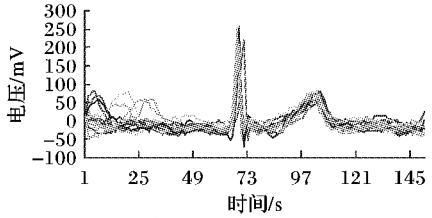
Fig. 6 Found motifs by BF and TSSJMC from dataset EOG

对比表 2 的实验结果可以看出，对于 EOG 和 ECG 数据集，TSSJMC 算法发现的模体个数与 BF 算法相同；而对于 EEG 和 Insect Behavior 数据集，TSSJMC 算法发现的模体个数略少于 BF 算法，这是由最大团算法导致的。因为采用的最大团算法虽然效率很高，但是它发现的团的大小有时略小。观察图 5～8 可以确定 TSSJMC 算法所发现的模体实例数与 BF 算法相同，说明 TSSJMC 算法具有较高的准确性，并且能够有效地发现多条模体。

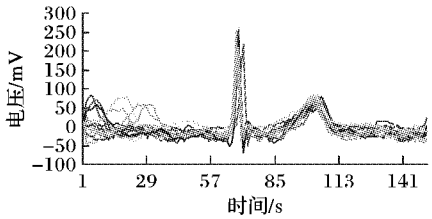
3.2.2 效率分析

本组实验中三个对比算法的相似性阈值 r 和模体长度 m 的设置与准确性分析实验场景相同。RP 算法涉及的参数有 PAA 分段数 *PAA_size*，SAX 的字符数目 α 和迭代次数 *repeat*。各个数据集 RP 算法参数设置如表 3 所示。

图 9 给出了三种算法发现模体所需运行时间的实验结果。从图 9 可以看出，TSSJMC 算法所需的运行时间远低于 BF 算法，同样略低于 RP 算法，验证了 TSSJMC 算法具有高效性；同时也表明 BF 算法效率不高，因为它是暴力搜索算法。结合算法准确性对比实验的结论可以说明，虽然 TSSJMC 算法发现模体的数量略少于 BF 算法，但其效率远高于后者，因此，相对来说 TSSJMC 算法是质量和效率相平衡的算法。



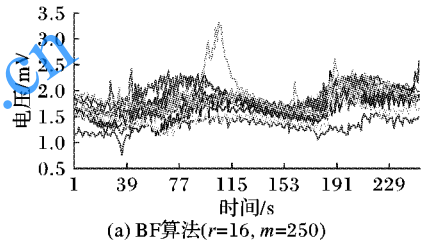
(a) BF算法(r=16.5, m=150)



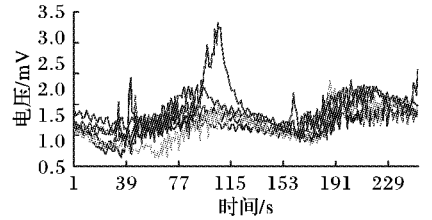
(b) TSSJMC算法(r=16.5, m=150)

图 7 BF 和 TSSJMC 算法在 ECG 数据集上发现的模体

Fig. 7 Found motifs by BF and TSSJMC from dataset ECG



(a) BF算法(r=16, m=250)



(b) TSSJMC算法(r=16, m=250)

图 8 BF 和 TSSJMC 算法在 Insect Behavior 数据集上发现的模体

Fig. 8 Found motifs by BF and TSSJMC from dataset Insect Behavior

表 3 各数据集 RP 算法参数设置

Tab. 3 Parameters setting of RP algorithm for comparison datasets

数据集	*PAA_size*	α	*repeat*	数据集	*PAA_size*	α	*repeat*
EEG	30	6	60	ECG	20	3	60
EOG	25	5	60	Insect Behavior	30	5	60

3.2.3 可扩展性分析

该实验由两部分组成：第一部分基于上述四个数据集，分别取模体长为 64、128、256、512 和 1024，三个算法运行时间结果如图 10 所示；第二部分则选用 EEG 和 EOG 数据集，固定模体长度，不断以 1000 的步长递增数据集的长度，三个算法的运行时间结果如图 11 所示。

分析图 10 可知：随着模体长度的增加，BF 算法的运行时间呈现指数型增加，而 TSSJMC 和 RP 算法的运行时间都远低于 BF 算法。在运行时间增长速率上，RP 算法虽比 BF 算法低，但随着模体长度的增加，其增长速率逐渐快于 TSSJMC 算

法,而 TSSJMC 算法的运行时间能够始终保持在两算法最好情况下的水平。

由图 11 可得,随着数据集长度的增加,三个算法的运行时间都呈递增趋势,BF算法增长最快,RP和TSSJMC算法增长相对缓慢。对比运行时间,TSSJMC 算法则要优于 RP 算法。因此,总体效果上,TSSJMC 算法具有更强的可扩展性。

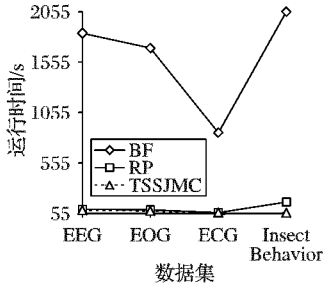


图 9　三种算法的执行时间对比

Fig. 9　Execution time comparison of three algorithms

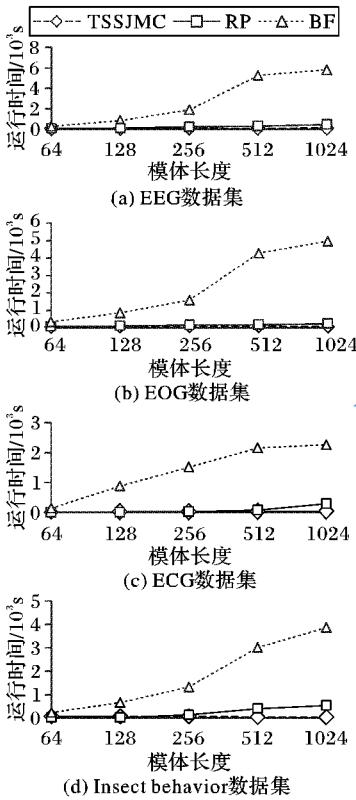


图 10　各个数据集中模体长度不断增加时三种算法的运行时间对比

Fig. 10　Running time comparison of three algorithms in each dataset with increasing motif length

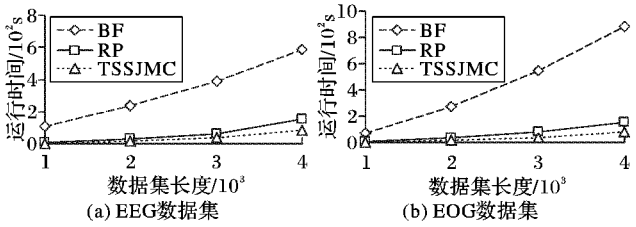


图 11　三种算法处理不同长度的各个数据集的运行时间对比

Fig. 11　Running time comparison of three algorithms on each dataset with different lengths

3.2.4　鲁棒性分析

基于 Insect Behavior 数据集,固定模体长度为 250,使用 Matlab 中 awgn 函数对数据集加入高斯噪声,控制信噪比以 10 为步长逐渐递增,设置 TSSJMC 算法的 $r = 16$,RP 算法的 $PAA\_size = 30$,$\alpha = 5$,$repeat = 60$。鲁棒性对比实验结果如图 12 所示。

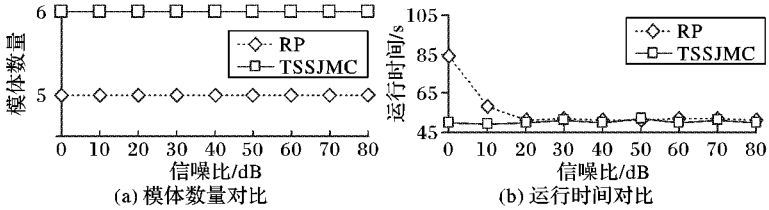


图 12　不同噪声水平下不同算法发现的模体数量和算法运行时间对比

Fig. 12　Comparison of number of motifs and running time of different algorithms at different noise level

图 12(a)中的实验结果表明:加入噪声对于 TSSJMC 和 RP 算法发现模体的数量没有影响,说明这两个算法都具有较强的鲁棒性。但图 12(b)中的对比结果表明:加入不同程度的噪声后,TSSJMC 算法性能表现稳定,始终保持在较好的水平;而 RP 算法性能快速恶化,受噪声的影响较为严重。因此,TSSJMC 算法相比 RP 算法有更强的鲁棒性,因为 RP 算法使用 SAX 算法进行符号化数据,噪声的加入使得 SAX 算法性能受到影响。

4　结语

针对时间序列模体发现算法计算复杂,并且无法发现多实例模体的问题,本文提出了一种计算简单并且能够发现多实例模体的基于子序列全连接和最大团的时间序列模体发现(TSSJMC)算法。该算法通过子序列全连接,构建子序列相似图,寻找图中的最大团三个步骤获得时间序列中的多实例模体。基于多个公开数据集的多组实验表明,本文提出的算法能够在较短时间内发现多条模体,并且具有高效性、准确性和更强的可扩展性和鲁棒性。文中算法发现的模体均为等长模体,未来我们将考虑发现不同长度的模体。

参考文献:

[1]　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]// Proceedings of the 2016 IEEE 16th International Conference on Data Mining. Piscataway, NJ: IEEE, 2016: 1317 - 1322.

[2]　周博,严洪森. 基于小波和多维泰勒网动力学模型的金融时间序列预测[J]. 系统工程理论与实践, 2013, 33(10): 2654 - 2662. (ZHOU B, YAN H S. Financial time series forecasting based on wavelet and multi-dimensional Taylor network dynamics model [J]. Systems Engineering — Theory and Practice, 2013, 33(10): 2654 - 2662.)

[3]　AILLIOT P, BESSAC J, MONBET V, et al. Non-homogeneous hidden Markov-switching models for wind time series [J]. Journal of Statistical Planning and Inference, 2015, 160: 75 - 88.

[4]　张淑清,师荣艳,李盼,等. 基于混沌关联积分的暂态电能质量扰动分类[J]. 仪器仪表学报, 2015, 36(1): 160 - 166. (ZHANG S C, SHI R Y, LI P, et al. Transient power quality disturbance classification based on chaos-correlation-integral [J]. Chinese Journal of Scientific Instrument, 2015, 36(1): 160 - 166.)

[5]　ARES J, LARA J A, LIZCANO D, et al. A soft computing frame-

work for classifying time series based on fuzzy sets of events [J]. Information Sciences, 2016, 330: 125 – 144.

[6] PADMAVATHI S, RAMANUJAM E. Naive bayes classifier for ECG abnormalities using multivariate maximal time series motif [J]. Procedia Computer Science, 2015, 47: 222 – 228.

[7] GE X, SMYTH P. Deformable Markov model templates for time-series pattern matching [C]// Proceedings of the 2000 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM, 2000: 81 – 90.

[8] KALPAKIS K, GADA D, PUTTAGUNTA V. Distance measures for effective clustering of ARIMA time-series [C]// Proceedings of the 2001 IEEE International Conference on Data Mining. Washington, DC: IEEE Computer Society, 2001: 273 – 280.

[9] KEOGH E, CHAKRABARTI K, PAZZANI M, et al. Dimensionality reduction for fast similarity search in large time series databases [J]. Knowledge & Information Systems, 2001, 3(3): 263 – 286.

[10] CHAKRABARTI K, KEOGH E, MEHROTRA S, et al . Locally adaptive dimensionality reduction for indexing large time series databases [J]. ACM Transactions on Database Systems, 2002, 27 (2): 188 – 228.

[11] LIN J, KEOGH E, LONARDI S, et al. Finding motifs in time series [C]// Proceedings of the 2nd Workshop on Temporal Data Mining. New York: ACM, 2002: 53 – 68.

[12] 马百鸣. 基于 DTW 度量的时间序列主旨模式提取[D]. 大连: 大连理工大学, 2011: 1 – 3. (MA B M. Motif extraction algorithm of time series based on DTW [D]. Dalian: Dalian University of Technology, 2011: 1 – 3.)

[13] PATEL P, KEOGH E, LIN J, et al. Mining motifs in massive time series databases [C]// Proceedings of the 2002 IEEE International Conference on Data Mining. Washington, DC: IEEE Computer Society, 2002: 370 – 377.

[14] LIN J, LI Y. Finding approximate frequent patterns in streaming medical data [C]// Proceedings of the 2010 IEEE 23rd International Symposium on Computer-Based Medical Systems (CBMS). Washington, DC: IEEE Computer Society, 2010: 13 – 18.

[15] MURAKAMI K, DOKI S, OKUMA S, et al. A study of extraction method of motion patterns observed frequently from time-series posture data [C]// Proceeding of the 2005 IEEE International Conference on Systems, Man and Cybernetics. Piscataway, NJ: IEEE, 2005: 3610 – 3615.

[16] FU T-C, CHUNG F-L, LUK R, et al. Preventing meaningless stock time series pattern discovery by changing perceptually important point detection [C]// Proceedings of the 2005 Second International Conference on Fuzzy Systems and Knowledge Discovery, LNCS 3613. Berlin: Springer, 2005: 1171 – 1174.

[17] EAMONN KEOGH J L. Clustering of time series subsequences is meaningless: implications for past and future research [J]. Knowledge & Information Systems, 2003, 8(2): 154 – 177.

[18] CHIU B, KEOGH E, et al. Probabilistic discovery of time series motifs [C]// Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM, 2003: 493 – 498.

[19] MINNEN D, ISBELL C, ESSA I, et al. Detecting subdimensional motifs: an efficient algorithm for generalized multivariate pattern discovery [C]// Proceedings of the 2007 Seventh IEEE International Conference on Data Mining. Washington, DC: IEEE Computer Society, 2007: 601 – 606

[20] LIN Y, MCCOOL M D, GHORBANI A A. Time series motif discovery and anomaly detection based on subseries join [J]. IAENG International Journal of Computer Science, 2010, 37(3): 259 – 271.

[21] SILVA D F, YEH C-C M, ENRIQUE G, et al. SiMPle: assessing music similarity using subsequences joins [C]// Proceedings of the 17th International Society for Music Information Retrieval Conference, New York: ISMIR, 2016: 23 – 29.

[22] GRABOCKA J, SCHILLING N, SCHMIDTTHIEME L. Latent time-series motifs [J]. ACM Transactions on Knowledge Discovery from Data, 2016, 11(1): 1 – 20.

[23] YANKOV D, KEOGH E, MEDINA J, et al. Detecting timeseries motifs under uniform scaling [C]// Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM, 2007: 844 – 853.

[24] LI Y, LIN J. Approximate variable-length time series motif discovery using grammar inference [C]// Proceedings of the 10th International Workshop on Multimedia Data Mining. New York: ACM, 2010: Article No. 10.

[25] LI Y, LIN J, OATES T. Visualizing variable-length time series motifs [C]// Proceedings of the 12th SIAM International Conference on Data Mining. Philadelphia, PA: Society for Industrial and Applied Mathematics (SIAM), 2012: 895 – 906.

[26] DUY T C, ANH D T. A fast method for motif discovery in large time series database under dynamic time warping [C]// Proceeding of the Sixth International Conference on Knowledge and Systems Engineering, AISC 326. Cham: Springer, 2015: 155 – 167.

[27] MUEEN A, ZHU Y, YEH M, et al. The fastest similarity search algorithm for time series subsequences under euclidean distance [EB/OL]. (2015-08-01) [2017-11-01]. http://www.cs.unm.edu/~mueen/FastestSimilaritySearch.html.

[28] MUEEN A, NATH S, LIU J. Fast approximate correlation for massive time-series data [C]// Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data. New York: ACM, 2010: 171 – 182.

[29] SAKURAI Y, PAPADIMITRIOU S, FALOUTSOS C. BRAID: stream mining through group lag correlations [C]// Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data. New York: ACM, 2005: 599 – 610.

[30] RAKTHANMANON T, CAMPANA B, MUEEN A, et al. Searching and mining trillions of time series subsequences under dynamic time warping [C]// Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM, 2012: 262 – 270.

[31] BELACHEW M T, GILLIS N. Solving the maximum clique problem with symmetric rank-one non-negative matrix approximation [J]. Journal of Optimization Theory and Applications, 2017, 173 (1): 279 – 296.

[32] MUEEN A, KEOGH E. Online discovery and maintenance of time series motif [EB/OL]. (2010-06-25) [2017-11-01]. http://alumni.cs.ucr.edu/~mueen/OnlineMotif/.

ZHU Yuelong, born in 1959, Ph. D., professor. His research interests include intelligent information processing, data mining.

ZHU Xiaoxiao, born in 1994, M. S. candidate. Her research interests include data mining.

WANG Jimin, born in 1976, M. S., associate professor. His research interests include data mining, intelligent information processing.