学号：1806010327

年级：2018 级

河海大學

本科毕业论文

深圳河道降雨溢流污染模式挖掘方法研究与应用

专　　业 计算机科学与技术

姓　　名 陈易轩

指导教师 陆佳民

评 阅 人

2022 年 4 月

中国　南京

BACHELOR'S DEGREE THESIS OF HOHAI UNIVERSITY

Research and Application of Rainfall Overflow Pollution Pattern Discovery Method in Shenzhen Rivers

College ： College of Computer and Information

Subject ： Computer Science and Technology

Name ： Yixuan Chen

Directed by ： Jiamin Lu Professor

NANJING CHINA

郑 重 声 明

本人呈交的毕业论文，是在导师的指导下，独立进行研究工作所取得的成果，所有数据、图片资料真实可靠。尽我所知，除文中已经注明引用的内容外，本设计（论文）的研究成果不包含他人享有著作权的内容。对本设计（论文）所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确的方式标明。本设计（论文）的知识产权归属于培养单位。

本人签名：_____　　日期：_____

摘要

随着信息技术的发展，尤其是大数据时代的到来，我们生活中的各行各业每时每刻都在产生大量的数据，如何从海量数据中发现有意义的关系成为了各个领域所面临的机遇与挑战。

时间序列数据是按照一定时间间隔和顺序进行记录和保存的观测值，广泛存在于自然、医学、社会、工业等各个领域中。从时间序列数据集中挖掘其中频繁出现的子序列，即频繁模式，此类方法称为模式挖掘。

在水利领域中，城市里河道的污染情况影响着城市形象、居民生活以及生态自然环境，挖掘水利相关的数据集之中蕴含的污染模式，对防治污染风险等具有重要价值。

本文试图从结合关联规则挖掘、因果关系这两个方法的角度出发，以深圳河道为研究对象，讨论其在水利领域下降雨溢流污染模式中的应用，通过能够对时空范围内不同的要素数据进行挖掘，从而分析河道污染事件的演化过程，形成污染成因链，作为决策的支撑。

关键字：多元时间序列；降雨溢流污染；模体挖掘；关联规则挖掘；决策支持

ABSTRACT

With the development of information technology, especially the arrival of the era of big data, all walks of life are generating a large amount of data at all times. How to find meaningful relationships from massive data has become an opportunity and challenge faced by all fields.

Time series data are observations that are recorded and preserved at a certain time interval and sequence, which are widely used in various fields such as nature, medicine, society and industry. Frequent patterns are mined from time series data sets. This method is called pattern mining.

In the field of water conservancy, the pollution of urban rivers affects the image of the city, the life of residents and the ecological and natural environment. It is of great value to explore the pollution patterns contained in the data sets related to water conservancy to prevent and control pollution risks.

This article attempts from the combined association rules mining, the perspective of causal relationship between the two methods, in the Shenzhen river as the research object, and discuss its application in water conservancy field under rainfall overflow pollution mode, through to the space and time within the scope of the elements of different data mining, and analyses the evolution of the river pollution incident process, form the pollution cause chain, as decision support.

Keywords: Multivariate time series; Rainfall overflow pollution; Motif discovery; Association rule mining; Decision support

目录

图目录

表目录

第一章 绪 论

1.1 技术背景和意义

时间序列是在浩如烟海的统计数据中，按照一定时间间隔记录并保存的观测值序列，其在我们的生活中其实也随处可见。比如，工业生产过程中各种传感器产生的数据[1][2]、医学领域中医疗器械产生的心电图和脑电图等数据[3][4]、金融领域中股票交易数据[5][6]、气象领域中气温气压等监测数据[7][8]、水文领域的水位降雨量监测数据[9][10]等都是社会生活中典型的时间序列数据。如今，如何用计算机技术从各领域记录的多元时间序列数据中，将其背后晦涩的数学规律转化为高度可读、可用的信息，为行业发展和改进提供决策支持，也是当下各行业中所迎来的发展机会[11]。

可见，周期性地挖掘多元时间序列关联规则探究事件的成因和规律性在当今行业内是十分重要的。一方面，挖掘蕴藏其中的自然演变规律和人类活动影响的信息，为决策制定提供支持。另一方面，通过提取数据中潜在的模式，对理解不同事件在邻近时间域内的相互作用机制具有重要的意义。

然而鉴于多元时间序列高维、海量等特点，难以直接将传统方法应用于原始时间序列数据，其中面临的挑战关键在于，在引入支持度的阈值，对单项指标提取多个模体后，其模体内的相似子序列存在重复的情况，对于这些重复的冗余的模体，如果不进行后续处理，将在符号化过程中互相复写，造成结果的混乱和算力的大量浪费。而特别在实际应用场景中，这些模体作为输入而挖掘出的规则结果会难以反映序列间的滞后关系。

这些数据往往包含多个监测参数，从而得到多条具有时间特性的序列数据，即多元时间序列，其记录的是同一现象随时间演化的不同状态，反映了不同属性间的相互影响。

例如，在暴雨影响下，深圳河湾易发生因雨污混合超出截流能力而导致的溢流污染事件，从而严重影响深圳河口的水质指标。而深圳河道周边的水质检测站收集了一定时间内的水质测量数据，其中降雨量与多种水质化学成分便是

多元时间序列类型的数据。它们的变化潜在描述了不同类型的水质变化事件，在不同水质因素影响下，各类水质化合物可能相互作用促进升高或降低，将会导致特定污染事件的发生。可以通过有效分析在不同降雨情势下各类溢流污染成因对河道水质产生的影响，来为河道水质调度提供决策支撑；同时通过分析评估水质监测站、雨量站等提供的不同类型的监测信息，进行模式挖掘，为下一阶段的可持续系统治理指明方向。

综上所述，本文研究具有时序关系的多元时间序列关联规则挖掘，引入支持度和相似度双阈值的同时，对模体挖掘的结果进行冗余剔除，同时以深圳市的水质检测数据为实际应用背景，具有一定研究意义和实际应用价值。

1.2 论文研究内容

通过挖掘时间序列数据，可以从中发现存在互相影响关系的指标，并可以进一步探究它们中反复出现的规律，此即为时间序列数据的模式挖掘。如今，时间序列数据的模式挖掘主要由关联规则挖掘实现，而关联规则挖掘的主要前置过程便是模体挖掘，即从时间序列中挖掘反复出现的子序列。

通过挖掘数据集中的**模体**并符号化是针对时间序列数据进行关联规则挖掘的常用方法[10]，尽管模体挖掘技术已经发展了一段时间，但针对水利领域的应用成果还有发展空间，本文将关联规则挖掘的方法应用到深圳市河道水质/雨量数据集之中进行研究与分析，从多元时间序列数据集中挖掘出潜在的周期性规律，为深圳市防治降雨溢流和水质污染提供决策支持。本研究旨在测试水利领域下模体挖掘与关联规则挖掘的实用性，并在原算法和数据处理流程的基础之上尝试改进，提出更具效率的模式挖掘方法并在水利领域下进行实践。

1.2.1 时间序列模体挖掘研究现状

模体挖掘最早是在 2012 年由加利福尼亚大学河滨分校的 Lin 等人[10]提出的，文中将相似性搜索中发现的重复模式命名为“模体”，因为其与生物学里的模体(motif)很相近[10]，代表着一组序列中出现的相似的片段和模式，例如蛋白质序列、DNA 序列等信息，对信息技术中的相似度搜索这个概念也可以类比

使用。模体挖掘现阶段可用于分类[10]、聚类[10][10]、异常检测[10][10]和规则挖掘[10][10]等任务，这些任务在生物、工业等领域都得到了广泛应用[10][10]。

而时间序列数据作为最常记录的数据形式之一，其量级十分庞大，在近年各领域飞速发展之时，海量的时间序列数据集也随之累计，等待人们挖掘其潜藏的信息价值。因此，对时间序列数据进行的模体挖掘被一众数据挖掘的专业人士在 2005 年的国际数据挖掘年会上列入了数据挖掘领域的十大挑战性问题之一[10]。

从时间序列模体挖掘的概念提出之后，该领域内发展出了许多针对时间序列数据的模体挖掘算法。这些算法主要分为三种类型：符号化表示的近似时间序列模体挖掘，基于聚类的时间序列挖掘方法以及结合图论的模体挖掘方法。首先，Eamonn Keogh 等人于 2002 年使用暴力算法(Brute Force Algorithm)实现了通过在子序列集合中搜索包含数量最多的个体作为时间序列模体[10]的挖掘算法，该算法思想简单粗暴、容易理解、准确性高，因此被广泛应用；然而，暴力算法也存在随着数据集的长度增加而计算成分增加速度过快的缺点，因此在之后针对暴力算法进行的优化算法被 Chiu 等人提出，该算法将符号化思想与概率论共同应用到时间序列的模体挖掘中去，被命名为随机投影算法(Random Projection Algorithm)[10]；聚类方法强调使用滑动窗口提取所有子序列，再使用各类聚类算法将其不断聚类得到结果，最后进行分析，然而此类方法被一些学者认为其结果缺乏实际意义的支撑[10]；图论方法的核心在于将模体挖掘问题转化为图中的问题，再使用图论手段解决，Lin 等人提出了子序列连接转化图的算法[10]，朱等人则提出了最大团转化法[10]，然而这类方法普遍存在中间过程复杂，计算成本高的问题。以上三种算法皆是基于支持度的算法，通过计算子序列满足相似度的数量来计算时间序列的模体。

除了基于支持度的算法外，Mueen 等人通过优化暴力算法，提出了基于相似度的时间序列模体算法 Mueen 算法并不断改进[10]；Yeh 等人则通过快速相似性搜索的使用提出了效率较高的时间序列挖掘算法，即 STAMP 算法[10]。

1.2.2 时间序列关联规则研究现状

关联规则最早是商业领域的概念[10]，用来记录顾客同时购买哪些商品，确认商品间存在的关联来制定商业策略和投资等决策。而时间序列关联规则是研究时间序列内部片段之间的变化趋势的预测手段，用来为某领域的预测与决策提供支持。

随着时间序列数据集在各领域的不断积累，如何从中进行规则挖掘这一问题也不断发展。通常经典的实践序列规则挖掘主要分为四步，如下图 1.1 所示，即数据预处理、数据压缩、规则获取和评价解释。

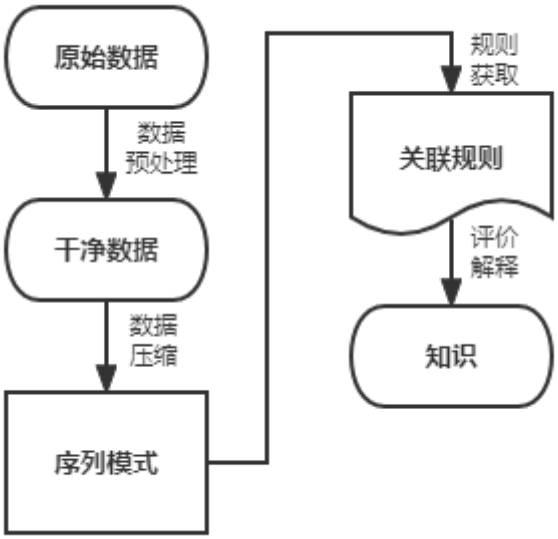


图 1.1 时间序列关联规则挖掘一般流程

国内外对于关联规则的研究不断发展的同时，有许多算法被提出，以算法的思想进行分类可以先大致分为串行、并行两类。首先，Agrawal 等人[10]最早提出了基于频繁项集的关联规则挖掘算法——Apriori 算法，而后随着针对 Apriori 算法的优化和不断发展，关联规则领域也迎来了理论不断的创新与探索。Park 等人[10]将散列哈希思想与经典算法相结合，用哈希来生成频繁项集，过滤了不满足支持度要求的子集，因此精简了候选集，该算法就是串行(DHP)算法，而后来 Han 等人[10]通过提出一种名为 FP-Tree 的数据结构采用分治算法自底向上构造，得以将节约空间成本，成为一种不产生候选集的挖掘算法，即 FP-growth 算法。在这几类算法基础上，学者们针对串行算法进行分布式处理，

得到了多种分布式算法。Agrawal 等人[10]在 Apriori 算法的基础上改进并提出了 CD、DD、CaD 三种并行算法，通过将候选集以不同方式分配至不同处理器不同程度地提升了相比串行计算的效率；而 Park 等人[10]在 DHP 算法的基础上进行哈希表的并行化，提出了 PDM 分布式挖掘算法。

此外，与 Apriori 算法核心不同的算法也不断涌现，基于数据流[10][10]、基于图[10][10][10]、基于序列[10][10]等不同思想的关联规则挖掘算法的提出，满足了关联规则挖掘在不同应用场景下的不同任务要求。

1.3 研究应用背景

深圳河全长 37km，流域总面积 312.5km^2。作为深港界河，深圳河在深圳一侧占 60%，覆盖了福田区、罗湖区、南山区和龙岗区的布吉街道和南湾街道等多个深圳市中心城区。

由于深圳湾水环境容量不足、水动力条件差，即便占比例不大的污水漏排入湾，也会对水质造成严重影响，具体包括：正本清源不彻底，导致面源污染物与雨水混合进入箱涵；降雨强度大，首场降雨将箱涵沉积物冲刷入河，造成河道水质下降；降雨量大，混合污水量超过污水处理厂的负荷能力，导致溢流；深圳河干流受各支流污染物影响，导致河口水质指标下降；受感潮影响，河口水质需要较长时间恢复。

城市水环境调度与防治工作一般通过水环境模拟来进行，但在对水质事件成因进行分析过程中，主要存在如下挑战：

- **决策影响因素多、数据资源散乱：**河道水质在暴雨场景下会受到多种因素影响，具体包括了降雨量、雨强及空间分布；水位、流量、流速等河道水情；近几日的河道水质、上游水质情况和河道底泥情况；深圳河口的潮时、潮高和潮位情况；需要同防汛业务协调调节一致，避免城市积涝等。这些不同影响因素的数据资源通过智慧水务大数据平台收集交换，但仍需要面向具体智慧应用目标整合封装，存在类型多样、存储分散，收集成本高等问题。

• **分析环境变化快：**伴随水环境治理工程建设密集开展，河道截污拦蓄联系和污染分布情况逐年变化，加剧水污染事件成因分析的复杂程度，也难以及时反映河道整理的最新进展。

本文将引入模体挖掘和规则挖掘等数据分析和挖掘手段，展开深圳河湾流域河道水质事件成因分析应用关键技术研究与应用服务。首先对河湾收集的散乱水质数据进行处理，再通过模式挖掘方法明确水质污染的决策影响因素之间的前后件关系，以在快速变化的环境中得以提前预测并防治污染。

1.4 研究目标和创新点

1.4.1 研究目标

综上所述，目前在理论方面，借由时间序列数据集挖掘得到模体序列，对模体序列进行关联规则挖掘的流程已经比较成熟，且有多种理论与算法可供选择；然而在实践方面，针对水利领域下垂直领域的时间序列数据挖掘还有反复尝试和改进的空间，本文将尝试采用经典的模体挖掘与规则挖掘算法对深圳河道的数据集进行模式挖掘和分析，将理论和实践相结合，得到规则挖掘的结果，为深圳市河道污染防治提供了决策支持的同时，也从方便关联规则挖掘的角度出发，对模体挖掘的算法进行改进，提出一种便于用作模式分析任务的时间序列模体挖掘算法。本文的主要研究目标包括：

（1）针对常规使用支持度阈值挖掘多个模体之间存在冗余模体，若用作关联规则挖掘会在序列符号化过程中出现冲突，浪费空间和计算效率且损失准确率这一问题，提出一种新的模体挖掘方法以解决该问题。文本通过使用去冗优化算法，对模体内部的相似子序列进行交叉匹配，剔除冗余模体，确保符号化不发生冲突。首先，设置滑动窗口并通过 Mueen 算法计算相似度，得到相似度矩阵；而后，设置相似度阈值和支持度阈值，保存满足支持度的窗口索引和对其满足相似度的所有子序列索引，保存在 P-Matrix Profile 数据结构中，得到 PMP1.0 元组；最后，通过去冗优化剔除 PMP1.0 中存在的冗余模体，得到 PMP2.0 元组，即模体集合。

（2）将整个工作流程投入实践应用，将深圳市河道水质/雨量作为研究对象。对其进行预处理、模体挖掘、关联规则挖掘、结果分析等处理得到模式分析结果，与深圳市本地实际结合，提供其河道中存在的周期性的污染规律，为污染和溢流的预测与防治提供决策支持。

本研究的技术路线如下**错误!未找到引用源。**2 所示。首先对数据集进行预处理，填补缺失值，对齐时间间隔，得到标准数据集；其次进行时间序列模体挖掘，设置窗口长度滑动划分子序列，使用 Mueen 算法计算相似度矩阵，设置相似度阈值与支持度阈值，计算满足相似度的子序列数量，大于等于支持度阈值则将窗口和子序列索引保存在 P-Matrix Profile 元组中，设为模体，得到 PMP 元组 1.0，再通过去冗优化剔除 1.0 元组中的冗余模体，得到精准的 PMP 元组 2.0；而后，将 PMP2.0 作为输入，将模体抽线为符号，对时间序列进行符号化，得到模体序列，对其使用 FP-growth 算法进行关联规则挖掘，得到规则集合；最后，对规则集合进行结果性分析，得到深圳市河道治理可预测的周期污染规律，为防治 提供决策支持。

1.4.2 创新点

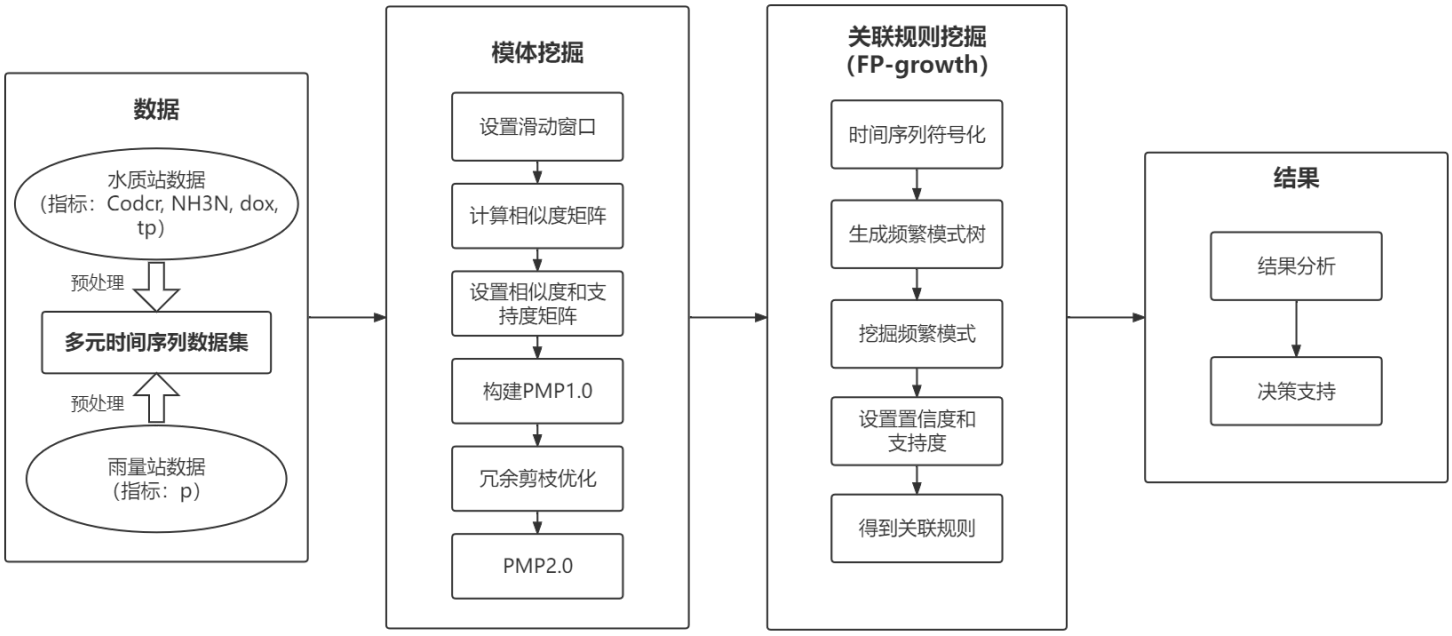


图 1.2 本文技术路线图

本文的主要创新点在于：

（1）提出了一种基于双阈值和去冗优化的时间序列模体挖掘方法。利用STOMP 算法计算出相似度矩阵，在相似度阈值的基础上加入支持度阈值构建元组 P-Matrix Profile，再通过去冗优化剔除模体集合中重复计算的模体元素，实现了更为精准、携带信息更丰富、更方便关联规则挖掘的模体挖掘工作流程。

（2）将改进后的模体挖掘算法应用到水利领域下的关联规则挖掘中去，改进了模式挖掘的效率，并将该方法实践，通过挖掘深圳市河道水质/雨量数据集得到了规则集，结合实际后分析得到了深圳市的污染模式周期规律，为防治决策提供了支持。

1.5 论文结构安排

本文的余下部分组织如下：第 2 章介绍多元时间序列的研究基础理论，包括有关多元时间序列相似度计算和关联规则挖掘的一些基础概念以及几个常用算法；第 3 章介绍模体挖掘的技术步骤、常用算法，提出了一种改进后的新模体挖掘算法，并将本文改进后的算法和经典算法进行多个指标的对比，证明其存在的优越之处；第 4 章应用第 2、3 章的模式挖掘方法，对深圳市河道水质和雨量数据集进行模式挖掘，分析关联规则挖掘成果，给出多元时间序列规则挖掘的实际应用，提供深圳河道管理的决策支持；第 5 章是对本次毕业设计的总结和展望。

第二章 基本理论概述

在介绍更为复杂的模式挖掘技术之前，本章首先要介绍本文的理论基础，即时间序列、规则挖掘的一些概念性知识。再者介绍几个典型算法。最后对从时间序列中挖掘关联规则的方法进行简要说明。

2.1 时间序列基础概念

2.1.1 什么是时间序列

时间序列是按时间顺序进行的观察结果的集合。

时间序列比大多数类型的数据更易于直观的观察。例如，看这个蓝色向量中的数字什么也告诉不了我们。但绘制数据之后，我们可以识别心跳，甚至可能诊断出这个人的疾病。当观测数据进行均匀采样时，观测指标可以代替观测时间。观测值可以有一个单位。

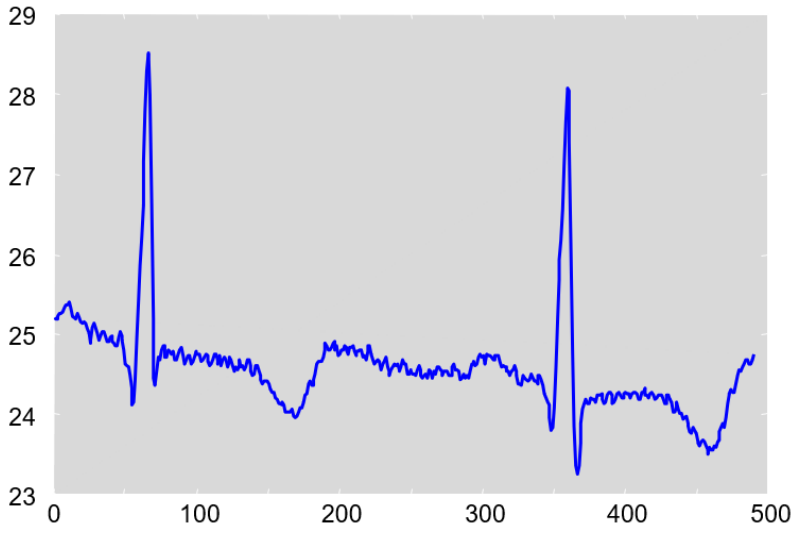


图 2.1 时间序列范例图

2.1.2 时间序列的标准化表示方法

实际上，所有时间序列数据集的每个子序列都必须是 z-标准化的。

在一些情况下，z-normalize 这一步没有实际上的含义，但在这些情况下如果这么考虑的话，其实相似度计算也没有类似的“实际含义”。举例来说，我们从两个不相关的人的心跳数据开始分析。如图 2.2 所示，即使没有标准化，两个序列的均值和标准差也几乎相同。如果这么看的话，我们还需要把它们标准化处理吗?

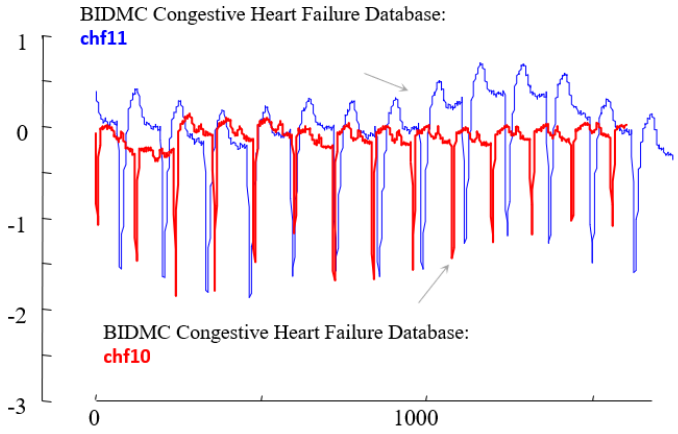


图 2.2 时间序列标准化例图

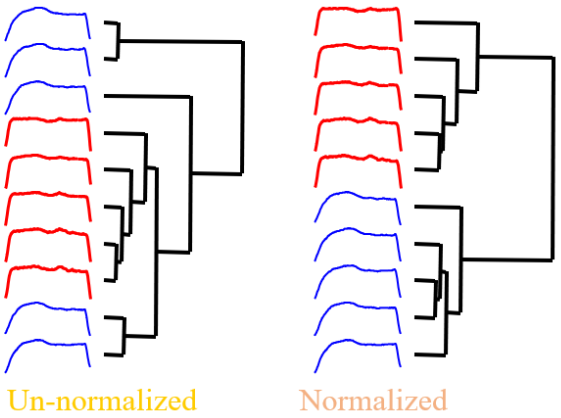


图 2.3 时间序列未标准化效果图

可以从图 2.3 左侧看到，这是没有标准化的结果，效果很差，一些蓝色的心跳相较于另一个蓝色的心跳而言，竟然更接近红色的心跳，这说明这些心跳的样本不具备自身的独特性，也就没有明显的特征可以提取。

而通过标准化处理后，如图 2.3 右侧，结果是完美的。这就告诉我们，即使找不到标准化这一步在实际中的含义，从数学的角度出发，为了得到更好的结果，标准化这一步也是十分必要的。

2.2 模体挖掘中的时间序列相似度计算问题

2.2.1 概述

上面最后构建的暴力算法是一种简单的模体挖掘中使用的时间序列相似度计算方法。基于 Matrix Profile 的时间序列模体挖掘算法——STOMP 算法是一个基于相似度计算的时间序列模体挖掘算法，其主要思路包括划分子序列、相似度计算和模体挖掘三个部分：首先，通过设置宽度为m的滑动窗口提取时间序列T的所有子序列；然后，利用矩阵存储所有子序列间的距离（欧氏距离）。最后，以距离矩阵为基础，提出新的数据结构 Matrix Profile 存储所有子序列对应的最小相似度距离探究时间序列T中基于相似度的模体。

Matrix Profile 是一个表征所有时间序列子序列与其最相似的子序列间距离的向量，其结构图如**错误!未找到引用源。**所示。图中，d_{ij}表示第i个子序列和第j个子序列之间的距离，$Min(D_i)$表示选取第i个子序列与其他子序列间距离的最小值，记为D_i，则 Matrix Profile 可表示为$(D_1, D_2, \dots, D_n)$。

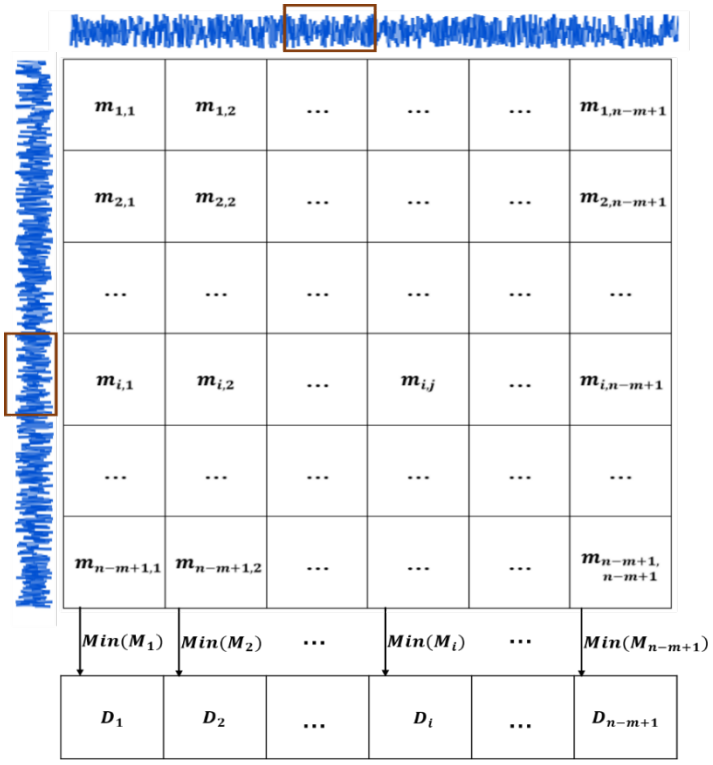


图 2.4 Matrix Profile 结构图

其中，距离矩阵的计算方法是计算所有长度为 m 的子序列之间的距离，我们可以使用欧几里得距离公式进行计算，但是在时间序列长度和维度都较大的时候，这种时间复杂度为 O(nm)的算法无疑是低效的，因此我们选择改进暴力算法，在这里使用 MASS(Mueen's Algorithm for Similarity Search)算法。

在暴力算法中，我们需要计算$\sum_{i=1}^{m} x_i y_i$，但是暴力计算复杂度为 O(nm)，我们可以使用卷积来计算，将其改进为时间复杂度为 O(nlogn)的更为高效的方式，这就是 MASS 的核心思想。

2.3.2 MASS 算法

如果 x 和 y 是多项式系数的向量，对它们进行卷积等于两个多项式相乘。

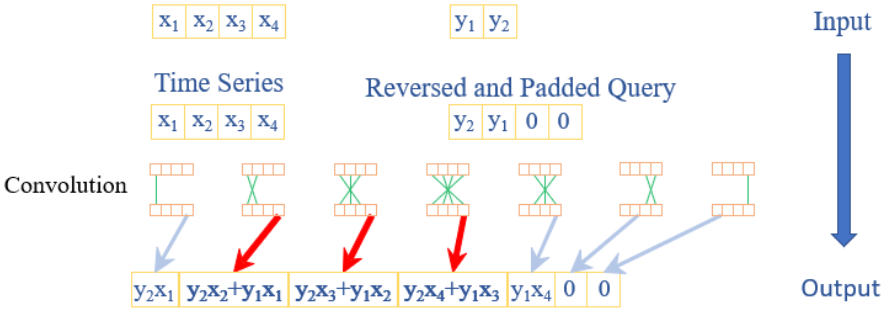


图 2.5 卷积法示意图

简而言之，卷积计算就是将 query 反向并补零，再交叉作乘得到一个包含多个点积的集合，高效得到$\sum_{i=1}^{m} x_i y_i$，改进过后的算法如下：

算法 2.1 MASS Algorithm 1.0

Algorithom 2.3: MASS (*T*, *query*)

输入：时间序列 T，query

输出：T 与 query 间的距离集合

```
d(1:n) = 0;
Q = zNorm(query);
Stdv = movstd(T,[0 m-1]);
Q = Q(end:-1:1);        //反转 query
Q(m+1:n) = 0;           //补零
dots = conv(T,Q);
dist = 2*(m-(dots(m:n))./Stdv));
dist = sqrt(dist);
d(1:n) = 0;
end
```

2.3.3 改进 MASS 算法

在计算卷积的时候，其实我们将输入的向量长度在输出的时候翻了一倍，但是 MASS 指用得到卷积输出的一半，另一半计算等同浪费。有没有什么办法只计算必要的一半而不管没用的一半呢，答案是肯定的。

引入傅里叶变换 FFT 和 IFFT。FFT (Fast Fourier Transform)是离散傅立叶变换的快速算法，可以将一个信号从时域变换到频域。同时与之对应的是 IFFT (Inverse Fast Fourier Transform)离散傅立叶反变换的快速算法。

我们知道，时域的卷积就是频域的乘法。那么我们这里的双倍长度了的卷积使用傅里叶变换即可转换为：

$$\textbf{conv(x,y)} = \text{ifft(fft(double\&pad(x)) . fft(double\&pad(y))}$$

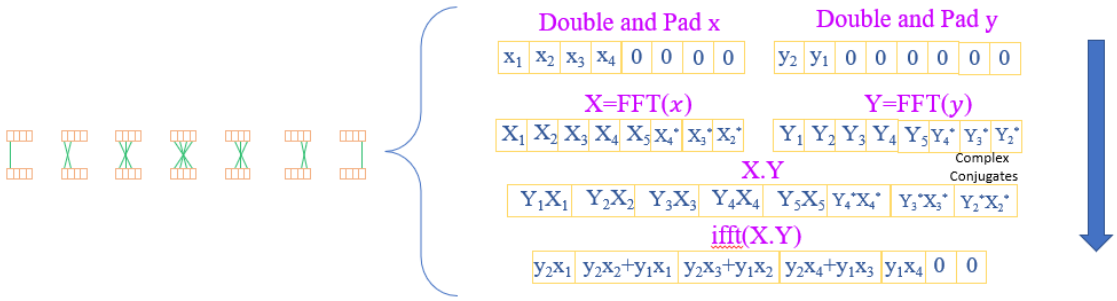


图 2.6 卷积双倍长度傅里叶变换示意图

如果我们不双倍 x 和 y，而是都保留一半的卷积，那么可以得到：

$$\textbf{half conv(x,y)} = \text{ifft (fft(x) . fft(y))}$$

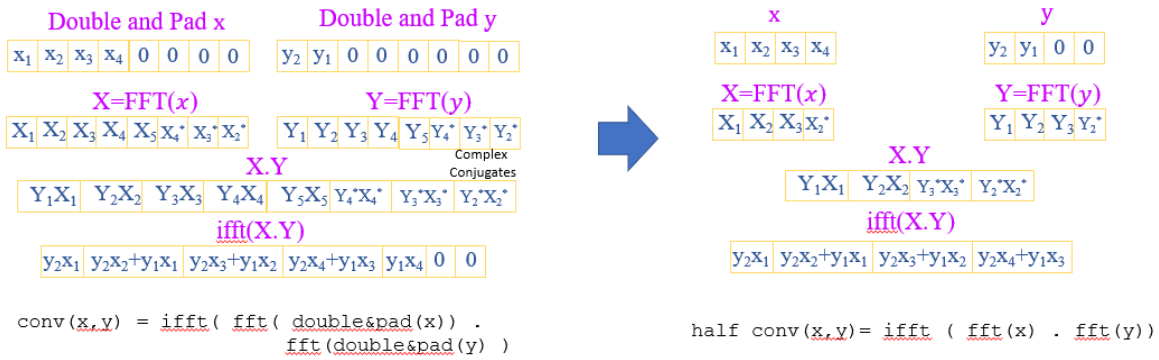


图 2.7 卷积傅里叶变换示意图

值得一提的是，卷积保留一半长度存在一个限制条件，即 n>m/2，不考虑这个条件的原因是我们一开始的假设就是建立在时间序列长度 n 远大于窗口长度 m 的情况下的。

最终改进后的 MASS 算法如下。

算法 2.2 MASS Algorithm 2.0

Algorithom 2.4: MASS (*T*, *query*)

输入：时间序列 T，query

输出：T 与 query 间的距离集合

```
1.  d(1:n) = 0;
2.  Q = zNorm(query);
3.  Stdv = movstd(T,[0 m-1]);
4.  Q = Q(end:-1:1);        //反转 query
5.  Q(m+1:n) = 0;           //补零
6.  dots = ifft( fft(T).* fft(Q) );
7.  dist = 2*(m-(dots(m:n))./Stdv));
8.  dist = sqrt(dist);
9.  end
```

该算法的最终复杂度为 O(nlogn)，与窗口长度 m 和维度均无关。

在本文的应用环节，最终也选用了 MASS2.0 作为相似度矩阵的计算方法，提高计算效率，在多元序列的矩阵计算中时间上的提升十分明显。

2.3 关联规则挖掘概述

针对时间序列数据的关联规则是由传统的关联规则挖掘发展而来，下面将着重介绍基本概念和传统方法。

2.3.1 基本概念

若设$I = \{i_1, i_2, \dots, i_n\}$表示项(*item*)的集合，包含$k$个项的集合称为$k$-项集。$D$表示由若干事务构成的数据集，用不同的$TID$标识事务，每个事务$T$是由一个或多个项构成的非空项集[11]，对应于$I$上的一个子集，即$T \subseteq I, T \neq \emptyset$。

关联规则一般表示为：$\{X \Longrightarrow Y\}$，X称为规则前件，Y称为规则后件，二者均为频繁项集(frequent itemset)，与规则生成和规则的评价解释统称为关联规则的三个主要步骤[错误!未找到引用源。]。频繁模式挖掘作为关联规则挖掘的前提，能够反映数据中反复出现的联系。频繁模式挖掘由支持度决定，而关联规则的生成则由支持度和置信度两者共同决定，规则的解释与评价则根据提升度来分析，下面简要说明上述基本概念[11]。

定义 2.1 支持度：在数据集D中，对于规则$X \Longrightarrow Y$，支持度表示项集X出现的事务个数占总事务个数的比值。其中，$count(X)$表示项集X出现的事务个数，$|D|$表示数据集D中包含的事务个数。

$$support(X) = \frac{count(X)}{|D|}$$

定义 2.2 频繁项集：频繁项集的频繁程度由项集出现的频率决定。若项集的支持度大于等于预先设置的最小阈值(min_sup)，则X是频繁项集：

$$support(X) \geq min_{sup}$$

定义 2.3 置信度：表示在规则前件X发生的条件下，规则后件Y发生的概率，其反应规则的可靠性：

$$confidence(X \Longrightarrow Y) = P(Y|X) = \frac{support(X \cup Y)}{support(X)} = \frac{count(X,Y)}{count(X)}$$

定义 2.4 提升度：通常用于衡量规则可靠性的指标，若提升度为 1 时，表示项集X和Y相互独立；若提升度小于 1 时，表示规则前件X和规则后件Y间呈负相关性；若提升度大于 1 时，表示规则前件X和规则后件Y间呈正相关性，规则前件X的出现促进规则后件Y的出现。

$$lift(X \Longrightarrow Y) = \frac{P(XY)/P(X)}{P(Y)}$$

定义 2.5 强关联规则：对于规则$X \Longrightarrow Y$，若同时满足支持度和置信度均大于等于最小支持度阈值和最小置信度阈值，则称该规则为强规则[11]。

2.3.2 经典算法

Apriori 算法作为典型的挖掘频繁模式算法，其主要思想源于先验性质：频繁项集的所有非空子集必定也是频繁的。因而采用逐层迭代的方式，选择通过k项集寻找$(k+1)$项集。

为克服 Apriori 算法会产生大量候选项集的缺点，FP-growth 算法采用分治策略递归搜索较短模式，通过连接后缀生成长频繁模式。

Apriori 算法和 FP-growth 算法的思想及优缺点对比分析如下**错误!未找到引用源。**所示。

表 2.1 Apriori 算法和 FP-growth 算法比较分析

	Apriori 算法	FP-growth 算法
算法过程	具体由两个子过程构成： (1) 通过扫描数据库D，累计每个项的计数并收集满足最小支持度的项，产生频繁 1 项集的集合，记为L_1； (2) 使用L_1找出频繁 2 项集的集合L_2，使用L_2找出L_3，如此下去，直到不能在找到频繁k项集。	主要基于两个主要阶段： (1) 将代表频繁项集的数据库压缩到一棵频繁模式树(FP-tree)，同时该树保留了项集关联信息； (2) 将经过压缩的数据库划分成一组条件数据库，每个数据库关联一个频繁项集，并分别对每个条件数据库进行挖掘。
算法优点	思想简单易于理解与实现。	避免了高代价候选产生，减少搜索开销。
算法缺点	(1) 需多次扫描数据库，时间复杂度较高； (2) 频繁项集连接过程可能产生较多候选项集，容易造成空间资源的浪费； (3) 仅适用于布尔型关联规则挖掘。	(1) 不适用于数据量较大的场景，否则将产生大量的频繁模式，占用一定的空间和时间资源，从而导致算法效率下降； (2) 挖掘频繁项集的过程需要反复遍历 FP-tree，保存相关信息需要大量额外的存储空间。

本文在应用部分将会选用 FP-growth 算法进行关联规则挖掘，首先是因为相较于 Apriori 算法的多次扫描，FP-growth 算法采用的压缩手段提升了搜索的效率，对于本文项目这样具有调参有较大需求的任务，在运行时间上具备优势；其次，本文的数据量在一条序列五千左右，规模并不是十分庞大，在 FP-growth 的适用范围内。

第三章 基于双阈值和去冗优化的时间序列模体挖掘方法

本章主要介绍的内容是针对时间序列数据的一种新型模体挖掘方法，即基于双阈值和去冗优化的时间序列模体挖掘。首先，本章将会介绍模体挖掘技术上的基本流程和思路，然后介绍相关的概念与定义；而后，将着重介绍本文提出的改进模体挖掘算法，具体包括滑动窗口计算相似度矩阵，提取基于阈值与支持度的 PMP 向量和去冗优化向量三个步骤。在文章结尾也会给出改进算法和常规算法的相关比对与分析。

3.1 概述

时间序列模体挖掘是指不依赖任何先验信息，从时间序列中挖掘出重复出现的未知模式[25]。本文经过阅读和研究发现，现有的时间序列模体挖掘方法仍存在较多问题，包括在对序列进行符号化处理时，其符号缺少支持度支撑其精准度；建立的 matrix profile 中存在处于阈值边缘的多处冗余元素等等问题。针对上述问题，本文提出了一种改进的模体挖掘算法 PMM(P-Matrix Profile Motif Mining Algorithm)，包括三个步骤。首先是计算相似度矩阵，根据所要研究的时间周期和实际含义确定滑动窗口，运用 stomp 算法进行窗口和所有子序列的相似度计算并存入相似度矩阵；然后设置对合适的阈值，控制支持度和模体的精确度和代表性，从相似度矩阵中提取出本文提出的 P – Matrix Profile, 一种包含在阈值范围内子序列坐标，符合支持度要求等信息的元组序列，初步筛选出模体；最后通过去冗，剪除初筛模体中不必要的冗余模体元素，得到更为精确的 pmp 模体集合，方法流程具体如图 3.1。

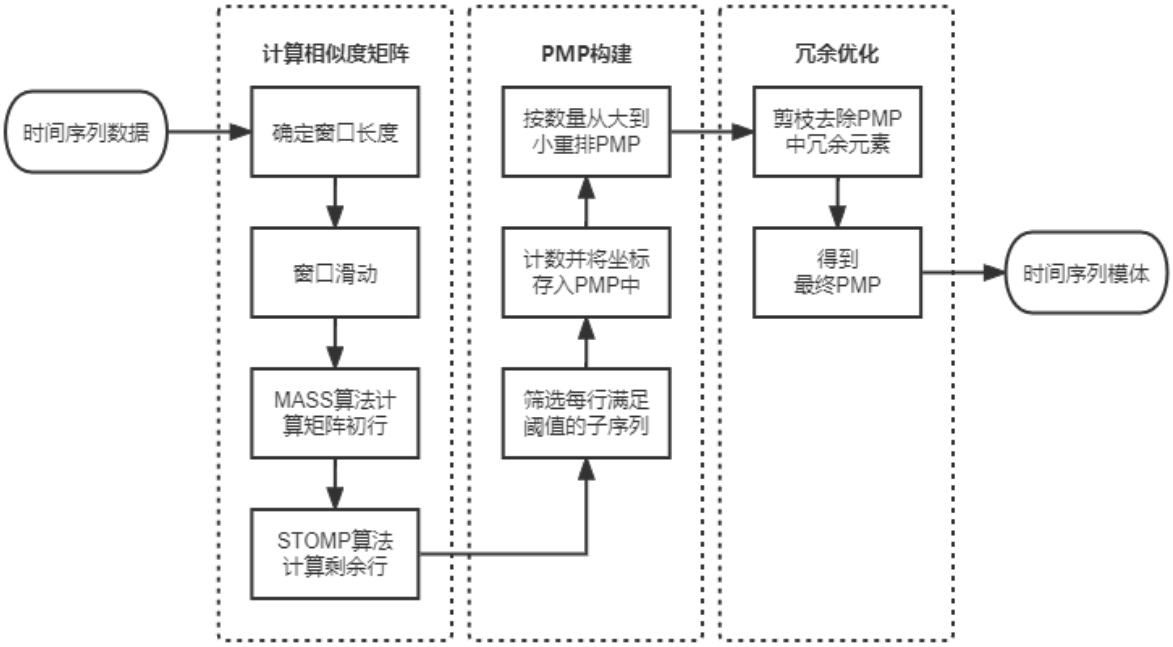


图 3.1 模体挖掘流程图

3.2 相关定义

本章方法实现过程主要涉及如下概念，下面进行详细说明。

定义 3.1 滑动窗口[25]：给定长度为n的时间序列T，固定滑动窗口宽度m，滑动窗口每次只移动一个数据，由该滑动窗口提取T中所有长度为m的子序列。

定义 3.2 时间序列子序列集：对于时间序列$T = \{v_1, v_2, \dots, v_i, \dots v_n\}$，采用滑动窗口依次顺序提取所有长度为$m$的子序列$S_i = \langle v_i, v_{i+1}, \dots, v_{i+m-1}\rangle$构成的集合，记为子序列集合$A$，$A = \langle S_1, S_2, \dots, S_{n-m+1}\rangle$，其中$S_i$可用$A[i]$表示，$i$是子序列$S_i$的起始点位置且满足$1 \le i \le n-m+1$，$m$是子序列长度（即滑动窗口宽度）。

定义 3.3 平凡匹配[25]：给定时间序列T和相似度阈值$threshold$，假设T存在起始位置为p的子序列S_p和起始位置为q的子序列S_q，若满足$D(S_p, S_q) \le threshold$，$D$为子序列间距离度量函数，则称子序列$S_p$和$S_q$匹配，即$S_p$为$S_q$的匹配子序列。若$p = q$，则$S_p$和$S_q$为平凡匹配；若$p \ne q$，且不存在起始位置$q'$的子序列$S_{q'}$，满足$q < q' < p$或$p < q' < q$,使得$D(S_p, S_{q'}) > threshold$，那么$S_p$和$S_q$为平凡匹配。其中，相似度阈值$threshold$需要预先设置。如下**错误!未找到引用源。**2 所示为平凡匹配的一个示例。

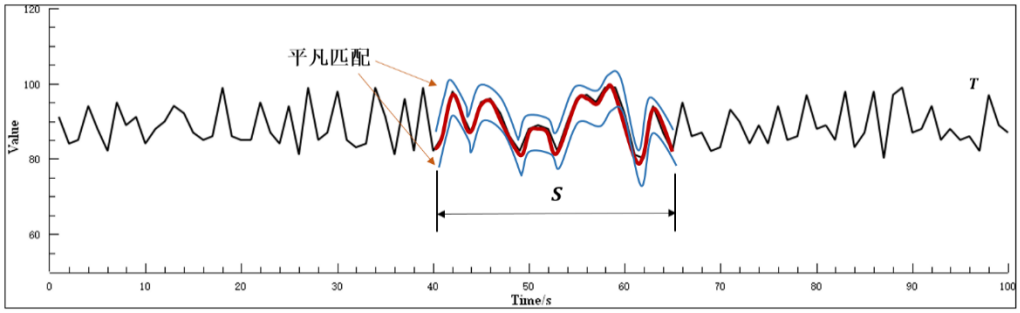


图 3.2 平凡匹配示例

定义 3.4 距离矩阵：给定时间序列T，通过相似性度量公式计算子序列集A中所有子序列间的距离并存入矩阵M。距离矩阵M为对称矩阵，通常表示为：

$$M = \begin{bmatrix} m_{1,1} & \cdots & m_{1,n-m+1} \\ \vdots & \ddots & \vdots \\ m_{n-m+1,1} & \cdots & m_{n-m+1,n-m+1} \end{bmatrix}$$

其中m_{ij}用于表示子序列$A[i]$和$A[j]$间的距离。

定义 3.5 Matrix Profile[45]：若存在距离矩阵M和相似度阈值$threshold$，提取距离矩阵M中所有子序列与其他子序列（剔除平凡匹配）满足相似度阈值的最小距离并存入向量 Matrix Profile，即向量 Matrix Profile 中保存着子序列与其最相似子序列间的距离。

定义 3.6 支持度：本文定义若存在距离矩阵M，相似度阈值 threshold，则一个滑动窗口序列$\boldsymbol{S_q}$的支持度为距离矩阵 M 该行中满足$D(S_p, S_q) \le threshold$的元素个数，提出该标准是为了确保模体必须满足自身具有特征性的同时反复出现，具有一定代表性。

定义 3.7 P-Matrix Profile：本文定义若存在距离矩阵M，相似度阈值 threshold，提取距离矩阵M中所有子序列与其他子序列（剔除平凡匹配）满足$D(S_p, S_q) \le threshold$的子序列索引和相似度存入元组 P-Matrix Profile，记为$PMP = \langle 1:\{i:d_{\mathrm{i}}, \dots, j:d_{\mathrm{j}}\}, \dots, \mathrm{i}:\{m:d_{\mathrm{m}}, \dots, n:d_{\mathrm{n}}\}, \dots, \mathrm{n}:\{d_{\mathrm{s}}, \dots, d_{\mathrm{t}}\}\rangle$，其中 i 为子序列索引，$d_{\mathrm{i}}$为该子序列和窗口的相似度，完成 PMP 初筛。

定义 3.8 时间序列模体[17]：给定时间序列T，相似度阈值$threshold$以及支持度阈值$limit$，若存在某个子序列集合A，其包含子序列彼此满足$D(A[i], A[j]) \le threshold$的数量最多，即支持度最高，则称子序列集合$A$中的子序列为时间序列$T$的模体(1-motif)。根据 P-Matrix Profile 结构性质，P_A中长度最大值对应子序列

及与其相似即为时间序列模体。本研究中，同时满足了支持度和相似度阈值的子序列集合中的子序列都规定为模体(p-motif)，即要满足$len(PMP_i) \geq limit$。

定义 3.9 冗余模体：给定时间序列T和相似度阈值$threshold$，对于 T 的模体(1-motif)而言，其支持度必然是子序列集合中最高的。假设模体集合为P_i,且有$S_i, S_j \in P_i$,其中S_i代表计算时滑动窗口所在的子序列，S_j代表集合中一个满足$D(S_i, S_j) \leq threshold$相似度阈值的子序列，则必然存$S_j \in P_j$,即$S_j$作为滑动窗口的相似度向量，因为$S_i, S_j \in P_i$，即$S_i$与$S_j$再某阈值内是相似的，可能存在一$S_k$满足：$S_k \in P_i$且$S_k \in P_i$，即$S_k$在计算支持度时重复被计算了多次，这带来了问题，按照符号化的规定时，我们将把不同集合中的S_k分别化为不同的两个模式，这显然对于后续的规则推导来说，是一种精准度上的损失。因此，在符号化之前，我们有必要将这些冗余模体剔除，将元组 P 中的元素按包含个数从大到小排列，即可遍历剔除低频集合中的冗余模体。

综上所示，本文约定使用的部分数学符号如**错误!未找到引用源。**所示：

表 3.1 本章部分符号表示

符号	含义
$T = \{v_1, v_2, \dots, v_i, \dots v_n\}$	时间序列
$S = \{v_i, v_{i+1}, \dots, v_{i+m-1}\}$	时间序列子序列
$A = \{S_1, S_2, \dots, S_{n-m+1}\}$	时间序列子序列集
M	距离矩阵
m	模体长度（子序列、滑动窗口长度）
D	相似度计算函数
$threshold$	相似度阈值
$limit$	支持度阈值

3.3 基于双阈值和去冗优化的模体挖掘方法

本章提出的模体挖掘方法主要步骤分为计算相似度矩阵、PMP 构建和冗余优化三部分。首先，用 stomp 算法可以计算出相似度矩阵；然后在有了相似度矩阵的情况下可以判定相似度是否满足阈值 threshold，再继续判断每个子序列集合是否满足支持度阈值 limit，最终把满足两个阈值的子序列和相似度存入 PMP 元组，得到 pmp1.0；最后对 pmp1.0 进行去冗优化处理，优化后得到最终的 P-Matrix Profile 模体元组，包括每个模体的母体索引以及相应的模体相似度和模体索引。本章所述算法如算法 3.1 所示.

算法 3.1 时间序列模体挖掘算法

Algorithom 3.1: PMM(T, window, *threshold, limit)*

输入：时间序列T，窗口长度$window$，相似度阈值$threshold$，支持度阈值$limit$

输出：基于双阈值和去冗优化的模体元组 pmp

1. $n = length(T)$ //n为时间序列长度
2. $mp = \text{stomp\_p}(T, window)$ //计算相似度矩阵 mp
3. $pmp = build\_p(\text{mp, threshold, limit})$ //构建 pmp 元组 1.0
4. $pmp = prun\_opt(pmp, n)$ //去冗优化冗余模体以优化 pmp 准确性
5. **return** pmp

3.3.1 相似度矩阵计算

相似度矩阵的计算是模体挖掘的数据基础，STOMP 算法可以很好地满足这个需求，通过移动均值和标准差算法在 O(n)时间内算出均值与标准差后，只需 O(1)时间即可快速计算出时间序列的相似度矩阵。

第二章中我们提到过时间序列的相似度计算公式，观察公式我们就可以发现：

$$d_{i,j} = \sqrt{2m\left(1 - \frac{T_iT_j - m\mu_i\mu_j}{m\sigma_i\sigma_j}\right)}, \qquad T_iT_j = \sum_{k=0}^{m-1} t_{i+k}t_{j+k}$$

我们可以通过 movmean 和 movstd 这两个经典的移动算法在 O(n)时间内首先算出每一行的均值和方差，在此基础之上，假设我们还能知道T_i和T_j，那么在 O(1)时间内就可以算出$d_{i,j}$。

STOMP (Scalable Time Series Ordered Matrix Profile)算法可以满足这一点要求。该算法发现了相似度矩阵中的邻行关系，即T_iT_j与$T_{i+1}T_{j+1}$之间的数学关系，如图 3.3 所示。

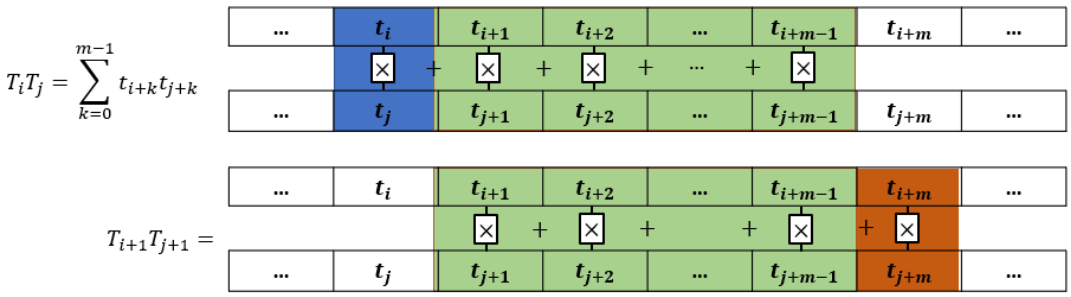


图 3.3 相似度矩阵邻行关系示意图

该关系的数学表达式为：

$$T_{i+1}T_{j+1}=T_iT_j-t_it_j+t_{i+m}t_{j+m}$$

STOMP 算法中，首先用第二章中所提到的 MASS 算法计算出矩阵的第一行数据，由矩阵的对称性第一列和第一行的数据是完全相同的，加上有了提前计算的均值和方差，而后只需用该公式逐行迭代即可计算出剩余的矩阵数据，最终返回每个与窗口最相似的子序列的相似度和其索引，迭代过程如图 3.4 所示。

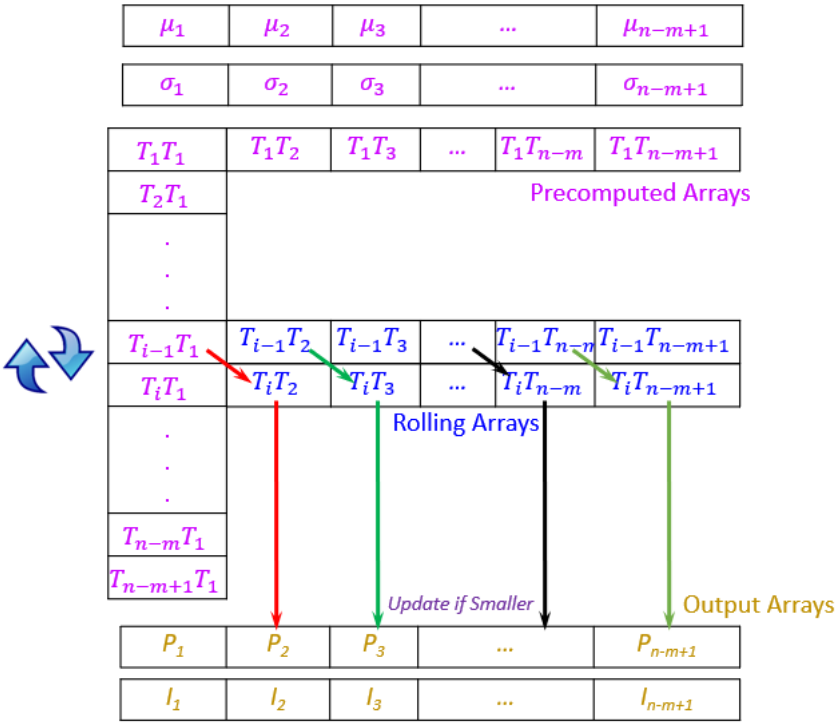


图 3.4 逐行迭代计算示意图

最终得到了一个时间 O(n^2)，空间 O(n)的算法，相似度计算部分的实现具体如算法 3.2 所示，STOMP 算法得到模体的总流程如算法 3.3 所示。

算法 3.2 STOMP 相似度矩阵部分算法

Algorithom 3.2: STOMP_DP(*T*, *m*)

输入：时间序列*T*,子序列长度*m*

输出：相似度矩阵 DM

```
n = len(T)
for i = 1:n-m+1
    mean = movmean(T(i:i+m-1))
    std = movstd(T(i:i+m-1))
first_row = MASS(T, T(i:i+m-1))
DM = iterate(first_row, mean, std)
return DM
```

算法 3.3 STOMP 算法流程

Algorithom 3.2: STOMP(*T*, *m*, *threshold*)

输入：时间序列T,子序列长度m，相似度阈值 threshold
输出：模体向量 MP

```
1.  n = len(T)
2.  MP(1:n-m+1) = inf;
3.  MPI(1:n-m+1) = -1;
4.  DM = STOMP_DP(T, m)
5.  for i in n:
6.      if DM[i] < threshold:
7.          MP[i] = min(DM[i])
8.  return MP
```

虽然 STOMP 算法在相似度矩阵的计算上进行了优化，但其最终输出仍是各个序列中相似子序列个数最多的单个模体，对于序列中存在多个潜在规律的数据，这样无疑在无形中浪费了数据价值，也对后续的规则挖掘的精度造成了损失。因为有必要将支持度引进，并在输出时保存多个模体，所以既要提出新的数据结构保存模体，也要对相似度计算部分加以改进，以记录支持度，为新的数据结构 PMP (P-Matrix Profile)元组的构建提供支持，对于该数据结构的构建方式在 3.3.2 中将会具体阐述，这是一个双层字典，记录了每个序列的多个模体序列和其相似子序列的索引以及相似度。因此这里对相似度矩阵的计算算法进行改进，在 STOMP 算法的基础上，加入支持度的记录过程，具体算法流程如下算法 3.4 所示。

算法 3.4 基于 stomp 算法改进的距离矩阵计算算法流程

Algorithom 3.2: stomp_p(*T*, *m*, *threshold*)

输入：时间序列T,子序列长度m，相似度阈值 threshold
输出：距离矩阵 DM

```
1.  n = len(T)
2.  DM = STOMP_DP(T, m)
3.  LIMIT = []
3.  for i in (0, n)   //记录支持度
4.      if DM[i] < threshold
5.          LIMIT[i] ++
7.  return DM
```

3.3.2 PMP 构建

通过 STOMP 得到距离矩阵后，本算法将开始构建 P-Matrix Profile1 元组 1.0。其原型来自 Yeh 等人[45]利用子序列距离矩阵，提出了 Matrix Profile 结构挖掘基于相似度的时间序列模体。本研究提出通过子序列距离矩阵构建 P-Matrix Profile 结构，在原型 Matrix Profile 的结构之上设置了相似度和支持度双阈值来进行时间序列模体挖掘，进一步提升了挖掘得到的模体的精准性与典型性；原型 MP 是向量形式，保存最大相似度，而 PMP 是元组形式，加入了相似度和索引，且很容易通过 len 方法得到支持度大小，提升了模体的信息丰富度，为规则推导提供了配套数据，提升了模体挖掘方法用作关联规则推导的便捷性。

Matrix Profile 保存了所有子序列最小距离的一维向量，其中最小值所对应的索引为时间序列基于相似度的模体，即时间序列中距离最小（最相似）的子序列对。本研究提出的 P-Matrix Profile 使用相似度和支持度双阈值，一是使用相似度阈值防止挖掘的模体集合中存在相似度不够的假阳性结果，提升了模体的相似性，即准确性；二是使用支持度阈值，确保模体的出现频次大于阈值，防止了模体在整个时间序列的长度上出现频次过低，因关联规则推导使用频次过低的模体不具备决策支持的意义，设置了支持度阈值可以有效提升模体的典型性，使得关联规则推导的推导式具备更高的预测效率。

根据上述步骤，给出相似度阈值$threshold$与结合支持度阈值$limit$，利用第一步计算的相似度矩阵$Distance\ Matrix$，构建 P-Matrix Profile 结构，具体如下**错误!未找到引用源。**所示。

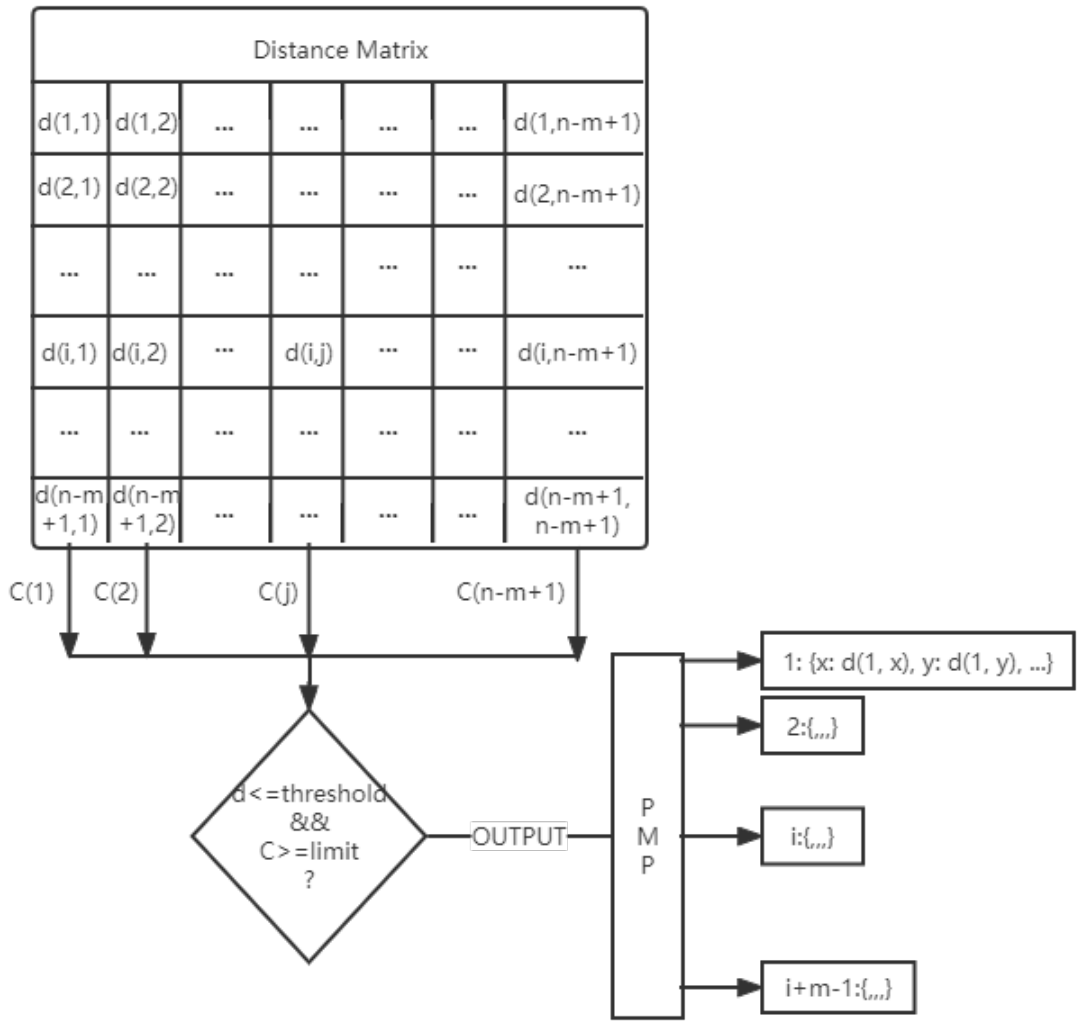


图 3.5 P-Matrix Profile 结构示意图

由定义 3.7 可知，P-Matrix Profile 是基于相似度矩阵的元组，其记录了相似度矩阵DM中所有子序列与其他子序列（剔除平凡匹配）满足相似度阈值$threshold$的子序列索引，相应相似度，和支持度大小。时间序列模体挖掘的具体过程如算法 3.3 所示。

算法 3.3 PMP 模体构建算法

Algorithom 3.3:build_p(*T*, *threshold*, *limit*)

输入：时间序列T，相似度阈值$threshold$，支持度 limit

输出：时间序列模体 PMP1.0

1. $n = length(T)$ //n为时间序列长度
2. $PMP = infs, D_M = infs, I_M = zeros, idxes = 1{:}n - m + 1$
3. DM = stomp_p(T, m)
4. **for each** col **in** DM

```
5.      c = 0, temp = infs
6.      for each index in idxes
7.        if d(index) ≤ threshold //满足相似度阈值加入临时变量
8.            temp.append({index: d(index)})
9.            c = c + 1 //支持度计数
10.     end for
11.     if c≥ limit //满足支持度才存入 PMP
12.       PMM[col].append(temp)
13.   end for
14.   return PMP
```

3.3.3 去冗优化

通过第一和第二步的处理我们已经得到了一个时间序列模体集合 PMP1.0，但是其准确性还有提升的可能性。如定义 3.9 所述，PMP 当前的状态无疑是存在多处冗余模体的。第三步的目标就是剔除这些冗余模体，使得关联规则推导过程中的符号表达更为精准。实现的过程就是从 PMP 中最长的模体集合开始，查找其中的元素作为窗口计算子序列相似度时，是否存在当前模体集合中相同的子序列索引，即是否有被重复计算的子序列。若有，则在较短的集合中将其删除，因为模体在关联规则推导充当前件或后件时都始终应当保证其典型性，而维护一个较长的模体集合可以很好的保持其典型性。在循环一遍的过程中，我们还可以继续去冗，较短的模体集合在剔除了重复模体后，有可能支持度不符合 limit 的要求了，这时候就应当将它们整体清零。最后就得到了一个不存在冗余模体，也二次符合支持度阈值的模体集合，即 PMP2.0。具体优化的实现步骤如算法 3.4 所示。

算法 3.4 去冗模体优化算法

```
Algorithom 3.4: prun_opt(PMP, n)
输入：时间序列模体集合PMM，时间序列长度n
输出：去冗的模体集合 PMP2.0
1.    indx = 1:n − m + 1
      Sort(PMM) //按长度降序排列
2.    for each ind in indx //从长到短匹配
        for each element in PMM[ind] //对集合中的每个子序列查重
          if Dup_exsit(PMM[ind], PMM[element.index]) //判断是否存在冗余
            Drop_duplicate(PMM[element.index]) //有冗余就在短集合中清除
          if len(PMM[element.index]) < limit //再次判断是否满足支持度阈值
            PMM[element.index]).drop //不满足就清除集合
        end for
```

end for

11. **return** *PMP*

3.4 实验设计与结果分析

为测试本章所提出的算法的效率，现将设计实验使用改进后的 PMM 模体挖掘算法与目前主流的几个算法进行模体挖掘的运行试验，并就其结果进行对比分析，明确所提出算法的优劣之处。

3.4.1 实验环境

本章实验在 Windows 11 操作系统下进行，具体软硬件环境如下**错误!未找到引用源。**所示。

表 3.2 实验软硬件环境

名称	相应标准
处理器	AMD Ryzen 7 5800H
内存	16GB
开放工具	PyCharm 2021
编程语言	Python 3.10

3.4.2 数据集

本章实验部分使用深圳市河道水质站和雨量站的部分数据，具体信息如**错误!未找到引用源。**所示：

表 3.3 数据集说明

序号	数据集名称	长度	简述
1	蔡屋水质站数据集	5484	CWS，该水量站 2020.10.01~2021.12.31 的每两个小时的水质指标数据，包括 codcr,nh3n,dox,tp 四项指标。
2	大小坑水质站数据集	5484	DXS，该水量站 2020.10.01~2021.12.31 的每两个小时的水质指标数据，包括 codcr,nh3n,dox,tp 四项指标。
3	汇入口水质站数据集	5484	HRS，该水量站 2020.10.01~2021.12.31 的每两个小时的水质指标数据，包括 codcr,nh3n,dox,tp 四项指标。
4	泥岗桥水质站数据	5484	NGS，该水量站 2020.10.01~2021.12.31 的每两个小时的水质指标数据，包括 codcr,nh3n,dox,tp 四项指标。
5	粤宝路桥水质站数据集	5484	YBS，该水量站 2020.10.01~2021.12.31 的每两个小时的水质指标数据，包括 codcr,nh3n,dox,tp 四项指标。
6	布吉雨量站数据集	5484	BJY，该雨量站 2020.10.01~2021.12.31 的每两个小时的雨量数据，包括雨量 p 一项指标。

7.	布吉河口雨量站数据集	5484	BHY，该雨量站 2020.10.01~2021.12.31 的每两个小时的雨量数据，包括雨量 p 一项指标。
8.	草埔雨量站数据集	5484	CPY，该雨量站 2020.10.01~2021.12.31 的每两个小时的雨量数据，包括雨量 p 一项指标。
9.	鹿丹村雨量站数据集	5484	LDY，该雨量站 2020.10.01~2021.12.31 的每两个小时的雨量数据，包括雨量 p 一项指标。
10.	笋岗闸上雨量站数据集	5484	LGY，该雨量站 2020.10.01~2021.12.31 的每两个小时的雨量数据，包括雨量 p 一项指标。

3.4.3 实验方法与结果分析

本章基于上述十个数据集，实验分为两步进行：

第一步，参数设置说明。第二步，从准确性、可拓展性和鲁棒性三个方面，将本文提出的时间序列模体挖掘算法 PMM 与现有算法进行比较，包括基准算法 Brute Force(BF)算法及原型算法 STOMP 算法[错误!未找到引用源。]、目前最新算法 SCRIMP++算法[错误!未找到引用源。]。其中，PMM 算法存在两次模体的筛选过程，因此在对比过程中分为两次统计，以更好地观察其每次筛选的数量是否符合实际情况。虽然 SCRIMP++算法只能挖掘基于的相似度的时间序列模体，但其距离计算过程采用了 Matrix Profile 结构，因此本文将主要对比子序列距离矩阵计算模块，最终模体挖掘模块仍采用本文所提方法。

为了满足算法结果的基本需要，使得对比实验的结果更清晰明显，对算法所需的众参数进行了设置。具体如下**错误!未找到引用源。**所示。表中展示了不同数据集的具体参数设置，本实验使用了汇入口水质站数据集(HRS)的四个指标以及布吉雨量站数据集的一个指标，共五项数据，各项数据长度均为 5484。因为实验中数据处理为两小时间隔，因此将窗口长度 m 设置为 12，以挖掘 24h 内变化的模体规律。支持度阈值 limit 均设置为 4，以此过滤一些非典型的模体。相似度 threshold 根据 PMM 算法的结果控制在 50~60 模体，维持在同一水平线以突出图表上的对比结果。

表 3.4 各数据集参数设置说明

序号	数据集名称	指标	m	$limit$	$threshold$
1	汇入口水质站数据集(HRS)	codcr	12	4	0.35
2	汇入口水质站数据集(HRS)	nh3n	12	4	0.32
3	汇入口水质站数据集(HRS)	dox	12	4	0.36

4	汇入口水质站数据集(HRS)	tp	12	4	0.4
5	布吉雨量站数据集(BJY)	p	12	4	0.45

实验 1：测试本章方法的准确性

为验证本文算法的准确性，采用 Brute Force 算法作为基准算法，比较分析不同算法挖掘得到的模体数量，具体参数设置情况见**错误!未找到引用源。**。

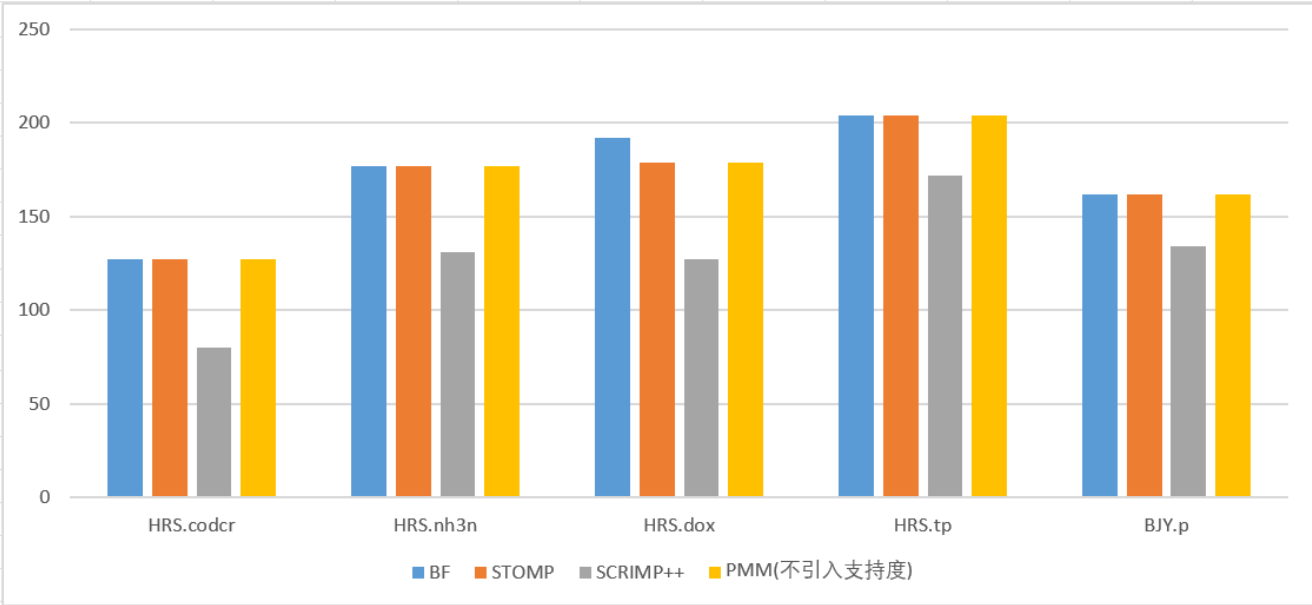


图 3.6 各算法挖掘模体数量对比结果

比较准确度时，可以用挖掘模体得到的模体数量是否基本一致来判断一个算法结果上是否准确，但是引入支持度无疑会大大削减模体的数量，所以在比较算法准确度时需要将 PMM 算法的支持度削减部分暂时停用，进行到相似度阈值前，将结果与其他算法的数量进行比较。

由**错误!未找到引用源。**所示实验结果可知，以 Brute Force 算法为基准算法，STOMP 算法和 SCRIMP++算法的模体数量基本相同，因为三者皆未引入支持度阈值，所以仅有相似度阈值筛选后数量基本不变，而 SCRIMP++在这三者间又略少的原因在于其是近似算法，所以在得到相似度的过程中可能存在精度损失。而 PMM 和其他算法在同样不引入支持度情况下，挖掘模体的结果数量基本相同，可见 PMM 算法挖掘出的模体具有符合标准的**准确性**。

实验 2：测试本章方法的冗余性

因为本文要解决的挑战在于，若要在时间序列数据的单一序列中挖掘多个模体，则在后续符号化过程中模体间会存在重复索引，导致会在同一时间点复写不同符号的冗余模体，因此计算模体的冗余性是验证算法有效的关键所在。

通过实验 1 得到了各个算法对于同样数据集的数量上的挖掘结果，可以利用该结果进行冗余度的计算。若两条模体内存在相同索引的子序列，则将长度较短的模体记为冗余模体。冗余度=冗余模体数量/模体总数。

由下图 3.7 可见，通过对实验 1 的数量结果进行分析，分别计算各个算法的冗余度，传统算法的冗余度在 4.52%～16.54%之间，而 PMM 因为存在将冗余模体全部剔除的步骤，其内部完全不存在冗余模体，因此冗余率为 0.00%。

冗余率一旦大于 0，在符号化过程中必然出现同一时间点被符号化为两个不同符号的问题，对规则挖掘也必然造成准确度的损失。**经实验 2 验证，PMM 算法具备零冗余度。**

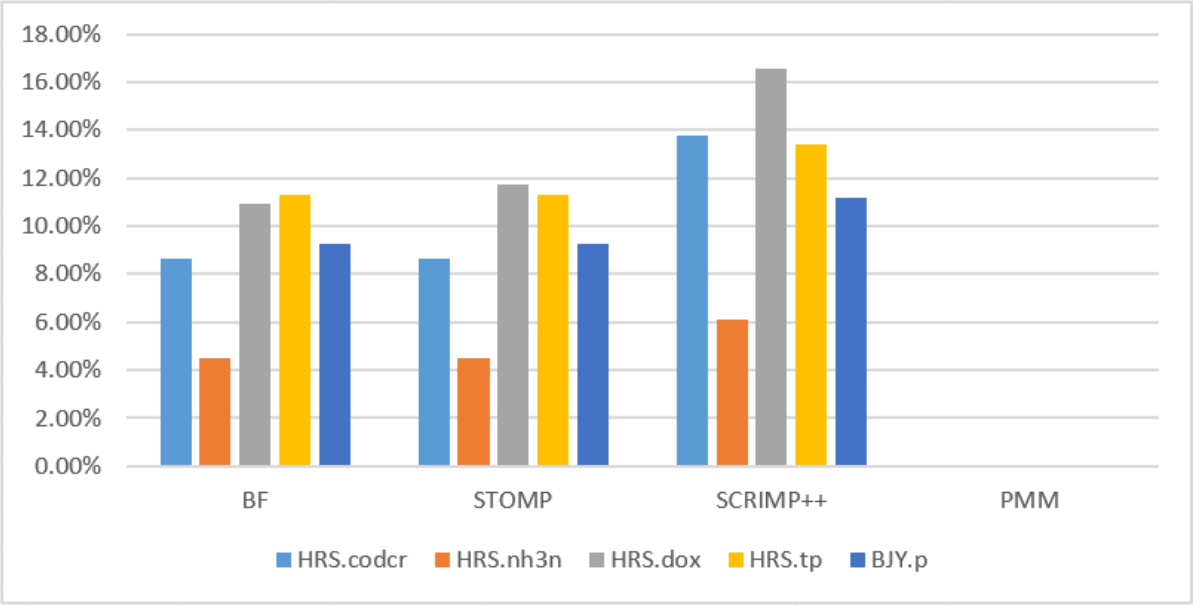
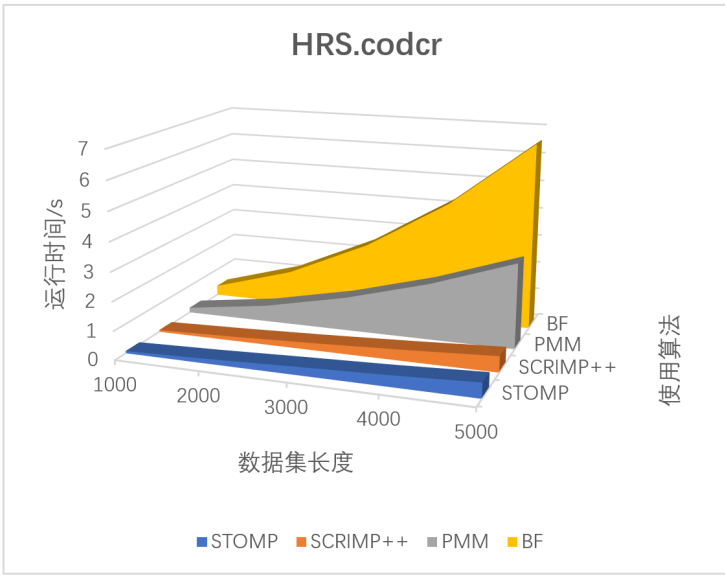


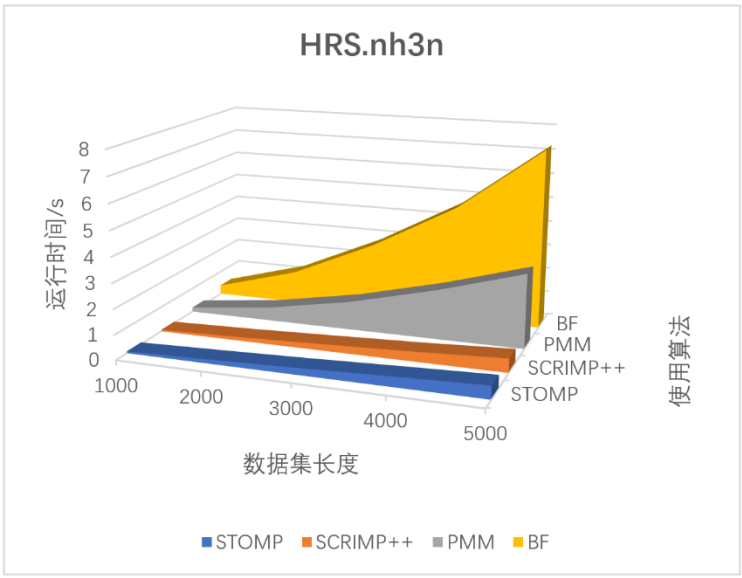
图 3.7 各算法冗余度计算结果对比

实验 3：测试本章方法可扩展性

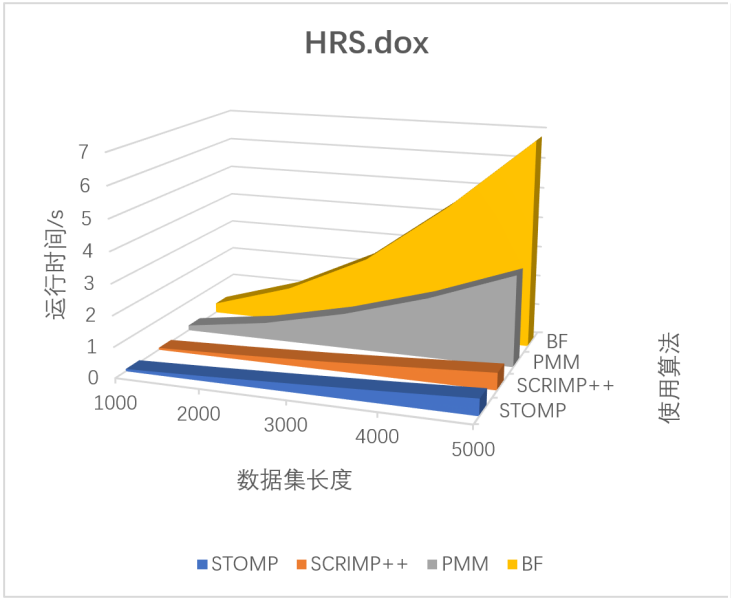
为验证本文算法的可拓展性，基于相同的模体长度，在各数据集不同长度数据集情况下，比较分析 Brute Force 算法、STOMP 算法、SCRIMP++算法和本文 PMM 算法共 4 个算法挖掘模体所需的运行时间。其中，模体长度不变，均设为 m=12，然后以固定步长（1000）增加实验数据集的长度。可拓展性实验结果如下**错误!未找到引用源。**8 所示。



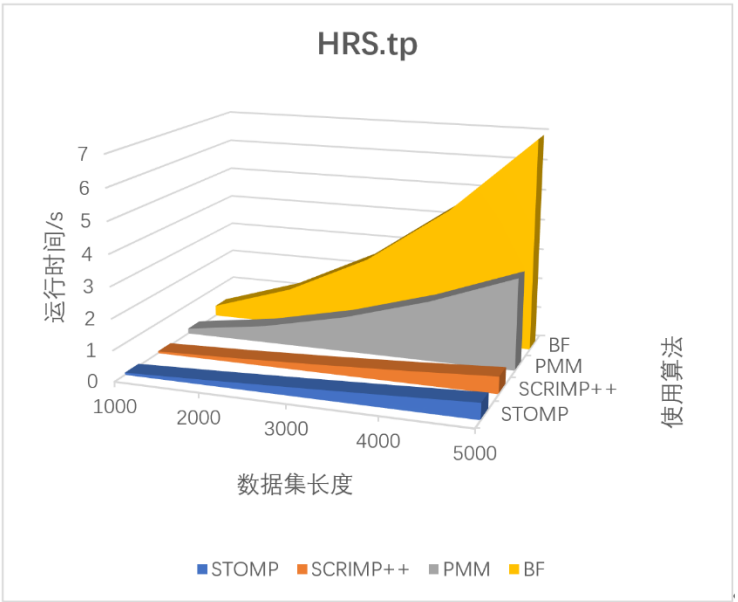
(a)HRS 数据集 codcr 指标数据



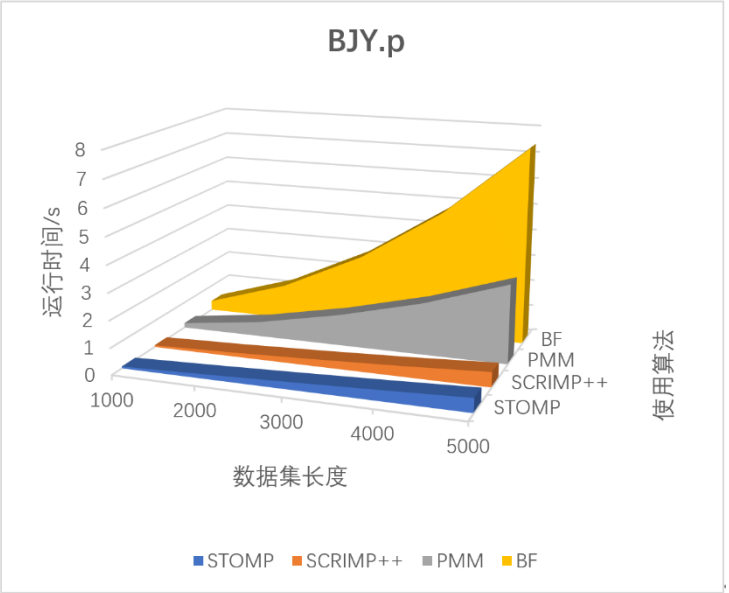
(b)HRS 数据集 nh3n 指标数据



(c)HRS 数据集 dox 指标数据



(d)HRS 数据集 tp 指标数据



(e)BJY 数据集 p 指标数据

图 3.8 各数据集拓展性实验结果

结果如上图 3.8 所示，在不同数据集下，各个算法的运行时间均随着数据集长度的增加而增长。

其中，即使使用了 MASS2.0 优化的 BF 算法还是存在增速随着数据集的不断增加而明显增长的问题，其暴力遍历两两计算相似度的方法造成了这种问题；相比之下，STOMP 算法和 SCRIMP++算法增速基本持平，且增幅均十分微小，在 5 个步长的数据集长度增量上时间只增加了 1s 左右，可见其效率。STOMP 与 SCRIMP++算法均对 Distance Profile 的计算过程进行了大幅优化，并返回最小相似度向量 Matrix Profile。

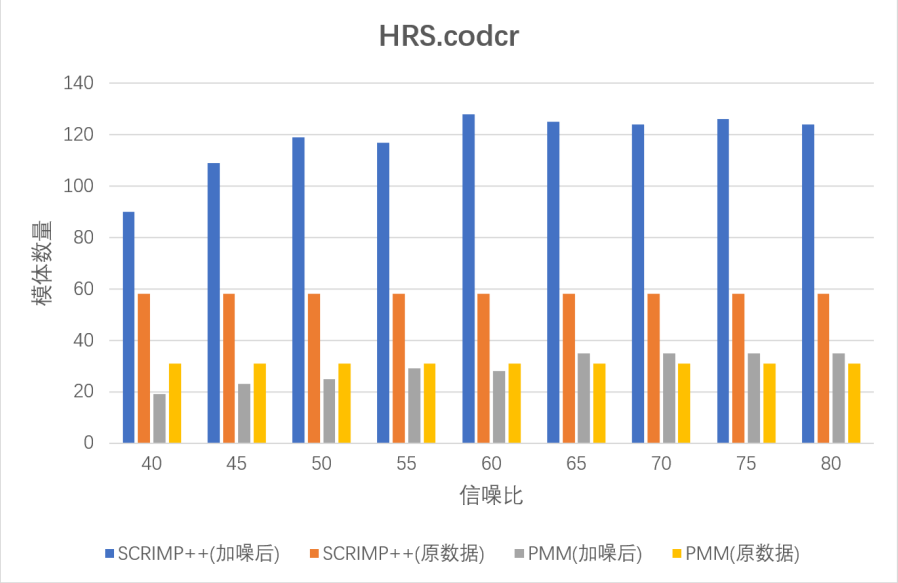
相比之下，本文提出的方法 PMM 算法存在增速均大于 STOMP 与 SCRIMP++的问题，其原因在于，PMM 虽然是在 STOMP 算法的基础上改进的算法，但是其 STOMP 算法核心返回的是一个保存了所有相似度的双层字典元组，且增加了支持度计算和后续去冗优化两步筛选程序，这就意味着至少在原 STOMP 算法的基础上增加了 $O((n-m+1)^2)$的复杂度，是牺牲速度换取模体精准度的做法。

由此可见，BF 算法的暴力匹配和相似度计算过程都存在明显复杂的缺点，相比其他算法完全不适合使用。而 STOMP 算法和 SCRIMP++算法虽然存在挖掘时间短的优势，但其挖掘出的模体中存在冗余导致精准度不够，且返回的 matrix profile 只有最相似的一个子序列，缺少支持度，不具典型性。而 PMM 算法虽然牺牲了一些速度，但却为关联规则挖掘提供了高丰富度的数据结构 P-Matrix

Profile，给出了精准的模体集合。因此，本文提出的方法兼具速度和模体两者的优势平衡，后续关联规则的挖掘实验也选用该方法进行模体挖掘。

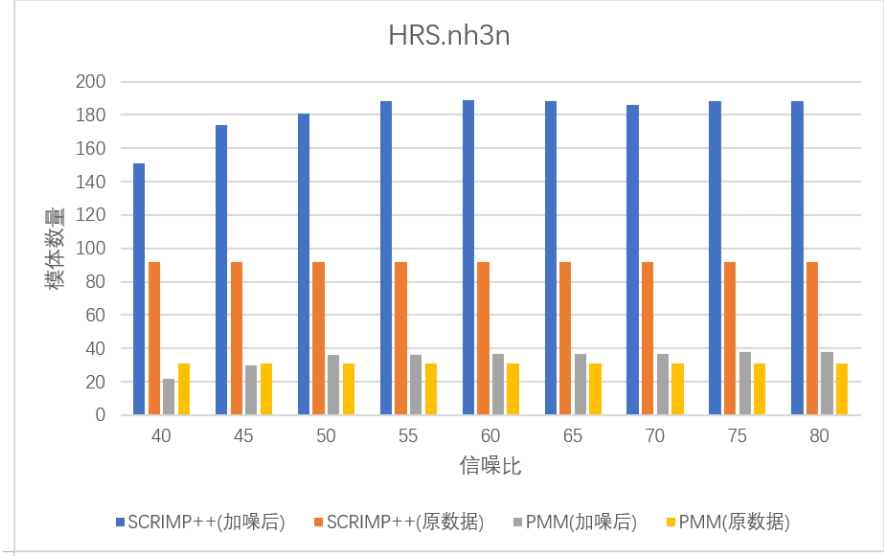
实验 4：测试本章方法鲁棒性

为验证算法的鲁棒性，本文使用 Python 编写了 wgn (White Gaussian Noise) 加噪函数对上述数据集添加高斯白噪声。控制不同程度的信噪比条件，基于原始数据挖掘得到的模体数量，对比分析本章方法和当前研究最新的经典算法 Scrimp++算法加入噪声前后的变化情况。实验选择与准确性分析过程相同的参数设置，控制信噪比范围在 40dB 到 80dB，每次变化 5dB。不同数据集上加噪前后各算法发现模体数量对比分析实验结果对比如图 3.9 各数据集鲁棒性实验结果

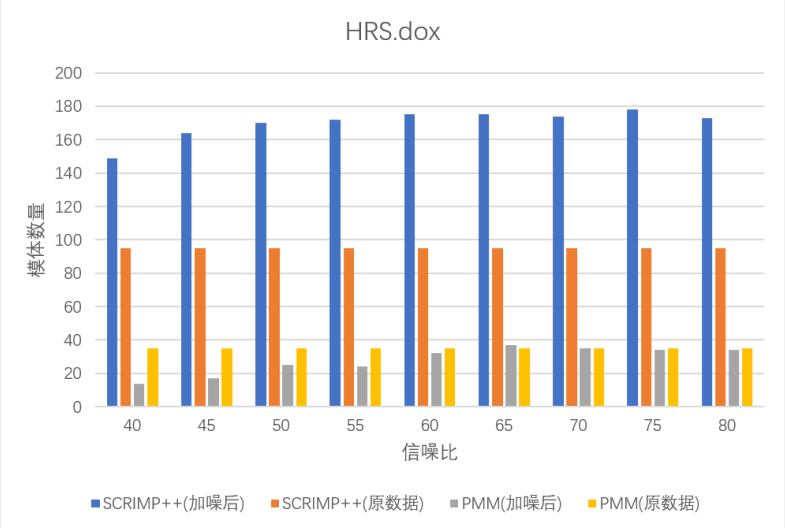


9 中子图(a)~(e)所示。

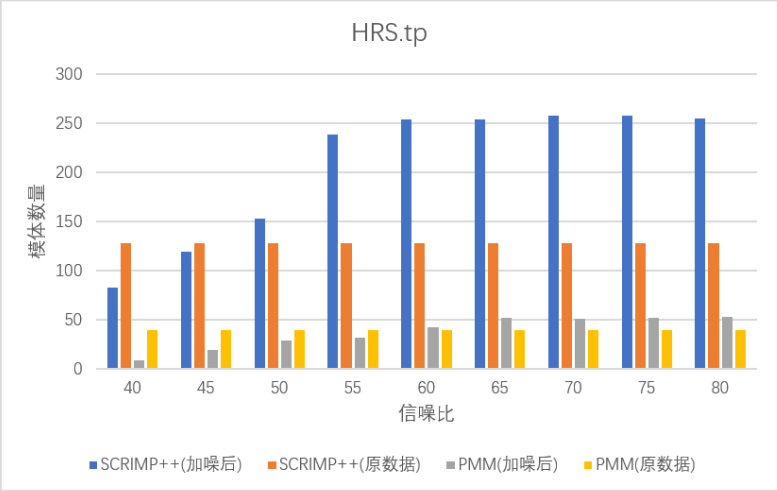
(a)HRS 数据集 codcr 指标



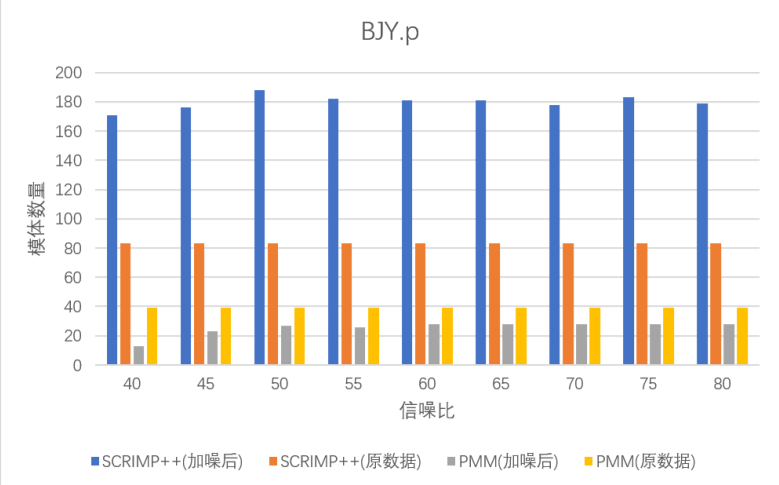
(b) HRS 数据集 nh3n 指标



(c) HRS 数据集的 dox 指标



(d)HRS 数据集 tp 指标



(e) BJY 数据集 p 指标

图 3.9 各数据集鲁棒性实验结果

分析图表实验结果可知，在加入一定噪声数据后，本文方法和 SCRIMP++算法发现模体数量由信噪比的变化受到不同程度的影响。对于 5 个数据集 HRS.codcr, HRS.nh3n, HRS.dox, HRS.tp, BJY.p,可以看到 SCRIMP++处理加入噪声的数据后，模体数量发生了巨大的变化，将近增加了一倍，说明噪声的存在对于其算法的影响十分巨大，结果挖掘出的模体也因为噪声而发生了变化，这是其鲁棒性不佳的体现。相比之下，新提出的算法 PMM 在数据中加入噪声后挖掘出的模体数量和原数据挖掘的结果相差不大，基本保持在 80%以上，除了在 40 左右低信噪比，也就是噪声较多的情况下可能存在差距略大的情况。

由此可见，在噪声水平较高的情况下，本文方法挖掘出的模体数量虽然发生了一定的变化，但仍能保持不失真的准确率；在噪声水平较低的情况下，本文方法的准确率基本不会受到影响，**进一步说明了本文方法在噪声数据情况下表现出较强鲁棒性和稳定性**。

综合上述，本章提出的 PMM 模体挖掘算法具有良好的性能。

3.5 本章小结

本章提出一种基于双阈值和去冗优化的时间序列模体挖掘方法。为挖掘得到具有实际意义的模体，首先利用 MASS 优化算法和 STOMP 算法计算相似度矩阵；然后设置相似度阈值 threshold 和支持度阈值 limit 对子序列进行筛选，得到模体构建 P-Matrix Profile1.0；最后使用去冗优化提出 PMP1.0 中的冗余模体，二次使用支持度阈值筛选模体，得到最终的模体元组 PMP2.0。本章采用多个深圳市河道站点收集的时间序列数据集进行实验，结果表明本章所提方法具有较好的性能，能够挖掘出更加准确合理的模体。

第四章 多元时间序列时态关联规则应用

本章首先介绍本文应用研究的内容与背景，其次叙述对于深圳河道数据集进行污染模式的挖掘出的研究流程和结果，包括问题分析、数据集介绍，数据预处理，模体挖掘，规则挖掘和结果分析六部分。

4.1 概述

时间序列数据广泛存在于自然、医学、社会、工业等各个领域中，挖掘多元时间序列中蕴藏着的自然演变规律和人类活动影响的信息，从而确定变量之间的关联关系，对理解事件演化过程并做出科学决策具有重要意义。

在水利领域中，城市里河道的污染情况影响着城市形象、居民生活以及生态自然环境。由于深圳的城市化程度不断提高，其河道的降雨溢流污染风险也随之提高。防止河道水质恶化是水务治理决策的目标，而决策的关键在于分析水质污染模式的成因，只有确定了污染模式与哪些因素有关，确定了水质污染变化发生的过程机理，才能对症下药。

因此，针对深圳河道污染物指标的时间序列数据集进行关联规则挖掘，对预测深圳市周边河道未来发生污染的时间周期和污染程度有十分巨大的帮助，通过挖掘关联规则也可以为将来的污染防治，从降雨和水中化合物的变化中得出未来一定时间周期内的污染模式预测。

本章所涉及的污染模式挖掘部分，主要由时间序列模体挖掘和关联规则挖掘两部分构成，在第二章和第三章均介绍过。对深圳市五个水质站和五个雨量站提供的水质和雨量时间数据集及逆行多元时间序列的关联规则挖掘。对挖掘出的关联规则进行研究分析，得出一些深圳市降雨溢流因素影响的污染模式的周期特征，为科学防治河道污染提供决策支持，保障河道环保安全。

4.2 基于模体的多元时间序列时态关联规则挖掘方法应用

4.2.1 问题分析

河道中的水质污染风险情况由多种因素共同影响和决定。首先是水中某些化合物的含量超标会导致水质污染，通过地表水环境质量标准基本项目标准限值[25]可通过多项水中的物质含量将水质从好到坏分为 I 类、II 类、III 类、IV 类、V 类和超 V 类水。其次，降雨也可能导致水质发生变化，产生溢流和污染等多种河道风险。

通过检测水质化合物和雨量随着时间变化的周期型变化并进行分析，就能够得到本地水质和雨量的时间序列数据集，从而分析当下的情况，为防治河道的污染风险做出贡献。

从时间序列中提取含有规律的信息，有以下几个技术难点。首先是如何从多维的、大量的时间序列数据中提取其特征，因为阈值的存在，反复调试的需要决定了提取数据特征的算法必须不仅具备有效性，同时也需要具备一定的高效性，模体挖掘可以有效挖掘出时间序列中具备典型性的特征数据。其次，是噪声数据的干扰，水质站和雨量站在采集数据的同时很可能存在被外界多种因素干扰导致所记录的数据与实际存在误差的情况，得到存在噪声的数据混杂在数据集中，使用算法的鲁棒性便显得十分重要了。

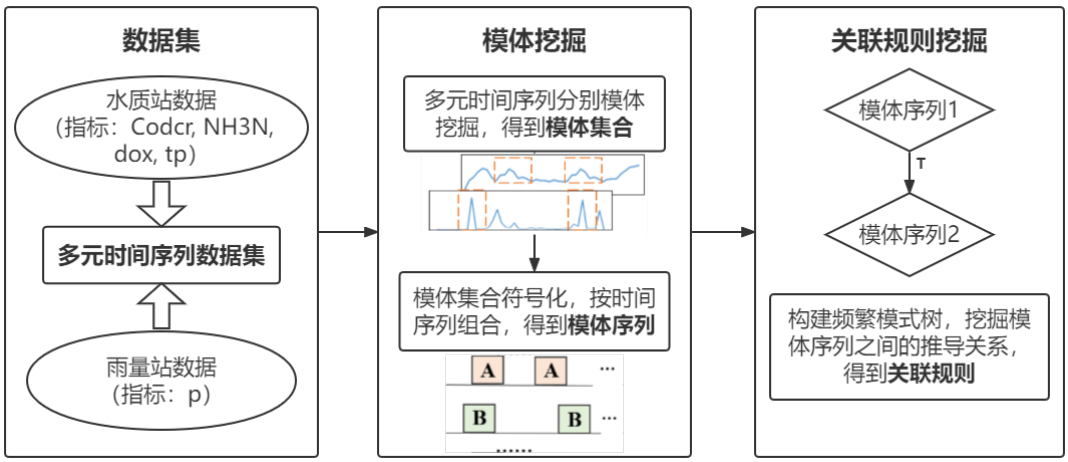


图 4.1 深圳河道数据集关联规则挖掘过程示意图

因此，将有效性、高效性、鲁棒性等多项需求纳入研究范围，最终本章选择使用第二和第三章所用的规则挖掘和模体挖掘算法完成对深圳市河道水质和雨量数据集的模式挖掘分析研究，具体流程如上图 4.1 所示。

第一步是对数据集进行预处理，将水质站和雨量站的原始数据处理后得到多元时间序列数据集；第二步，对多元时间序列数据集中的所有序列各自进行模体挖掘，得到每条序列的模体集合，再对集合进行符号化，得到模体组成的符号序列；最后，采用第二章末尾提到的关联规则挖掘方法对模体序列进行规则挖掘，得到多元时间序列规则，经过筛选，得到最终的挖掘结果，将结果结合深圳市河道的实际情况进行数据分析。

4.2.2 数据集

本章实验使用的数据集来自深圳市智慧水务综合监测管理平台，数据集包括五个水质站和五个雨量站的水质和雨量检测信息，时间跨度为从 2020 年 10 月 1 日起，至 2021 年 12 月 31 日，每小时记录一次指标数据，数据集中存在缺失的情况，以空白表示。

本次数据集提供方主要分为水质站和雨量站两类，两类数据集所提供的数据指标也有所不同。

其中，水质站的数据由蔡屋水质站、大小坑水质站、汇入口水质站、泥岗桥水质站、粤宝路水质站共五个站点提供，水质站的数据包含五个指标：

1）CODcr, 即化学需氧量[25]，是以化学方法测量水样中需要被氧化的还原性物质的量。废水、废水处理厂出水和受污染的水中，能被强氧化剂氧化的物质（一般为有机物）的氧当量。在河流污染和工业废水性质的研究以及废水处理厂的运行管理中，它是一个重要的而且能较快测定的有机物污染参数。水中含量大于 40mg/L 为污染程度严重的超 V 类水；

2）NH_3N, 即氨氮[25]，非离子氨是引起水生生物毒害的主要因子，可导致水富营养化现象产生，是水体中的主要耗氧污染物。而氨离子相对基本无毒。水中含量大于 2.0mg/L 为污染程度严重的超 V 类水；

3）DOX，即溶解氧[25]，由于水被污染，有机腐败物质和其他还原性物质的存在，溶解氧就被消耗，所以越干净的水，所含溶解氧越多；水污染越厉害，溶解氧就越少。水中含量小于 2mg/L 为污染程度严重的超 V 类水；

4）tp，即总磷[25]，总磷是水样经消解后将各种形态的磷转变成正磷酸盐后测定的结果，以每升水样含磷毫克数计量。水中含量大于 0.4mg/L 为污染程度严重的超 V 类水。

雨量站的数据由布吉雨量站、布吉河口雨量站、草埔雨量站、鹿丹村雨量站、笋岗闸上雨量站共五个站点提供。雨量站的数据只有雨量 p 一个指标。

指标的名称和参数变量名具体如表 4.1 所示。

表 4.1 数据集参数说明

序号	变量名称	描述	单位
1	CODcr	化学需氧量	毫克每升（mg/L）
2	NH_3N	氨氮	毫克每升（mg/L）
3	DOX	溶解氧	毫克每升（mg/L）
4	tp	总磷	毫克每升（mg/L）
5	p	雨量	毫米（mm）

下**错误!未找到引用源。**所示为数据集中各指标选取长度为 500 的时间序列

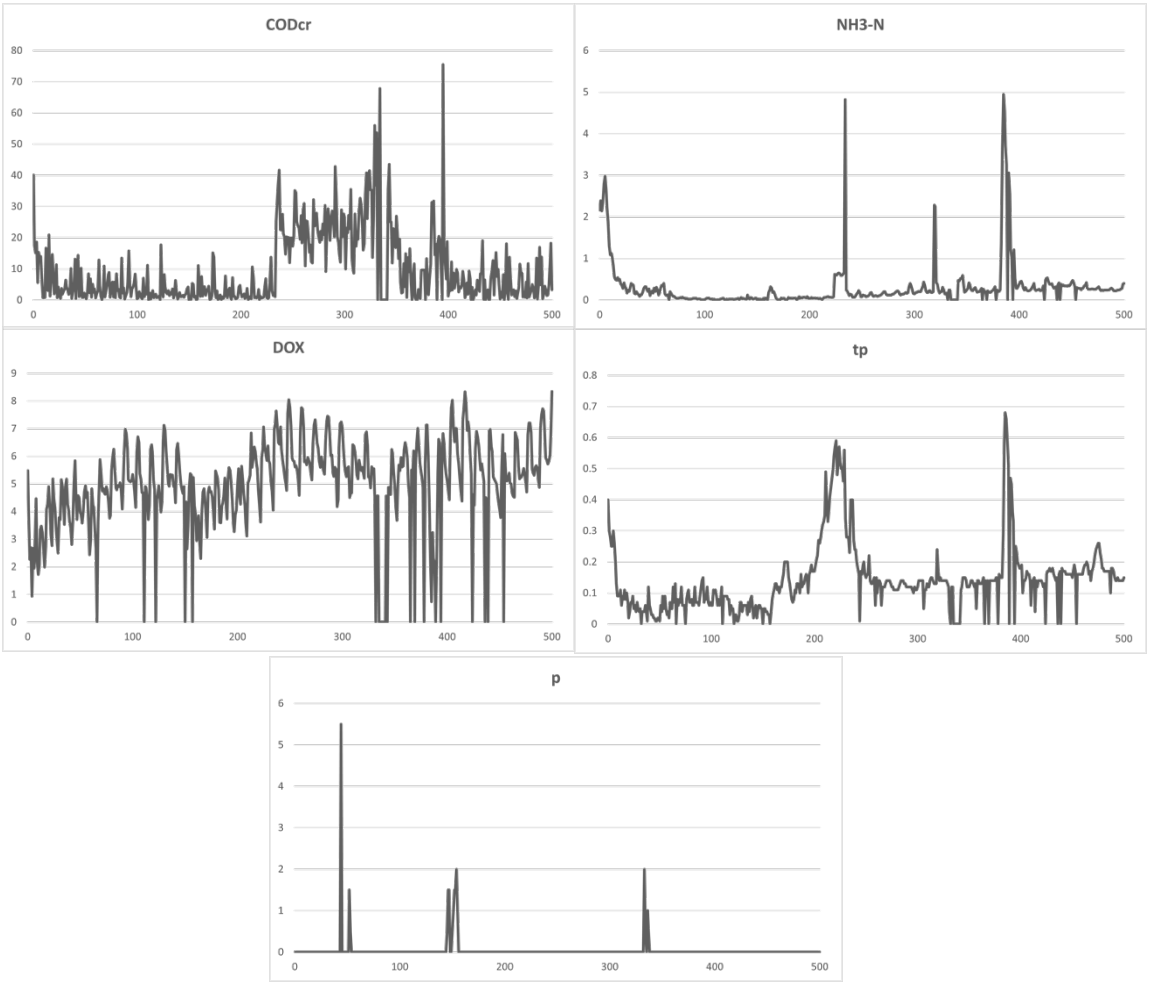


图 4.2 部分数据展示图

数据展示图。

4.2.3 数据预处理

原始数据来自上述的 10 个监测站点，但其中存在数值缺失的情况，原数据集中以相应日期时间后的空白来表示，以此与零值区分开。因此，在进行模体挖掘和后续其他工作之前，有必要对原始数据进行数据预处理。包括缺失值的处理，并对 1 小时间隔记录的数据进行处理，精简数据规模，以便进行后续的研究。

首先是缺失值的处理。我们可以选择直接整条剔除缺失的条目，但是这样可能破坏数据之间的连贯性，作为时间序列数据集，时间上的连贯性无疑是十分重要的，直接删除数据等同于直接去掉了时间线上的一个个时间节点，有可能影响模体挖掘的结果的准确性。此类情况下可以采用插补法填补缺失值，维持时间的

连贯性，也不破坏数据原本的趋势走向。基本的插补法主要分为单一插补和多重插补两种方法：单一插补[25]是指采用一定方式,对每个由于无回答造成的缺失值只构造一个合理的替代值，并将其插补到原缺失数据的位置上，替代后构造出一个完整的数据集；多重插补[25]是由哈佛大学的 Rubin 教授在 1977 年首先提出的，该方法是从单一插补的基础上衍生而来的。指给每个缺失值都构造 m 个替代值(m>1)，从而产生了 m 个完全数据集，然后对每个完全数据集采用相同的数据分析方法进行处理，得到 m 个处理结果，然后综合这些处理结果，基于某种原则，得到最终的目标变量的估计。

除了上述两种基本的插补法之外，近年来兴起了基于机器学习的数值插补方法。主要分为 K 最临近(K-Nearest Neighbors，KNN)[25]、聚类算法[25]和神经网络填补[25]三类。本文的数据集中的缺失值较少，所以选择采用 KNN 算法进行填补。

解决了缺失值的问题后，该数据集仍需要进行精简处理。原始数据包含了各个站点从 2020 年 10 月 1 日至 2021 年 12 月 31 日每小时记录一次的水质/雨量数据，数据条目大约在 10000 条左右，不管是画图还是后续模体挖掘，一旦时间序列的跨度一大，数据图像的趋势就不明显，因此有必要对其进行精简，使用 KNN 算法填补的同时，也要将时间间隔从 1 小时提高到 2 小时，最终在填补和精简的数据处理完成后，各站的数据长度均被统一为 5484，解决了各个数据集长度不一，时间不存在对应关系，以及存在缺失值的问题。

最后，在训练和作图时，也有必要给各个数据集标注一个较为简短的代号，这里选择用三个字母，即数据集原名的前两个首字母加上水质站的‘S’或是雨量站的‘Y’加起来作为各个数据集的代号。

经过数据预处理之后的十个数据集的具体说明见下错误!未找到引用源。。下一步就是对该多元时间序列数据集进行模体挖掘，再通过挖掘关联规则，得到规则序列，分析深圳市河道各参数间的关系，为未来的河道降雨溢流风险防治提供决策支持。

表 4.2 数据集详细信息说明

序号	数据集名称	长度	简述
1	蔡屋水质站数据集	5484	CWS，该水量站 2020.10.01~2021.12.31 的每两个小时的水质指标数据，包括 codcr,nh3n,dox,tp 四项指标。
2	大小坑水质站数据集	5484	DXS，该水量站 2020.10.01~2021.12.31 的每两个小时的水质指标数据，包括 codcr,nh3n,dox,tp 四项指标。

3	汇入口水质站数据集	5484	HRS，该水量站 2020.10.01~2021.12.31 的每两个小时的水质指标数据，包括 codcr,nh3n,dox,tp 四项指标。
4	泥岗桥水质站数据	5484	NGS，该水量站 2020.10.01~2021.12.31 的每两个小时的水质指标数据，包括 codcr,nh3n,dox,tp 四项指标。
5	粤宝路桥水质站数据集	5484	YBS，该水量站 2020.10.01~2021.12.31 的每两个小时的水质指标数据，包括 codcr,nh3n,dox,tp 四项指标。
6	布吉雨量站数据集	5484	BJY，该雨量站 2020.10.01~2021.12.31 的每两个小时的雨量数据，包括雨量 p 一项指标。
7.	布吉河口雨量站数据集	5484	BHY，该雨量站 2020.10.01~2021.12.31 的每两个小时的雨量数据，包括雨量 p 一项指标。
8.	草埔雨量站数据集	5484	CPY，该雨量站 2020.10.01~2021.12.31 的每两个小时的雨量数据，包括雨量 p 一项指标。
9.	鹿丹村雨量站数据集	5484	LDY，该雨量站 2020.10.01~2021.12.31 的每两个小时的雨量数据，包括雨量 p 一项指标。
10.	笋岗闸上雨量站数据集	5484	LGY，该雨量站 2020.10.01~2021.12.31 的每两个小时的雨量数据，包括雨量 p 一项指标。

4.2.4 模体挖掘

本章采用第三章所提出的基于双阈值的去冗优化模体挖掘算法，即 PMM (P – Matrix Profile Motif Mining Algorithm)算法。首先要设置模体挖掘过程中的一些必要参数，包括滑动窗口长度，相似度阈值，支持度阈值。

因为本文选择 1 天作为研究对象，研究 24h 作为周期的水质变化模式，因此选用 12 作为窗口长度，滑动一次给出一个 24h 内的时间子序列数据。为了将模体挖掘的数量控制在每个数据集 5 条之内，设置的相似度阈值和支持度阈值如表 4.3 和 4.4 所示。

表 4.3 各数据集相似度阈值说明

序号	数据集名称	*R*(CODcr)	*R*(NH3N)	*R*(DOX)	*R*(tp)	*R*(p)
1	CWS	0.970	0.750	0.970	0.840	/
2	DXS	0.975	0.560	1.000	0.520	/
3	HRS	1.100	0.700	1.110	0.850	/
4	NGS	0.880	0.800	0.820	1.050	/
5	YBS	0.780	0.900	0.980	0.850	/
6	BJY	/	/	/	/	0.750
7	BHY	/	/	/	/	0.750
8	CPY	/	/	/	/	0.750
9	LDY	/	/	/	/	0.750
10	LGY	/	/	/	/	0.800

表 4.4 各数据集支持度阈值说明

序号	数据集名称	*R*(CODcr)	*R*(NH3N)	*R*(DOX)	*R*(tp)	*R*(p)

1	CWS	20	20	20	20	/
2	DXS	20	20	20	20	/
3	HRS	20	20	20	20	/
4	NGS	20	20	20	20	/
5	YBS	20	20	20	20	/
6	BJY	/	/	/	/	80
7	BHY	/	/	/	/	80
8	CPY	/	/	/	/	80
9	LDY	/	/	/	/	80
10	LGY	/	/	/	/	80

根据以上设置的阈值进行模体挖掘，因篇幅所限，本章仅展示部分挖掘结果的详细信息并对其进行绘图以便观察和结果分析的进行。其中，水质站挖掘出的模体数据信息如表 4.5 所示，其模体形态通过 python 绘制成图，具体如图 4.3 水质站数据集的部分模体结果

所示；而雨量站挖掘出的模体数据信息如表 4.6 所示，其模体形态具体如图 4.4 所示。

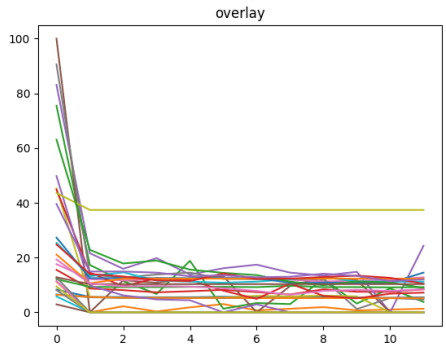
表 4.5 水质站数据集模体挖掘部分结果展示

序号	变量名称（数据集. 指标）	模体数量	部分模体信息（窗口索引：子序列索引）
1	CWS.codcr	4	**1813**：（109、394、948、1018、1180、1459、1547、1580、1589、1657、1698、1813、2066、2317、2367、2373、2740、2755、2922、3091、3172、3307、3523、3822、3878、4254、4573、4768、4828） **2759**：（1168、1611、2589、2710、2759、2829、2843、2911、2949、3080、3633、3663、3799、3825、3923、3957、4174、4280、4771、4846、5347）
2	DXS.nh3n	3	**3319**：（77、2373、2550、2803、3113、3221、3319、3352、3441、3638、3670、3808、3944、3980、4015、4040、4100、4136、4148、4305、4528、4662、4720、4857、4877、5238） **3983**：（68、2794、3104、3310、3343、3432、3629、3661、3752、3799、3899、3935、3971、3983、4115、4139、4290、4336、4431、4868、5229）
3	HRS.dox	3	**1337**：（980、1268、1337、1393、1397、1493、1529、1570、1636、2013、2410、2428、2471、2486、2514、2599、2620、2643、2767、2812、

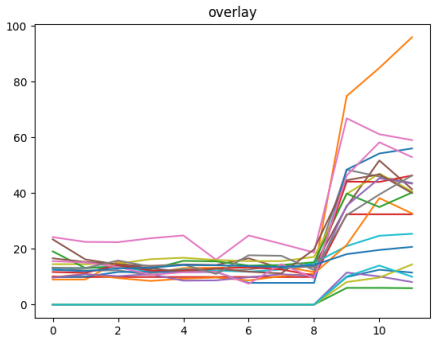
			2825、3125、4469、4505、4681、4926、5074、5272、5432） **5063**：（969、1398、1518、1579、1749、2459、2473、2562、2609、2632、2698、2778、2814、2829、3922、4663、4910、4962、5063、5251、5297、5420）
4	YBS.tp	4	**2557**：（68、102、399、400、440、477、478、724、808、1096、2158、2557、2657、2820、3221、3492、3496、3554、4022、4168） **4245**：（773、916、1029、1232、1384、1559、1597、1654、1807、1821、2050、2126、2459、2594、2699、2962、3259、3272、3384、3657、4245）

表 4.6 雨量站数据集模体挖掘部分结果展示

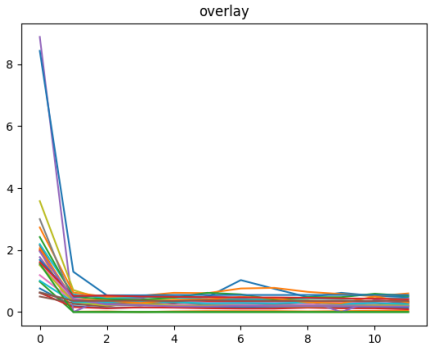
序号	变量名称（数据集. 指标）	模体数量	部分模体信息（窗口索引：子序列索引）
1	BJY.p	3	**2591**：（51、154、336、639、1592、1790、1834、1855、1880、2086、2282、2502、2519、2572、2591、2694、2788、2801、2825、2837、2851、2852、2936、2956、3018、3068、3102、3121、3171、3198、3218、3245、3259、3329、3364、3413、3440、3464、3523、3571、3583、3608、3653、3727、3751、3771、3833、3870、3872、3930、3941、3968、3989、4062、4169、4182、4209、4255、4267、4298、4386、4387、4406、4491、4536、4540、4559、4838、4981、5006、5009、5356、5389、5426）
2	CPY.p	3	**3682**：（32、40、133、323、627、1575、1773、1823、1840、1864、1869、2075、2141、2268、2362、2487、2505、2553、2578、2671、2683、2777、2825、2838、2839、2902、2944、3006、3029、3091、3110、3161、3207、3220、3247、3317、3331、3402、3429、3451、3475、3540、3572、3587、3626、3657、3682、3710、3729、3760、3794、3796、3822、3848、3919、3953、3978、3999、4026、4051、4075、4158、4171、4187、4243、4256、4276、4375、4395、4450、4453、4513、4516、4826、4971、4995、5294、5337）



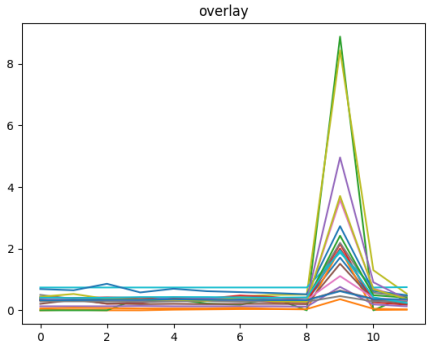
(a)CWS.codcr 模体结果 1813



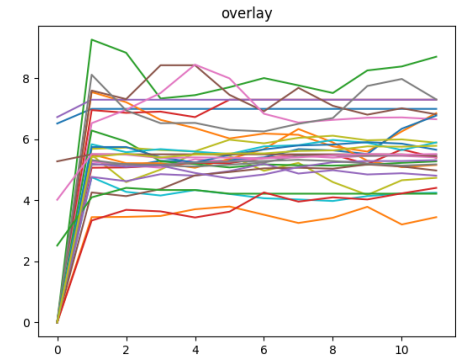
(b) CWS.codcr 模体结果 2759



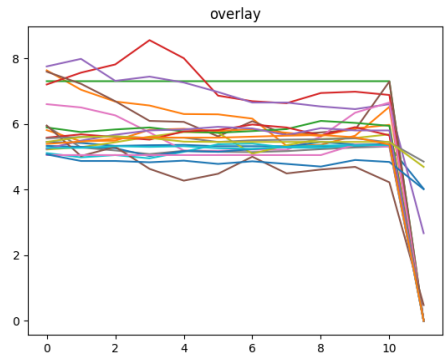
(c) DXS.nh3n 模体结果 3319



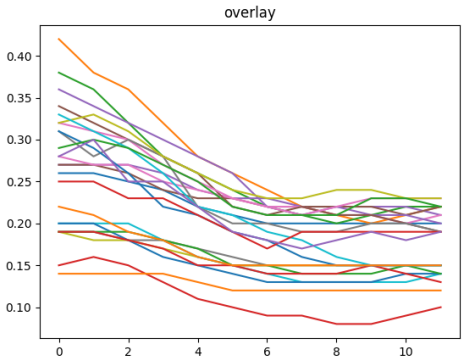
(d) DXS.nh3n 模体结果 3983



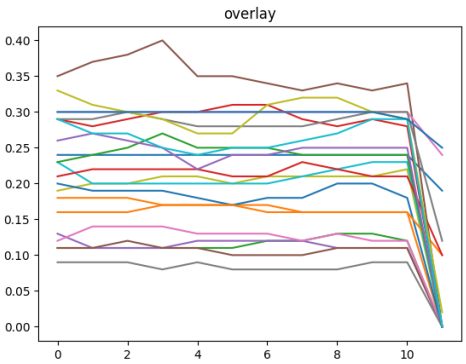
(e) HRS.dox 模体结果 1337



(f) HRS.dox 模体结果 5063

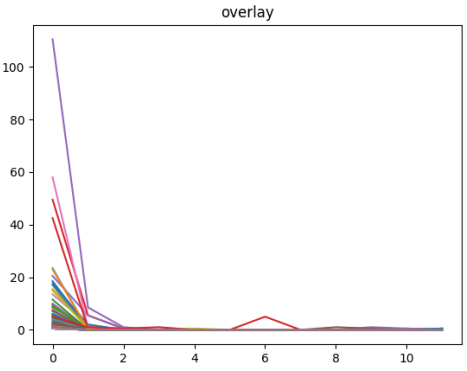


(e) YBS.tp 模体结果 2557

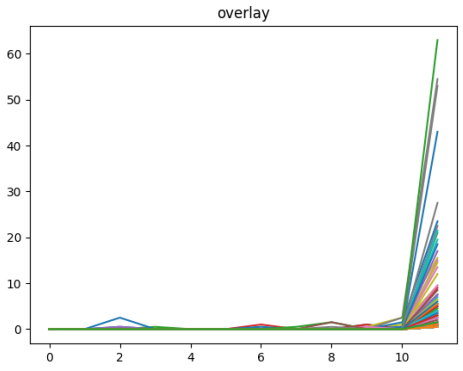


(f) YBS.tp 模体结果 4245

图 4.3 水质站数据集的部分模体结果



(a) BJY.p 模体结果 2591



(b) CPY.p 模体结果 3682

图 4.4 雨量站数据集的部分模体结果

表 4.4 各数据集支持度阈值说明

序号	数据集名称	*R*(CODcr)	*R*(NH3N)	*R*(DOX)	*R*(tp)	*R*(p)
1	CWS	20	20	20	20	/
2	DXS	20	20	20	20	/
3	HRS	20	20	20	20	/
4	NGS	20	20	20	20	/
5	YBS	20	20	20	20	/

6	BJY	/	/	/	/	80
7	BHY	/	/	/	/	80
8	CPY	/	/	/	/	80
9	LDY	/	/	/	/	80
10	LGY	/	/	/	/	80

根据以上设置的阈值进行模体挖掘，因篇幅所限，本章仅展示部分挖掘结果的详细信息并对其进行绘图以便观察和结果分析的进行。其中，水质站挖掘出的模体数据信息如表 4.5 所示，其模体形态通过 python 绘制成图，具体如图 4.3 水质站数据集的部分模体结果

所示；而雨量站挖掘出的模体数据信息如表 4.6 所示，其模体形态具体如图 4.4 所示。

与**错误!未找到引用源。**中的模体信息首项表示各个数据集通过模体挖掘得到模体的索引，即模体窗口初始处在该时间序列数据集中的位置索引；冒号后是所有符合相似度阈值的子序列索引。所有上述表格中展示的挖掘出的模体以及未展示出的模体结果均为对应数据集中从索引开始，且长度为m的子序列，此处 m 为上文参数设置处提到的窗口长度 12。

可以从挖掘结果的表 4.5 和表 4.6 中看到使用 PMM 算法挖掘得到的 P-Matrix Profile 时间序列模体不仅包含模体本身的起始索引，还包含所有符合相似度阈值的子序列索引，这一点也提供了其支持度大小，三点合一，具备信息上的丰富性；而通过图 4.4 和图 4.5 中可以通过观察就发现其挖掘出的模体在时间的行进上表现出明显的规律性，在 12 的窗口长度，即 24h 的时间周期上均存在明显的相似发展规律，因此其模体算法给出的结果也具备准确性和周期性。

通过使用第三章的模体挖掘算法对十个数据集进行处理，得到挖掘的模体集合后，我们采用第二章结尾提到的关联规则挖掘算法对这些模体集合进行分析，以此挖掘出深圳市不同地点的水质指标和雨量指标之间存在的时间序列上的关联规则。因为本文数据集长度均为 5484，未达到大数据级别的时间开销，也对精准度和效率有一定要求，因此本章选用 FP-growth 算法对数据进行关联规则挖掘。

通过 FP-growth 得到关联规则挖掘的结果，得到了十个数据集的模体组合成的序列之间的时间推导关系，通过结果分析来确认 CODcr、NH3N、dox、tp、p 这些参数之间的周期性关联规则。具体过程首先将每个数据集中的模体进行符号化，将整个时间序列转化为模式序列。将该模式序列作为关联规则挖掘的输入，设置相应的最小支持度 R_s 和最小置信度阈值 R_c，本文将 R_s 设置为 30，R_c

设置为0.5。继续设置$min_overlap = 0.5$，$min_T = 12$，$max_T = 24$，即$12 \leq T \leq 24$，进行规则挖掘，得到规则数据集。

挖掘结果中，满足最小支持度和置信度阈值的规则共有367条。由于篇幅限制，本章仅在此展示筛选过后的36条规则，如表4.7所示。表中规则推导的时间跨度均为滑动窗口长度m，即T = 24小时。

表 4.7 关联规则挖掘结果部分规则展示

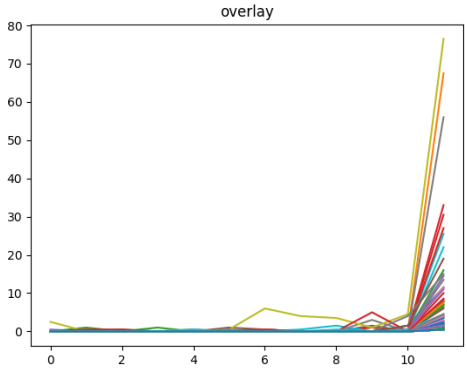
序号	规则	置信度
1	['YBSdox_b', 'YBSnh3n_a']=>['CWScodcr_a']	1.000
2	['SGYp_a', 'NGSnh3n_a']=>['CPYp_a']	1.000
3	['SGYp_a', 'NGSnh3n_a', 'BHYp_a']=>['BJYp_a', 'CPYp_a']	1.000
4	['BJYp_a', 'NGSnh3n_a', 'BHYp_a']=>['SGYp_a', 'CPYp_a']	1.000
5	['BHYp_a', 'NGSnh3n_a', 'CPYp_a']=>['SGYp_a', 'BJYp_a']	1.000
6	['LDYp_b', 'SGYp_b']=>['BHYp_b']	1.000
7	['LDYp_a', 'YBSnh3n_a']=>['BHYp_a']	1.000
8	['YBSnh3n_a', 'BJYp_a']=>['CPYp_a']	1.000
9	['BHYp_b', 'CWScodcr_a']=>['LDYp_b']	1.000
10	['CPYp_a', 'BHYp_a', 'NGSnh3n_a']=>['LDYp_a', 'SGYp_a']	1.000
11	['SGYp_a', 'BHYp_a', 'CPYp_a']=>['LDYp_a']	1.000
12	['BJYp_a', 'LDYp_a', 'CPYp_a']=>['BHYp_a']	0.991
13	['YBSdox_b', 'CWScodcr_a']=>['YBSnh3n_a']	0.909
14	['LDYp_a', 'BHYp_a']=>['BJYp_a']	0.905
15	['YBSnh3n_a', 'CWSnh3n_a']=>['CWScodcr_a']	0.867
16	['LDYp_a']=>['BJYp_a']	0.841
17	['HRScodcr_a', 'CWSnh3n_a']=>['HRSnh3n_a']	0.833
18	['BJYp_a', 'CPYp_a']=>['LDYp_a', 'SGYp_a']	0.832
19	['HRSnh3n_a', 'HRStp_c']=>['HRScodcr_a']	0.778

20	['BJYp_a']=>['SGYp_a', 'CPYp_a']	0.770
21	['HRSnh3n_a', 'DXSnh3n_a']=>['YBSnh3n_a']	0.769
22	['SGYp_b']=>['LDYp_b']	0.711
23	['DXSnh3n_a', 'CWScodcr_a']=>['YBSnh3n_a']	0.706
24	['HRScodcr_a', 'HRStp_c']=>['HRSnh3n_a']	0.700
25	['HRSnh3n_a', 'YBSnh3n_a']=>['CWScodcr_a']	0.688
26	['HRStp_c']=>['HRScodcr_a']	0.667
27	['YBScodcr_a', 'CWScodcr_a']=>['YBSnh3n_a']	0.667
28	['YBScodcr_a', 'CWScodcr_a']=>['CWStp_a']	0.667
29	['NGStp_b']=>['NGSdox_b']	0.643
30	['HRSnh3n_a', 'YBSnh3n_a']=>['DXSnh3n_a']	0.625

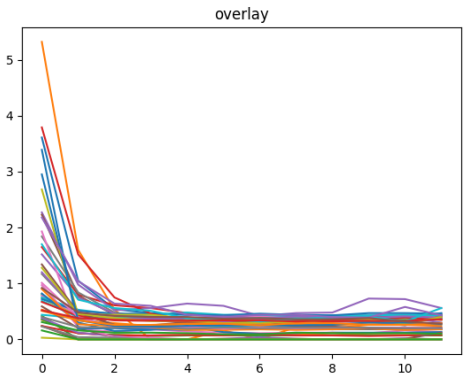
河道中的雨量和水质指标之间都是相互影响的。通过宏观的分析，我们可以从规则挖掘的结果中发现，各个雨量站所在地点的雨量数据的变化，更容易对其他地点的雨量和水质数据的变化产生推动作用，这也就是规则结果中雨量指标大量出现在前件中的体现；而雨量的变化很难被水质的变化影响，除了 NGSnh3n_a 这个模体的发生可能会对周围的雨量产生影响外，其他水质的指标变化对雨量不经常产生影响，这也符合我们对气象知识的基本认识；而水质中的各项指标，如 CODcr、NH3N、dox 等，则更容易对本站中的其他水质污染物的指标产生影响，也更容易对其他水质站点所在地的水质产生连锁效应。上述是对结果进行的宏观分析，可以发现，十分契合气象和水质的基本发展规律，该结果具有与事实相符的客观性，可以进行进一步的具体分析，得到具体的趋势之间的时序关系，为河道污染防治提出具体决策支持。

下面选取表中部分规则进行解释说明。

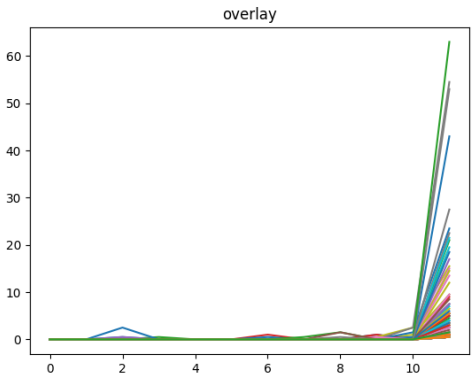
错误!未找到引用源。**规则 4：**['SGYp_a', 'NGSnh3n_a']=>['CPYp_a']为强关联规则，反映了在 24 小时内，如果出现笋岗闸上雨量站检测到周期内第 20~24 小时内雨量陡增，出现如图 4.5 – (a)所示的规律，且泥岗桥水质站检测到周期内 0~2 小时内水中氨氮剧减，出现如图 4.5 – (b)所示的规律，以上两者同时出现时，那么在 24 小时后，周期内第 20~24 小时期间草埔雨量站周围出现降雨量大幅增长的可能性接近 100%，该预测结果如图 4.5 – (c)所示。



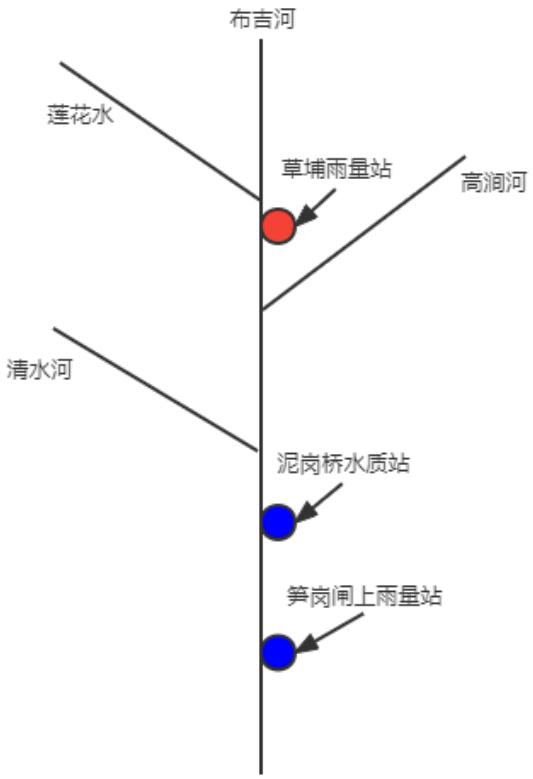
(a) SGYp_a 模体（前件）



(b) NGSnh3n_a 模体（前件）



(c) CPYp_a 模体（后件）

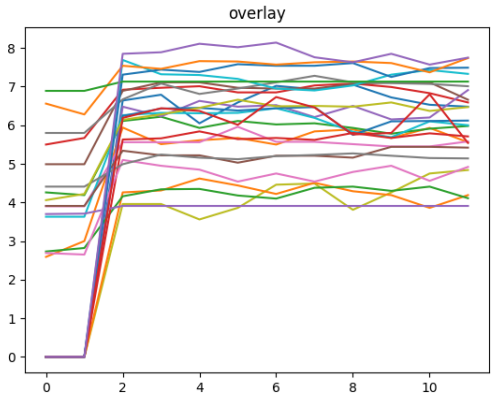


(d) 所涉监测站地理位置关系

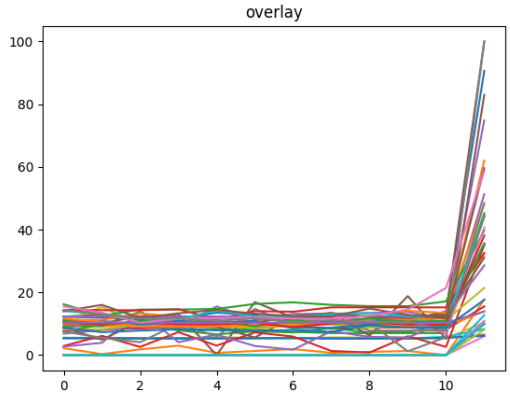
图 4.5 规则 4 前后件详情趋势示意图

为了更好明确该规则的实际含义，其中前后件所涉及水质站/雨量站的地理位置关系如上图 4.5 – (d)所示，其中前件站点的地理位置用**蓝色**表明，后件用**红色**表明。

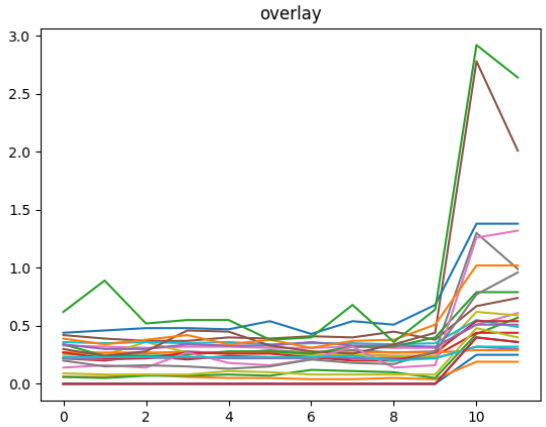
错误!未找到引用源。**规则 13：**['YBSdox_b', 'CWScodcr_a']=>['YBSnh3n_a']为强关联规则，反映了在 24 小时内，如果粤宝路桥水质站检测到周期内第 2~4 小时内水中 dox 指标剧增，出现如图 4.6 – (a)所示的规律，且蔡屋围泵站检测到周期内 20~24 小时内水中氨氮剧减，出现如图 4.6 – (b)所示的规律，以上两者同时出现时，那么 24 小时后，在粤宝路桥水质站周期内第 18~20 小时，该站监测地区出现降雨量大幅增长的可能性约为 90.9%，该现象如图 4.6 – (c)所示。



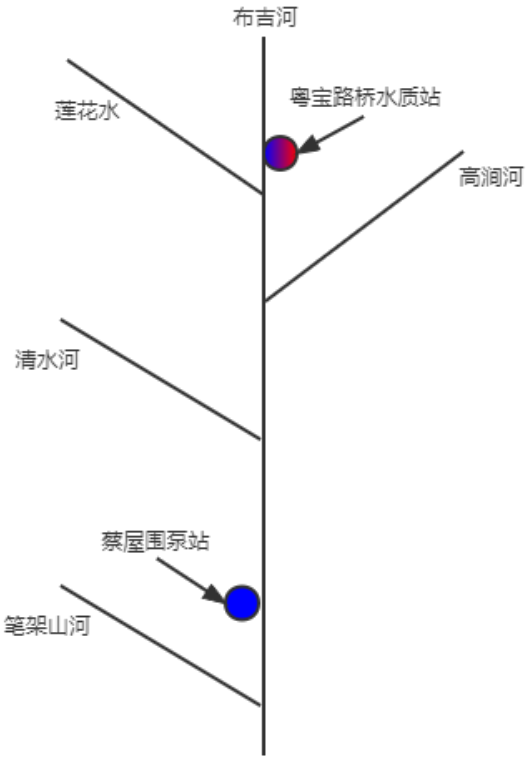
(a) YBSdox_b 模体（前件）



(b) CWScodcr_a 模体（前件）



(c) YBSnh3n_a 模体（后件）



(d) 所涉监测站地理位置关系

图 4.6 规则 13 前后件趋势示意图

为了更好明确该规则的实际含义，其中前后件所涉及水质站/雨量站的地理位置关系如上图 4.6 – (d)所示，其中前件站点的地理位置用**蓝色**表明，后件站点用**红色**表明，若既为前件又为后件，用**红蓝渐变**表明。

综上所述，上述关联规则能够体现降雨量与 CODcr、NH3N、DOX、tp 等多个指标变化的复合过程，与实际气象变化规律相符。本文多元时间序列时态关联规则挖掘方法能够充分考虑滞后关系，时态关联规则的应用具有实际意义；并且第三章模体挖掘结果是精确结果，能够保存时间序列变化趋势和其他序列间影响的信息，能够进一步解释分析规则，为有效推进深圳市河道污染综合防治工作提供参考。

4.3 本章小结

本章将第二章所介绍的相似度计算和关联规则挖掘与第三章所介绍的模体挖掘方法相结合，应用到实际场景，通过对深圳市河道的水质/雨量数据集进行

模式挖掘和结果性分析，得到了多元时间序列上的关联规则集合。本章首先阐述了研究对象，即深圳市河道数据集的基本情况，并对其进行了标注和整理；其次，对数据集进行了预处理操作，使其在时间上进行了对其操作并使其符合模体挖掘的标准；而后，对水质和雨量的时间序列数据集各自进行了模体挖掘，得到了模体集合；最后，对模体集合进行符号化得到模体序列，对该序列进行关联规则挖掘，得到规则序列，对得到的序列结果进行结果性分析，绘制图表，得到应用上的决策建议，为深圳市河道的降雨溢流风险和水质污染的预测提供了周期性的预测建议，通过对污染模式分析提出了有价值的防治规律性措施，具有一定意义。

第五章 总结与展望

5.1 全文总结

在各个领域高速发展，计算机技术推动了科技浪潮前进的如今，海量的时间序列数据不断积累，等待其中的规则被挖掘，内化进入行业，再推动领域的发展。这些未经处理和挖掘的原始时间序列数据集应当如何利用，模式挖掘和相关的挖掘和分析算法给出了很好的答案。

通过对时间序列数据集进行预处理，使其符合模体挖掘的标准后，将多维、多元的处理后时间序列数据组合起来进行模体挖掘，可以得到序列中反复出现，具备特征性的模体集合。再将这些模体集合放在一起进行符号化，得到模体序列，对其进行关联规则挖掘，即可得到规则的集合。对这些规则进行科学的结果分析，即可对领域内周期性的时间进行推动、预防或其他处理，为决策上的调度和制定提供了大量支持。本文就深入研究了模体挖掘方法和关联规则挖掘的实际应用。

（1）对多元时间序列中的相似度计算和关联规则挖掘方法进行了深入研究。从欧几里得距离开始，研究并比较了多种相似度计算方法，并将算法和计算场景相结合，明确了模体挖掘场景下 MASS 和 MASS 改进算法的优势；探讨了关联规则挖掘问题中的一些基本定义，并将两种经典的挖掘算法，即 Apriori 和 FP-growth 算法进行了介绍和性能上的优缺点分析。

（2）深入研究了多元时间序列的模体挖掘方法，提出了一种基于双阈值和去冗优化的多元时间序列模体挖掘方法，并将其与传统的 BF、stomp 等算法进行了性能上的优缺点对比。该方法提出的模体挖掘方法主要分为三步：首先，通过设置滑动窗口计算相似度矩阵；其次，建立提取基于阈值与支持度的 PMP(P-Matrix Profile)向量；最后对 PMP 向量进行进一步的去冗优化，得到具备更高准确性和鲁棒性的 PMP 2.0 向量。也通过实验证明了通过以上步骤计算出模体所具备的优越性。

最终，利用上述所有研究内容和提出的方法，本文将其应用到了深圳市的水质和雨量数据集上，针对深圳市河道的降雨溢流污染模式进行了研究和分析。通过模体挖掘和规则挖掘，对结果进行科学分析，从实际出发发现了切实存在的河

道污染模式，为深圳市河道水质污染和降雨溢流方面的预防提供了一定参考，在未来的预测和防治的决策上提供了支持，保障生产生活的正常秩序，也再次证实了模式挖掘方法对于挖掘时间序列数据潜藏规律的重要意义。

5.2 未来展望

本文提出了一种基于双阈值和去冗优化的多元时间序列模体挖掘方法，提高了多元时间序列数据的模体挖掘准确度，丰富了数据结构，方便关联规则挖掘场景下的后续规则挖掘。然而在利用深圳市水质/雨量时间序列数据集进行实际应用时，本文仍有一些问题有待解决，后续计划集中在以下几个着力点开展相关工作：

1. 本文提出的新模体挖掘算法因为需要拓展数据结构，丰富数据携带量，以便后续规则挖掘，所以在算法的运行时间效率上做了一些妥协，包括在去冗优化的过程中也存在使用简单的循环来处理数据的情况。虽然因为数据密度的增加，无疑在运行效率这方面暂时无法与 stomp 或 scrimp++算法相提并论，但在时间复杂度方面仍有优化空间；

2. 在模体集合进行符号化，将时间序列转化为符号序列这一过程中，因为滑动窗口的起始索引并不一定是窗口长度的整数倍，因此模体符号的起点也不总是与窗口长度重合，而规则挖掘的本质又要求这些符号在时间上必须是等分的，因此只能将非倍数起点的模体符号延长满足倍数要求，在这一过程中定然造成了一些精度的损失，后续考虑如何优化这一处理过程以求降低精度损失。

参考文献

[1] 丁小欧, 于晟健, 王沐贤, 等. 基于相关性分析的工业时序数据异常检测[J]. 软件学报, 2020(3): 726-747.

[2] Funde A, Dhabu M, Paramasivam A, et al. Motif-based association rule mining and clustering technique for determining energy usage patterns for smart meter data[J]. Sustainable Cities and Society, 2019, 46: 101415.

[3] 施明明, 李娜, 胡锦峰. 关联规则在社区居民常见慢性病关联性分析中的应用[J]. 预防医学, 2018, 30(8): 766-770.

[4] Pradhan G, Prabhakaran B. Association Rule Mining in Multiple, Multidimensional Time Series Medical Data[J]. Journal of Healthcare Informatics Research, 2017, 1(1): 92-118.

[5] 张人上, 曲开社. 基于加权关联规则和文本挖掘的金融新闻传播 Agent 实现[J]. 计算机应用与软件, 2015, 32(6): 188-191.

[6] Devarasan E, Prasanna S, et al. Association rule mining using enhanced apriori with modified GA for stock prediction[J]. International Journal of Data Mining Modelling & Management, 2016, 8(2): 195-207.

[7] Rashid A, Nohuddin E, Zainol Z. Association rule mining using time series data for malaysia climate variability prediction[C]//Proceedings of the International Visual Informatics Conference. Springer, Cham, 2017: 120-130.

[8] 喜度, 殷笑茹, 牛霭琛. 基于关联挖掘的自动站数据质控方法的改进[J]. 气象水文海洋仪器, 2018, 35(4): 73-78.

[9] 朱跃龙, 彭力, 李士进, et al. 水文时间序列模体挖掘[J]. 水利学报, 2012(12): 40-48.

[10] 李俊, 苏怀智, 周仁练. 基于关联规则的土石坝渗流推理预测方法及应用[J]. 水利水电科技进展, 2019, 39(5): 89-94.

[11] Han J, Hong C, Dong X, et al. Frequent pattern mining: current status and future directions[J]. Data Mining and Knowledge Discovery, 2007, 15(1): 55-86.

[12] 崔妍, 包志强. 关联规则挖掘综述[J]. 计算机应用研究, 2016, 33(2): 330-334.

[13] 朱旭, 朱晓晓, 王继民. 基于 Matrix Profile 的时间序列变长模体挖掘[J]. 计算机与现代化, 2021 (05): 44.

[14] Lin J, Keogh E, Lonardi S, et al. Finding motifs in time series[C]//Proceedings of the 2nd Workshop on Tempo ral Data Mining, at ACM SIGKDD' 02. 2002: 53-68.

[15] Buza K, Schmidt-Thieme L. Motif-based classification of time series with Bayesian networks and SVMs[M]Ifink A, Lauson B, Seidel W, et al. Advances in Data Analysis, Data Handling and Business Intelligence. Springer Berlin Heidelberg, 2010: 105-114.

[16] TRUONG C D., ANH D T. A novel clustering-based method for time series motif discovery under time warping measure[J]. International Journal of Data Science& Analytics, 2017. DOI: 10.1007/S41060.017-0060-3.

[17] PHU L, ANH D T. Motif-based method for initialization the K-means clustering for time series data[J]. Journal of Computational & Applied Mathematics,2011,236(7): 1733-1742.

[18] Torkamani S, Dicks A, Lohweg V. Anomaly detection on ATMs via time series motif discovery[C] IEEE International Conference on Emerging Technologies and Factory

Automation.2016: 1-8.

[19] 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): 8-20.

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

[21] Shokoohi-Yekta M, Chen Y, Campana B, et al. Discovery of meaningful rules in time series[C]//Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2015: 1085-1094.

[22] 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.

[23] 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.

[24] Qiang Y, Xindong W. 10Challenging problems in data mining research[J]. International Journal of Information Technology & Decision Making, 2006, 5(4): 597-604.

[25] Torkamani S, Lohweg V. Survey on time series motif discovery[J]. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 2017, 7(2): e1199.

[26] Chiu B, Keogh E, Lonardi S. Probabilistic discovery of time series motifs[C]//Proceedings of the 9th ACM SIGKDD international conference on Knowledge discovery and data mining. 2003: 493-498.

[27] Keogh E, Lin J. Clustering of time-series subsequences is meaningless: implications for previous and future research[J]. Knowledge and information systems, 2005, 8(2): 154-177.

[28] 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.

[29] 朱跃龙, 朱晓晓, 王继民. 基于子序列全连接和最大团的时间序列模体发现算法[J]. 计算机应用, 2019, 39(02): 110-116.

[30] Mueen A, Keogh E, Zhu Q, et al. Exact discovery of time series motif[C]//Proceedings of the SIAM international conference on data mining. 2009: 473–484.

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

[32] Jiawei H, 范明. 数据挖掘概念与技术, 机械工业出版社, 2001, 8(1), 149-150.

[33] Agrawal R, Srikant R. Fast algorithms for mining association rules[C]//Proc of International Conference on Very Large Databases. 1994: 487-499.

[34] Park J S, Chen M S, Yu P S. An effective hash-based algorithm for mining association rules[J]. SIGMOD Record,1995,25(2):175-186.

[35] Jiawei H, Jian P, Yiwen Y. Mining frequent patterns without candidate generation[C]//Proc of ACM SIGMOD International Conference on Management of Data. New York: ACM Press, 2000: 1-12.

[36] Agrawal R, Shafer J. Parallel mining of association rules[J]IEEE Trans on Knowledge and Data Engineering, 1996, 8(6): 962-969.

[37] Giannella C, Han Jiawei, Pei Jian, et al. Mining frequent pattens in data streams at multiple time granularities[J]. Next Generation Data Mining, 206, 35(1): 61-84.

[38] Yun C, Haixun W, Yu P S, et al. Moment: maintaining closed frequent itemsets over a stream

sliding window[C]//Proc of the 4th International Conference on Data Mining. 2004: 59-66.

[39] Inokuchi A, Washio T, Motoda H, An Apriori-based algorithm for mining frequent substructure from graph data[C]//Proc of the European Symposium on the Principle of Data Mining and Knowledge Discovery. 2000: 13-23.

[40] Kuramochi M, Karypis G. Frequent subgraph discovery[C]//Proc of the 1st IEEE International Conference on Data Mining. 2001: 313-320.

[41] Xifeng Y, Jiawei H. Span: graph-based substructure pattern mining[C]//Proc of the 2nd IEEE International Conference on Data Mining. 2002: 721-724.

[42] Agrawal R, Srikant R. Mining sequential patterns[C]//Proc of the 11th International Conference on Data Engineering. 1995: 3-14.

[43] Srikant R, Agrawal R. Mining sequential patterns: generalizations and performance improvements[C]//Proc of the 5th International Conference on Extending Data Base Technology. 1996: 3-17.

[44] 崔妍, 包志强. 关联规则挖掘综述[J]. 计算机应用研究, 2016, 33(2): 330-334.

[45] 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 IEEE 16th International Conference on Data Mining. 2016: 1317-1322.

[46] Kalakrishnan M, Chitta S, Theodorou E, et al. STOMP: Stochastic trajectory optimization for motion planning[C]//2011 IEEE international conference on robotics and automation. IEEE, 2011: 4569-4574.

[47] Yan Z , Yeh C , Zimmerman Z , et al. Matrix Profile XI: SCRIMP++++: Time Series Motif Discovery at Interactive Speeds[C]//Proceedings of the IEEE International Conference on Data Mining. 2018: 837-846.

[48] 夏青. 中国地表水环境质量标准修订研究[J]. 环境监测管理与技术, 1998, 10(2): 7-11.

[49] 河海大学《水利大辞典》编辑修订委员会编. 水利大辞典：上海辞书出版社，2015

[50] 许国强, 曾光明, 殷志伟,等. 氨氮废水处理技术现状及发展[J]. 湖南有色金属, 2002, 18(2):5.

[51] 陈佳.溶解氧测定仪与碘量法测定溶解氧的结果比较[J].教育教学论坛,2013(32):238-240.

[52] 雷立改, 马晓珍, 魏福祥, 等. 水中总氮, 总磷测定方法的研究进展[J]. 河北工业科技, 2011, 28(1): 72-76.

[53] Schafer J L. Multiple imputation: a primer[J]. Statistical methods in medical research, 1999, 8(1): 3-15.

[54] Schafer J L. Multiple imputation: a primer[J]. Statistical methods in medical research, 1999, 8(1): 3-15.

[55] Crookston N L, Finley A O. yaImpute: an R package for kNN imputation[J]. Journal of Statistical Software, 2008, 23: 1-16.

[56] Zhang S, Zhang J, Zhu X, et al. Missing value imputation based on data clustering[M]// Transactions on computational science I. Springer, Berlin, Heidelberg, 2008: 128-138.

[57] Nordbotten S. Neural network imputation applied to the Norwegian 1990 population census data[J]. JOURNAL OF OFFICIAL STATISTICS-STOCKHOLM-, 1996, 12: 385-402.

致谢

我在河海大学的四年时光匆匆走过，本科的学习生涯即将在此画上句号，在这段旅途迎来结束之际，我想在这里感谢所有在这四年求学之中给予我关怀，授予我知识，慷慨无私分享善意的人们，我的老师、同学、家人和朋友们。没有这些人在我的求学之路上熠熠生辉，我的大学时光将会显得如此黯淡。

首先，我想感谢在这过程中给予我学业上指导的所有老师。特别是我的班导师和项目导师陆佳民副教授，感谢陆老师在大学四年作为班导师给我和我班同学的谆谆教诲和学业上的鼎力相助。上课时，陆老师缜密的思维逻辑和幽默风趣的讲解让我和同学们都受益良多；在学术上，在老师指导下完成的知识图谱项目也让我从老师身上看到了严谨的学术作风和深厚的知识储备。冯钧教授也作为项目导师在学术研究方面给了我巨大的帮助，在如何将知识转化为应用的经验这一方面，对我帮助良多。

其次，我要感谢师兄和同学们的帮助和支持。感谢滕志新师兄在我开展毕业设计过程中给我一对一的指导，提供了研究思路的同时也耐心地帮我纠正了许多问题，是我今后学习的榜样。感谢吴锬、胡腾波、范斌三位同学在大学四年的陪伴和帮助，他们让我的大学生活充满了阳光与欢乐。

最后，我想感谢我的家人们，是他们在我背后支撑着我前行，他们是我努力的原因，也是我前行的动力。他们的坚持，在我身后永远是我最坚强的后盾，让我更有勇气。

感谢河海大学让我看到不一样的天空，让我们相遇，让我们学习，感谢这求学之旅中的所有人，和带给我美好与宝贵的回忆。

作者：陈易轩

2022 年 5 月于南京