**审定成绩：**

|  |  |  |  |
| --- | --- | --- | --- |
| **重庆邮电大学**  **毕业设计（论文）** | | | |
|  | | | |
|  | **中文题目** | **关联分类规则挖掘软件设计与实现** |  |
|  |  |
| **英文题目** | **Design and implementation of association classification rule mining software** |
|  |  |
| **学院名称** | **计算机科学与技术学院/人工智能学院** |
| **学生姓名** | **文锐锴** |
| **专 业** | **智能科学与技术** |
| **班 级** | **04052001** |
| **学 号** | **2020211608** |
| **指导教师** | **周应华 副教授** |
| **答 辩 组**  **负 责 人** | **姓名 职称** |
|  | |

**2024 年 5 月**

**重庆邮电大学教务处制**

**计算机科学与技术/人工智能**学院本科毕业设计(论文)诚信承诺书

本人郑重承诺：

我向学院呈交的论文《 关联分类规则挖掘软件设计与实现 》，是本人在指导教师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不含任何其他个人或集体已经发表或撰写过的作品成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明并致谢。本人完全意识到本声明的法律结果由本人承担。

年级 2020

专业 智能科学与技术

班级 04052001

承诺人签名 文锐锴

年 月 日

学位论文版权使用授权书

本人完全了解重庆邮电大学有权保留、使用学位论文纸质版和电子版的规定，即学校有权向国家有关部门或机构送交论文，允许论文被查阅和借阅等。本人授权重庆邮电大学可以公布本学位论文的全部或部分内容，可编入有关数据库或信息系统进行检索、分析或评价，可以采用影印、缩印、扫描或拷贝等复制手段保存、汇编本学位论文。

（注：保密的学位论文在解密后适用本授权书。）

|  |  |
| --- | --- |
| 学生签名： | 指导老师签名： |
| 日期： 年 月 日 | 日期： 年 月 日 |

摘要

排版参照此格式。

传统的关联规则挖掘算法意在从海量数据中挖掘出有意义的规则，以描述数据之间存在的相互联系。而分类问题则是利用分类器，根据样本的属性对样本类别进行预测。对于前者，所能发现的规则是无法事先知晓的；对于后者，目标类别已经事先划定并且最终的判定结果只能有一个。因而合理地整合两种方法，利用样本属性的内在联系，构建出与样本类别间的映射关系，就可以得到一个基于特殊的关联规则实现的分类器。这种特殊的关联规则也称为类关联规则，包含了样本属性及其对应的类别。

本文首先选取并介绍了CBA-M1算法、CBA-M2算法、APR算法和CMAR算法的基本原理，然后通过需求分析、软件设计来分析软件基本功能，并基于Vue3和Flask框架的前后端分离架构进行软件开发，算法主体使用python语言实现。在系统设计方面，本文采用了软件工程原理，构建了一个模块化的系统架构，包括数据预处理、类关联规则提取和分类器构造等关键模块。此外，实现的软件简便易用，能够实现多种算法的分类结果以及不同算法的对比分析。

实验部分，使用UCI公开数据集对软件进行了测试，并对不同算法的挖掘结果和效率进行了对比分析。结果表明，所开发的软件在准确性和效率上均表现良好，符合预期效果，能够满足复杂的数据挖掘需求。同时，实验发现创新的剪枝算法与原CBA算法相比，能够在保证准确率的同时，显著地减少类关联规则数目并降低算法运行时间。

**关键词：**关联规则，分类，数据挖掘，CBA算法，APR算法，CMAR算法

**中文关键词中避免包含英文词汇，英文缩写等**

**Abstract**

排版参照此格式。

Traditional association rule mining algorithms aim at mining meaningful rules from massive data to describe the interrelation between data. The classification problem is to use the classifier to predict the sample category according to the attribute of the sample. For the former, the rules that can be discovered cannot be known in advance; For the latter, the target category has been defined in advance and there can be only one final decision. Therefore, a classifier based on special association rules can be obtained by reasonably integrating the two methods and using the intrinsic relation of sample attributes to build a mapping relationship with the sample category. This special association rule, also known as class association rule, contains sample attributes and their corresponding classes.

This paper first selected and introduced the basic principles of CBA-M1 algorithm, CBA-M2 algorithm, APR algorithm and CMAR algorithm, then analyzed the basic functions of the software through demand analysis and software design, and carried out software development based on the front-end separation architecture of Vue3 and Flask framework. The main body of the algorithm was implemented in python language.In the aspect of system design, this paper adopts the principle of software engineering to build a modular system architecture, including key modules such as data preprocessing, class association rule extraction and classifier construction. In addition, the software is simple and easy to use, and can realize the classification results of various algorithms and the comparative analysis of different algorithms.

In the experimental part, the software is tested using UCI public data set, and the mining results and efficiency of different algorithms are compared and analyzed. The results show that the developed software performs well in accuracy and efficiency, meets the expected effect, and can meet the complex data mining requirements. At the same time, the experimental results show that compared with the original CBA algorithm, the innovative pruning algorithm can significantly reduce the number of class association rules and the running time of the algorithm while ensuring the accuracy.

Keywords: Association Rules, Classification, Data mining, CBA algorithm，

APR algorithm，CMAR algorithm

目录

目录一般按三级标题编写，要求标题层次清晰，目录为自动生成。

[第1章 引言 1](#_Toc68176813)

[1.1 研究背景和意义 1](#_Toc68176814)

[1.2 国内外研究现状 1](#_Toc68176815)

[1.2.1 国外研究现状 1](#_Toc68176816)

[1.2.2国内研究现状 2](#_Toc68176817)

[1.3 主要内容和工作安排 2](#_Toc68176818)

[第2章 论文结构及文字格式 3](#_Toc68176819)

[2.1论文结构 3](#_Toc68176820)

[2.2学位论文中的引言 3](#_Toc68176821)

[2.2.1引言的目的 3](#_Toc68176822)

[2.2.2 引言构成及写作要求 4](#_Toc68176823)

[2.3本章小结 4](#_Toc68176824)

[第3章 注释、图表、公式和计量单位格式 5](#_Toc68176825)

[3.1 注释 5](#_Toc68176826)

[3.2图表格式 5](#_Toc68176827)

[3.2.1 图格式 5](#_Toc68176828)

[3.2.2 表格式 8](#_Toc68176829)

[3.3公式格式 10](#_Toc68176830)

[3.4计量单位格式 11](#_Toc68176831)

[3.5本章小结 11](#_Toc68176832)

[第4章 其他格式要求 12](#_Toc68176833)

[4.1 页面要求 12](#_Toc68176834)

[4.2页眉 12](#_Toc68176835)

[4.3页脚 12](#_Toc68176836)

[4.4打印要求 12](#_Toc68176837)

[4.4.1 页面设置 12](#_Toc68176838)

[4.4.2 字体 13](#_Toc68176839)

[4.4.3 字号 13](#_Toc68176840)

[4.5论文查重要求 13](#_Toc68176841)

[4.6论文非学术性错误 14](#_Toc68176842)

[4.7本章小结 14](#_Toc68176843)

[第5章 科学道德与学风 15](#_Toc68176844)

[5.1 科学道德与学风问题 15](#_Toc68176845)

[5.2 科研不端行为 15](#_Toc68176846)

[5.2.1 科研不端行为的定义 15](#_Toc68176847)

[5.2.2 科研不端行为的表现形式 17](#_Toc68176848)

[5.3 科研不当行为 18](#_Toc68176849)

[5.3.1 科研不当行为的定义 18](#_Toc68176850)

[5.3.2科研不当行为的表现形式 18](#_Toc68176851)

[5.4 科研规范 20](#_Toc68176852)

[5.4.1 研究数据收集、记录和保存中的规范 20](#_Toc68176853)

[5.4.2 研究数据使用中的规范 21](#_Toc68176854)

[5.4.3 引文的规范 21](#_Toc68176855)

[5.5 本章小结 22](#_Toc68176856)

[第6章 参考文献的标注和要求 23](#_Toc68176857)

[6.1 参考文献的重要性 23](#_Toc68176858)

[6.2 顺序编码体系 23](#_Toc68176859)

[6.2.1正文中引用文献的标注方法 23](#_Toc68176860)

[6.2.2 文后参考文献表的著录方法 24](#_Toc68176861)

[6.3 参考文献要求 25](#_Toc68176862)

[6.4本章小结 26](#_Toc68176863)

[第7章 总结与展望 27](#_Toc68176864)

[7.1 主要工作与创新点 27](#_Toc68176865)

[7.2 后续研究工作展望 27](#_Toc68176866)

[参考文献 28](#_Toc68176867)

[致谢 30](#_Toc68176868)

[附录A 科技写作中非学术性低级错误的主要表现 31](#_Toc68176869)

# 

# 第1章 引言

每章单独另起一页，**从下一个奇数页开始**。参考此格式进行排版。文中红色字体为重点要求，黑色字体仅为格式参考。

页眉：从目录页开始到论文最后一页，均需设置页眉。页眉内容：偶数页居中对齐为“**重庆邮电大学本科毕业设计（论文）**”，奇数页居中对齐是**各章章名**；字体采用宋体5号。页眉之下有一条下划线。封面、摘要没有页眉，也没有边框。

## 1.1 研究背景和意义

随着全球数据总量的持续攀升，数据分析与挖掘技术在各个领域的应用变得愈发重要。据国家统计局数据显示，截至2023年底，我国数据总量已达到数十ZB级别，并且这一数字预计将在未来几年继续增长。这一数据膨胀趋势使得如何有效地从海量数据中挖掘出有价值的信息，成为了一个亟待解决的社会问题。利用数据挖掘技术提升决策质量和效率，对于国家经济发展和社会进步具有重要意义。

在具体行业中，零售、医疗、金融等领域对数据挖掘技术的需求尤为强烈。根据前瞻产业研究院发布的《中国大数据行业市场前瞻与投资战略规划分析报告》，未来市场规模将突破七万亿元。各行业纷纷加大对数据挖掘技术的投入，以期通过数据分析提升业务洞察力和竞争优势。特别是在金融行业，通过数据挖掘技术进行风险和欺诈检测，已成为行业的常态化需求。因此，对于挖掘出海量数据中存在的潜在模式以及对数据进行分类有着很大的需求。

关联分类规则挖掘结合了关联规则挖掘和分类两种方法的优点，不仅能够发现隐藏在数据中的关联模式，还能通过这些模式对样本进行准确分类。作为数据挖掘领域的一个重要问题，人们对其进行了大量的研究，并衍生出多种不同的关联分类规则算法，其中较为经典的有CBA算法和CMAR算法。CBA算法包含两个部分，第一部分生成一组类关联规则而非传统的关联规则，这一部分实现的关键在于如何筛选出候选项集，以及如何对类关联规则进行规则剪枝。算法的第二部分就是训练分类器，CBA的分类器构造包含M1和M2两种算法。近几年，研究人员在CBA-M1算法的基础上进行改进，提出了基于主动剪枝规则实现的APR算法。CMAR算法不像CBA算法依赖单一的类关联规则，而是利用多条类关联规则的组合进行分类。本论文选取了CBA-M1算法、CBA-M2算法、APR、CMAR算法进行实现，并整合在一个软件中，可以有效处理各类数据集。通过实验验证多种算法在不同数据集上的性能，发现该软件在不同的数据集中均有良好的分类效果。关联分类规则软件的设计与实现，为行业提供了一个强有力的分析工具，有助于提升相关行业的数据处理能力和决策水平。

## 1.2 国内外研究现状

根据自己的课题内容，可设置若干3级目录。3级目录下按照1）→（1）→①…的层次编号。

### 1.2.1 国外研究现状

基于关联规则的分类算法在数据挖掘和机器学习领域中有着广泛的应用，该算法可以有效地提升分类任务的准确性、效率和可靠性。目前国外研究团队在关联分类规则算法的基础上有着多种创新，针对不同领域的具体应用对算法进行了优化。

Shao在2020年提出了一种基于相关性加权类关联规则挖掘（CWCAR）实现的软件缺陷预测模型，该模型利用多加权支持度框架来处理类别之间的不平衡。Ong提出了一种用于癌症生物标志物发现的信息性前k类关联规则（iTCAR）算法，

旨在将微阵列数据与生物信息相结合，生成信息性的类关联规则。此外，Guan提出了基于核均值嵌入提出了一种定量关联分类（QAC）算法，用于评估关联规则的可能性。D'Angelo在2021年开发了一种基于关联规则的恶意软件分类方法，通过使用API调用的常见子序列来识别常见的恶意行为。Abu-Arqoub在2021年引入了一种基于RIPPER算法的新型关联分类算法（ACRIPPER），用于减少挖掘的规则数量并解决类别之间的不平衡问题。

随着技术的进步，新的框架和方法也不断被提出，这些方法专门解决从庞大数据集中选择有效规则的挑战，并将关联规则与其他算法相结合，以适应健康医疗和自然语言处理等领域的特定需求。Malik在2023年开发了一种用于多组学数据挖掘的新框架，解决了从关联规则生成的大量规则中选择有效规则的挑战。同时，Benamira引入了一种新的神经网络模型，该模型与医疗保健决策任务相结合，构建出一种融合了规则可解释性和高性能神经网络的医疗决策模型。在自然语言处理领域，Chi提出了一种零样本文本分类的自训练策略，强调使用标签进行训练以减少模型训练时间。另一方面，Singh在2023年提出了一种用于电子邮件分类规则学习的在线系统（EmFORE），该模型克服了在线环境中优先归纳逻辑编程方法的局限性。

这些研究不仅说明关联分类规则技术在不同领域有着广泛的适用性，同时也表明了这种算法的有效性深受研究人员的欢迎。

### 1.2.2国内研究现状

基于关联规则的分类算法在国内的数据挖掘和机器学习领域也得到了广泛的研究和应用。国内的研究团队在关联分类规则算法的基础上进行了多种创新，特别是针对特定的行业需求和数据特性进行了深入的定制和优化。这些优化措施有效地提高了算法在处理大规模数据集时的性能，同时也增强了算法在复杂数据环境下的稳定性和准确性。

关联分类算法在数据挖掘领域中得到了广泛应用和创新改进。黎文娟提出的基于余弦度量的关联分类算法（IACD）通过使用余弦度量来评估项集与类别之间的相关性，并通过新的规则强度对规则进行排序和剪枝，有效减少冗余规则数量，并在多个UCI数据集上显示出较高的分类准确率。秦晨普等人针对传统关联分类算法的不足，如资源消耗大和规则剪枝难，提出了一种基于分类修剪的改进方案ACCP，通过分块挖掘和改进频繁项集及规则修剪过程，显著提高了分类准确率和运行效率。

李家辉针对支持度置信度框架难以产生大量高质量规则的问题，开发了改进的关联分类算法IAMC，该算法通过多次学习提取更多高质量规则，并引入了综合置信度和补类支持度的新度量标准来提升规则质量，对于未被覆盖的实例则采用决策树方法再次提取规则，以此提高了多个UCI数据集的分类准确率。

郭明怀通过关联挖掘算法对图书馆读者的行为特征进行了分析，有效地划分了读者类型，该方法不仅提高了分析效率和直观性，还减少了计算成本。翟悦等人提出了一种带约束的改进关联分类规则挖掘方法，通过扩展概念格结构和引入差集概念，加速了支持度和置信度的计算，并在时间和空间性能上都表现出色，满足了重用性要求。

这些研究表明，关联分类算法的持续创新不仅解决了现有技术的局限性，而且在多个应用场景中展示了其强大的性能和实用性。

## 1.3 主要内容和工作安排

写全文共分为5章，内容结构安排如下：

第1章为引言，引入课题的研究背景及意义。

第2章是关联规则分类算法，介绍CBA、APR和CMAR算法的实现原理。

第3章是关联规则分类软件设计与实现，介绍软件的设计与功能实现。

第4章为算法比较，借助开发的软件，采用多种数据集对不同的算法进行性能分析。

第5章为未来与展望。

# 第2章 论文结构及文字格式

参考此格式进行排版。标题名称，自己根据情况进行修改。

关联分类规则算法是一种将关联规则挖掘与分类任务相结合的技术。通过在数据集中挖掘频繁项集，得到一种将属性值与特定类标签相关联的特殊规则，称为类关联规则。在分类任务中，这些类关联规则用于预测未知实例的类别，该算法的核心理念就是通过从训练集中找到强相关的规则、构建性能良好的分类器来提高分类的准确性。与其他传统的分类方法相比，关联分类规则算法具有较强的可解释性，能够帮助揭示数据中的潜在模式。

## 2.1CBA算法

CBA算法，即基于关联规则的分类算法（Classification Based on Association），它结合了关联规则挖掘和分类器构建的思想，能够从数据中发现类关联规则并进行分类预测。该算法包含两个部分，第一部分是规则生成，这一步是在Apriori算法的基础上进行修改实现的，其核心思想与Apriori算法相似，不过生成的是针对特定类的关联规则而非传统的关联规则。在这个过程中，首先识别出频繁项集，然后对这些频繁项集进行剪枝，以去除那些不满足最小支持度和最小置信度的规则，从而减少规则的数量并提高规则的质量。第二部分是训练分类器，分类器构造中包含M1和M2两种算法。其中M1算法利用贪心策略，按序逐步抽取规则，与数据集中的剩余样本一一匹配，最后构建一个有序规则集合作为最终的分类器。虽然M1算法形式相对简单，易于编程实现，但是当面对大型数据样本时，其效率并不高，因此又提出了M2算法。M2算法通过更复杂的规则选择和管理机制来优化分类过程，特别适合处理大规模和复杂的数据集。

### 2.1.1 规则生成

CBA算法的规则生成部分就是去寻找所有满足最小支持度和最小置信度约束的规则项集，一个规则项集就是一个由条件集（x）及类标签（y）形成的组合（x->y）。条件集是一组特定的属性，而类标签则是可能的输出分类之一。

类关联规则的支持度，反映了同时满足条件集和特定类标签的数据实例在整个数据集中的占比。而置信度，度量了在满足条件集的数据实例中，同时也符合规则定义的类标签的占比。对于一个类关联规则X ⇒Y，支持度计算公式为：

Support(*X*⇒*Y*) =

其中，count(X∪Y) 表示在数据集中同时包含X和Y的数据项的数量，而𝑁是总的数据项数。这条类关联规则的置信度计算公式为：

Confidence(*X*⇒*Y*) =

其中，count(X∪Y) 同样表示在数据集中同时包含X和Y的数据项的数量，而count(X)是数据集中包含X的数据项数目。

如果一个规则项集满足最小支持度约束minSup，则其是一个频繁规则项集，与之相对应的则是非频繁的规则项集。如果一个规则的置信度满足最小置信度约束，那么它就是一个准确的规则。类关联规则应当都是频繁且准确的规则。

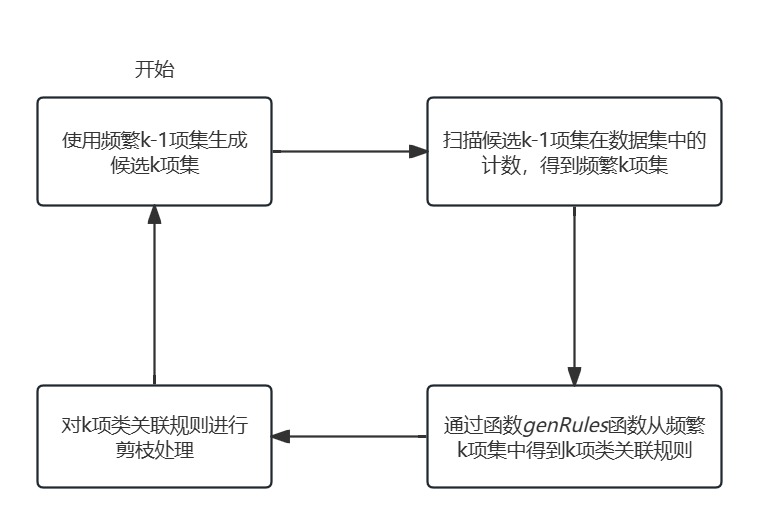


图1 生成类关联的关键步骤

### 2.1.2 分类器构建的CBA-M1算法

CBA-M1算法的主要核心是基于贪心策略，尽可能早的选出高优先级的规则。当存在多种规则时，需要确定某个样本优先使用其中的哪一个规则,故需要对生成的类关联规则进行优先级排序，排序规则如下：

给定两个规则，规则ri优先级高于规则rj的条件是：

（1）ri的置信度大于rj的置信度。

（2）ri和rj置信度相同，但是ri的支持度大于rj的支持度。

（3）ri和rj的支持度和置信度均相同，但是ri比rj产生的时间要早。

CBA-M1算法的基本思想就是根据生成的类关联规则，构建出一组高优先级有序规则序列

< r1, r2, ··· , rn, default\_class >

作为最后的分类器。序列中的规则按照给定的顺序降序排列，default\_class是默认类，当前面的规则都不能覆盖这条记录时，则将其划分入默认类中。

M1算法首先根据规则序列依次选择规则，并逐一与样本进行匹配。如果样本条件满足当前规则的条件，则说明匹配成功，此时将移除所有符合该规则条件的样本。然后，使用下一条规则继续匹配剩余的数据集。每轮中未匹配成功的样本将被分类为默认类，即剩余数据集中出现次数最多的类标签。在分类器构建过程中，会测试类关联规则集合中的每一条规则，并记录添加每条规则后的累计分类错误数。匹配结束后，将筛选出使得分类错误总数最低的规则，丢弃所有优先级低于此规则的条目，只保留该规则以及优先级高于此规则的规则集合。这些保留的规则构成关联规则分类器，将用于对测试集进行最终的分类。

### 2.1.3 分类器构建的CBA-M2算法

M2算法的中心思想和M1算法基本一致，仍然是先按照上述规则对类关联规则进行优先级排序，找出临界点并丢弃临界点之后的规则，不同的是算法的匹配方法。整个算法包括三个阶段，首先通过缩小候选规则的方法，减少不必要的规则判定，提高训练效率。CBA-M2算法在阶段一中首先对数据集中的样本进行遍历，找出正确分类该样本的最高优先级规则crule，以及错误分类该样本的最高优先级规则wrule。若crule的优先级高于wrule，则该样本被正确分类，并将这条规则添加至指定集合U中。若wrule的优先级高于crule，则需要决定该样本由crule覆盖还是由wrule覆盖，记录后在阶段二处理。阶段二中，遍历集合U，如果某个样本的wrule是其他样本的crule，那么就由wrule覆盖这个样本。否则，从集合U中返回所有错误分类该样本且优先级大于该样本crule的规则。阶段三中，对前两个阶段得到的规则集进行筛选，丢弃那些不会提高分类器准确率的规则。和M1算法类似，以分类错误数最少的规则为临界点，在此规则之后的所有规则都将被丢弃，因为它们只会产生更多的错误。

## 2.2 APR算法

APR作为近几年提出的算法，在分类器构建过程中引入了动态剪枝机制，与CBA-M1算法相比，APR算法能够在分类器构建过程中实时调整规则的优先级和有效性。CBA-M1算法按照既定的规则序列依次匹配样本，完成匹配后会将所有满足条件的样本移除，然而，丢弃的数据实际上会影响其他优先级较低的规则，从而影响分类器性能。而APR算法则在每次规则匹配后，将主动更新剩余规则在未被匹配数据集中的支持度和置信度，对不满足最小支持度和最小置信度的规则进行剔除，减少冗余，确保仅保留最有效的规则，从而提高了分类器的效率和准确性。此外，APR算法通过减少规则之间的重叠和优化规则集合的大小，显著提高了分类过程的效率和准确度，特别适用于处理大规模和复杂的数据集。

## 2.3 CMAR算法

CMAR（Classification based on Multiple Association Rules，基于多重关联规则的分类）在关联分类规则算法的基础上进行改进，旨在克服传统关联规则分类中的过度依赖单个高置信度规则的问题。CMAR算法与传统关联规则分类方法不同，它不依赖单个高置信度的规则，而是从关联规则中挑选出一组高置信度、相关的规则进行分类预测。同时，使用卡方分析方法进行加权，确保选择的多重关联规则集能够为新样本提供更全面的分类决策，减少预测中偏差从而提高分类的准确性。

为了高效地挖掘满足支持度和置信度要求的规则，CMAR使用FP-growth方法的变种结构，通过构建关联FP-tree结构，能够快速地处理大型数据库中的频繁项集挖掘任务。同时，CMAR采用一种新的数据结构CR-tree用于存储和检索已挖掘的规则。该结构通过前缀树的方式来共享规则中的重复项，大大降低了了规则集的检索时间。

对新样本进行分类预测时，CMAR首先从CR-tree中检索出与新样本特征相匹配的规则集合，这些规则按照目标类别进行分组，每个分组对应一个可能的分类标签。然后，CMAR根据每个规则的卡方统计值和加权分析方法计算该分组的综合权重，从而评估分组的整体效果。通过对比不同类别分组的组合效果分数，CMAR选择综合效果最强的组标签作为最终预测的分类标签，从而对新样本进行准确的分类。如果匹配的规则不一致或存在冲突，则通过综合所有规则的效果进行最终决策，以确保分类预测的稳定性和准确性。这种基于多重规则的分类策略有效避免了单一规则引起的偏差与过拟合问题，提高了CMAR模型的预测性能

2.4 本章小结

本章主要介绍了关联分类规则算法，作为一种将关联规则挖掘与分类任务结合的技术，算法的关键在于通过数据集中挖掘频繁项集，生成将属性值与特定类标签相关联的类关联规则。这些规则在分类任务中用于预测未知实例的类别，提高分类的准确性。本章详细探讨了三种主要的关联分类算法：CBA、APR和CMAR算法。

这些关联分类算法各有其优势和应用场景，它们提供了一种相较于传统分类方法更具解释性的方式，帮助揭示数据中的潜在模式，同时提高了分类任务的准确性。

# 第3章 注释、图表、公式和计量单位格式

认真阅读本章内容，图、表、公式等，参考此格式进行排版。

关联分类规则挖掘软件需要实现多种算法对指定数据的关联规则挖掘，软件应具有良好的交互性、直观的界面设计以及实现数据可视化，帮助用户更好地理解和分析结果。

## 3.1 需求分析

关联分类规则挖掘软件需要实现对多种关联分类规则算法的效率和准确率进行比较，支持用户自定义选择数据集，可以自由设定最小支持度和最小置信度参数后，选择其中的某一个关联分类规则算法进行分析，最后得到该算法在当前阈值条件下针对当前数据集生成的类关联规则、准确率、运行时间等结果，使用户能够充分了解该算法在该数据集下的运行效率。为了对比不同算法的性能，需要使用柱形图来展示运行结果，方便用户清晰地看到不同算法的特点及差异。

为了让用户能够自定义选择数据集，需要实现本地文件上传功能，让用户能够上传指定格式的数据文件，并且软件需要识别文件中的数据及其对应的数据格式。考虑到用户可能处理的数据量较大，系统需要有能力处理大规模的数据集，同时支持基本的数据预处理功能，如缺失值处理、异常值检测和数据类型转换。数据预处理功能的实现需要确保数据上传过程简便快捷，尽量减少预处理过程对原数据产生的影响。为了让用户观察到数据变化，软件界面应当能够展示上传后的原数据，以及数据预处理操作后的数据。

由于用户在使用软件时，需要对不同算法以及自身需求设置不同的阈值，所以软件需要提供参数设置部分，允许用户灵活地调整阈值大小。同时，软件页面中应提供一个算法选择区域，实现对不同算法的调用。为了展示算法的运行结果、比较不同算法的准确率和效率，软件应在结果展示页面中展示分类器中的类关联规则并提供多种算法评价指标，帮助用户更好地了解算法的分类效果和性能。

## 3.2总体设计

为了提高软件的可维护性和可扩展性，计划采用前后端分离的软件架构。前端部分负责用户界面展示和交互，后端负责处理数据和算法运算，两者之间通过网络请求实现高效交互，这种模式不仅能提高开发的灵活性，还便于后续的系统维护和升级。

软件的前端部分，着重于提供一个用户友好的操作界面，利用Vue3的响应式编程和组件化的特性，构建了一个模块化和高响应的用户界面。该框架的组合式API使状态管理变得更加直观和灵活，极大地适应了复杂交互界面的需求。前端部分需要实现数据上传区、参数设置选项、算法选择按钮和结果展示区域在内的多个界面组件。界面设计简洁明了，易于操作，同时结合了Element-Plus组件库和ECharts可视化库来实现数据分析结果的动态可视化展示，确保用户能够轻松上传数据，选择合适的挖掘参数和算法，并且能够清晰地查看挖掘结果。

软件的后端设计主要包括三个核心模块：

（1） 数据处理模块，负责接收前端上传的数据文件并对数据进行预处理操作，如数据清洗、离散化和数字编码；

（2） 算法实现模块，根据用户选择的算法和参数设置执行关联规则挖掘，处理用户上传的数据并生成结果；

（3） 网络通信模块，借助flask框架搭建服务端，将算法处理后得到的结果以JSON格式返回前端，供前端展示。

后端中的网络请求部分由Python的Flask框架实现，一是算法使用python代码实现较为方便，二是考虑到Flask框架的轻量级和易于扩展的特性，非常适合用于快速开发数据密集型的Web应用。后端主要负责处理复杂的数据操作和执行关联分类规则算法，包括数据的接收、预处理、算法处理以及结果的存储和网络请求的响应结果返回。

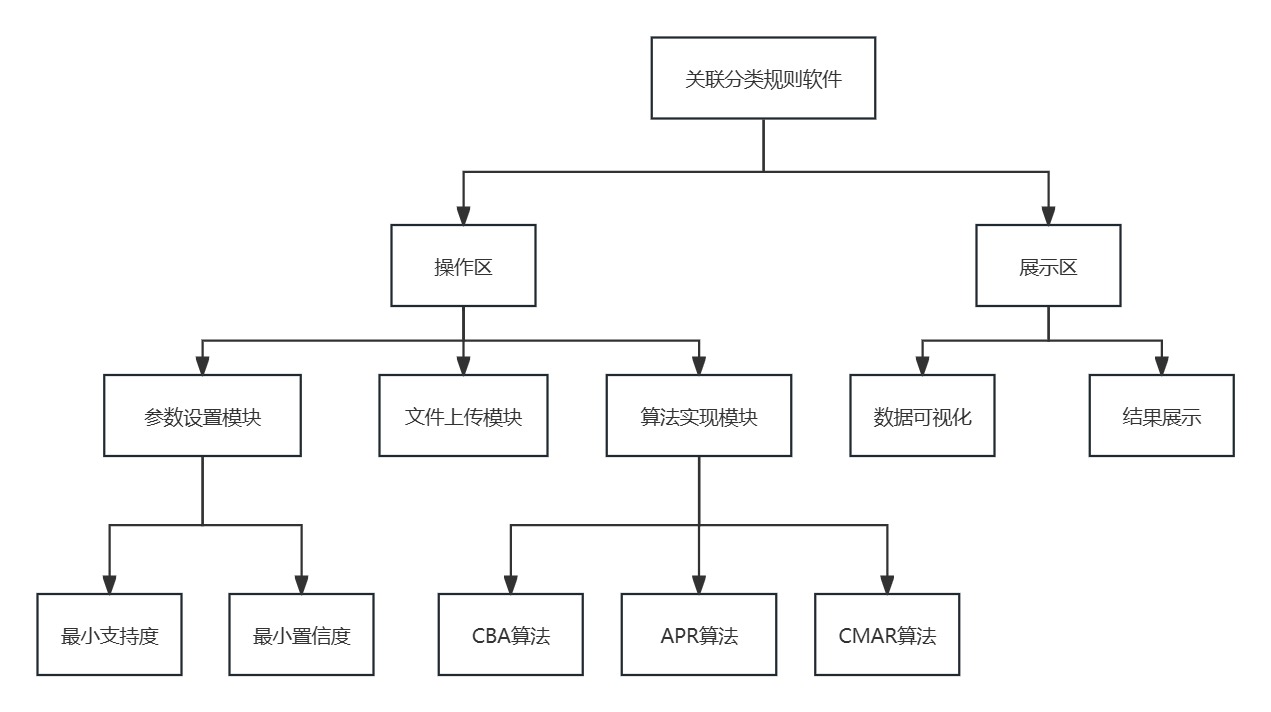


图3.1 素数测试软件层次图

## 3.3 详细设计

### 3.3.1功能设计

软件的主要功能包含：文件上传与数据预览、参数设置、算法实现三个部分。其中结果展示部分包括展示生成的类关联规则以及算法的准确率、时间开销等评价指标。

1）文件上传与数据预览

文件上传部分需要满足用户能够自定义上传本地CSV文件，该部分负责接收用户上传的数据，并借助POST网络请求将数据传送给服务端。同时，将用户上传的数据在客户端页面进行展示，让用户可以在上传后立即查看数据的内容，确保上传的数据符合用户的预期。

2）参数设置

参数设置部分需要满足用户使用鼠标或键盘自由编辑最小支持度和最小置信度，并且阈值大小应精确到小数点后两位数，以满足用户对于这些参数的精细化调整需求。同时，功能实现中需要提供用户友好的界面，方便用户响应式地输入和调整最小支持度和最小置信度的数值，在用户进行修改后，页面需要即时反馈参数设置的变化。

3）算法实现

从实际中采集到的数据往往存在着格式不统一、属性值缺失、噪声点等一系列问题。尽管测试时所使用的数据是从UCI机器学习资源库中下载的，这些数据已经得到了相关人员的一些预处理，但是仍然存在数据缺失以及需要对连续型数据进行离散化处理两个问题。下面就这两个方面进行介绍。

一般来说，对于缺失值，需要分两种情况区考虑。对于缺失值较多的属性，应当毫无保留的删去，因为这样属性的存在将会对最后分类器的性能带来极大影响。对于缺失值较少的属性，则应当考虑保留。算法的预处理过程中，设定的缺失率阈值为0.5，当缺失率小于阈值时，利用缺失值所属特征的众数进行填充；对于缺失率高于阈值时，则丢弃这一列。

预处理过程中有一个问题是对连续型属性值离散化处理。所谓连续型属性值，就是该属性的取值范围并非有限集，与之相对的则是离散型特征，其取值只可能落在一个有限集合中。连续型属性值非常常见，但是在关联分类规则算法中，只能处理离散型的特征。因此需要对连续型属性进行离散化处理，将指定区间内的数值映射为同一类型。实现过程中，采用了基于信息增益的离散化方法，该算法根据数据的混乱程度找出最佳的离散化临界点。

基于递归实现的最小熵划分算法的伪代码如下：

|  |
| --- |
| 算法1 递归的最小熵划分算法 |
| 1. *T* = 2. function Split(S) 3. sort(*S*) 4. *W* = |
| 1. for each candidate *c* in *S* do 2. calculate g(c) = Gain(A, c, S) 3. if (2) is not satisfied then 4. *W* = *W* c 5. end if 6. end for 7. if *W* = then 8. return 9. else 10. return 11. end if 12. end function 13. function Partition(*S*) 14. *t* = Split(*S*) 15. if t = then 16. return 17. else 18. *T* = *T* *t* 19. S1 = 20. S2 = 21. Partition(*S1*) 22. Partition(*S2*) 23. end if 24. end function 25. function main 26. Partition(*S*) 27. sort(*T*) 28. return *T* 29. end function |
|  |

关联分类算法的实现原理在第二章节已经进行过介绍，为了更加准确地描述算法的实现过程，本处给出CBA算法、CMAR算法以及APR算法。

|  |
| --- |
| 算法2 CBA-RG算法 |
| 1. *F1* = {large 1-ruleitems} 2. *CAR1* = genRules(*F1*) 3. *prCAR1* = pruneRules(*CAR1*) 4. k = 2 5. while Fk-1 ≠ do 6. *Ck*= candidateGen(*Fk-1*) 7. for each data case d*D* do 8. *Cd* = ruleSubset(*Ck*, *d*) 9. for each candidate c *Cd*do 10. c.condsupCount++ 11. if d.class = c.class then 12. c.ruleCount++ 13. end if 14. end for 15. end for 16. *Fk* = {*c* *C*k | c.rulesupCount minsup} 17. *CARk* = genRules(*F*k) 18. *prCARk* = pruneRules(*CARk*) 19. *k*++ |
| 1. end while 2. CARs = kCARk 3. prCARS = kprCARk |

|  |
| --- |
| 算法3 CBA-CB M1算法 |
| 1. *R* = sort(*R*) 2. for each rule *r* *R* in sequence do 3. temp = 4. for each case *d* *D* do 5. if d satisfies the conditions of *r* then 6. store *d*.id in *temp* and mark *r* if it correctly classifies *d* 7. end if 8. end for 9. if *r* is marked then 10. insert *r* at the end of *C* 11. delete all the cases with the ids the in *temp* from *D* 12. selecting a default class for the current *C* 13. compute the total number of errors of *C* 14. end if 15. end for |
| 1. Find the first rule *p* in *C* with the lowest total number of errors and drop all the rules after *p* in *C* 2. Add the default class associated with *p* to end of *C*, and return *C* (classifier) |

|  |
| --- |
| 算法4 CBA-CB M2算法: Stage 1 |
| 1. *Q* = ; *U* = ; *A* =; 2. for each case *d* *D* do 3. *cRule* = maxCoverRule(*Cc*, *d*) 4. *wRule* = maxCoverRule(*Cw*, *d*) 5. U = U {cRule} 6. cRule.classCasesCovered[*d*.class]++ 7. if *cRule* > *wRule* then 8. Q = Q {cRule} 9. mark *cRule* 10. else *A* = *A* <*d*.id, *d*.class, *cRule*, *wRule*> 11. end if 12. end for |

|  |
| --- |
| 算法5 CBA-CB M2算法: Stage 2 |
| 1. for each entry < *dID*, *y*, *cRule*,*wRule* > 2. if *wRule* is marked then 3. *cRule*.classCasesCovered[*y*]— 4. *wRule*.classCasesCovered[*y*]++ 5. else 6. *wSet* = allCoverRules(*U*, *dID*.case, *cRule*) 7. for each rule *w wSet* do 8. *w*.replace = *w*.replace {<*cRule*, *dID*, *y*>} 9. *w*.classCasesCovered[*y*]++ 10. end for 11. Q = Q wSet 12. end if 13. end for |

|  |
| --- |
| 算法6 CBA-CB M2算法: Stage3 |
| 1. *classDistr* = compClassDistri(*D*) 2. ruleErrors = 0 3. *Q* = sort(*Q*) 4. for each rule *r* in *Q* in sequence do 5. if *r*.classCasesCovered[r.class] 0 then 6. for each entry <*rule*, *dID*, *y*> in r.replace do 7. if the *dID* case has been covered by a previous *r* then 8. *r*.classCasesCoverred[*y*]— 9. else 10. *rule*.classCasesCovered[*y*]— 11. end if 12. end for 13. ruleErrors = ruleErrors + errorsOfRule(r) 14. classDistr = update(r, classDistr) 15. defaultClass = selectDefault(classDistr) 16. totalErrors = ruleErrors + defaultErrors 17. Insert <*r*, *default-class*,*totalErrors*> at the end of *C* 18. end if 19. end for 20. Find the first rule *p* in *C* with the lowest *totalErrors*, and then discard all the rules after *p* from *C* 21. Add the default class associated with p to end of *C* 22. Return C without totalErrors and default-class |

## 3.4本章小结

本章主要描述了关联分类规则软件的设计过程，该软件在设计上强调直观性、用户友好性，同时注重交互效果和用户体验。在界面布局设计上，采用左侧操作栏和右侧展示栏的简约风格，方便用户能够轻松理解和快速操作，提升使用效率。左侧操作栏集中了所有功能选项和操作工具，用户可以通过它快速访问不同的功能模块；右侧展示栏则用于显示用户操作的结果和详细信息，如数据分类结果、类关联规则等。此外，本章还对算法实现模块进行了细致设计，并提供了伪代码描述，这为后续的编程实现提供了极大的帮助。在软件实现的后续阶段，将继续根据实际需求对设计进行调整和优化，以确保最终产品的高效性和实用性。

# 第4章 软件实现与测试

仅参考此格式进行排版。

关联分类规则挖掘软件采用前后端分离架构，前端基于Vue3框架开发，负责实现数据的上传与展示，以及算法结果的可视化功能，后端采用flask框架搭建，负责实现算法的主体功能，并根据请求参数返回给前端运行结果。

## 4.1 页面实现

基于 Vue 3 实现的前端页面中，页面设计分为操作栏和数据展示两大部分，这两部分密切配合以实现软件的所有功能。

操作栏部分主要负责用户操作和参数设置，包含多个交互组件。用户可以通过上传文件按钮选择本地文件进行上传。点击该按钮后，会弹出一个文件选择面板，用户可以在其中选择要上传的文件。确认文件后，文件将上传到服务器。同时，页面右侧的数据展示部分将显示上传文件的预览内容。此外，操作栏还包括数据预处理按钮和针对不同算法选择的按钮。点击数据预处理按钮后，将会触发监听事件，服务器会对上传的数据进行一系列处理，包括数据清洗、数据离散化和数字编码等。处理后的数据会返回到客户端，客户端将其渲染到数据展示区域，并覆盖原始数据，方便用户预览和对比数据预处理前后的差异。

操作栏还提供了两个滑动条，用于调节最小支持度和最小置信度。这些滑动条允许用户根据需要设置算法参数，从而影响算法的执行结果。操作栏中添加有不同算法的按钮，按钮被点击后，将触发其对应的监听事件，服务端处理数据并将结果返回给客户端进行展示。

数据展示部分则负责显示数据的详细信息和算法处理结果。初次上传文件后，原始数据将显示在此区域。经过数据预处理后，新的处理数据也会在这里展示，用户可以清楚地看到数据处理前后的变化。当用户选择不同的算法并设置相关参数后，算法的执行结果也会在数据展示部分展示。

通过操作栏于数据展示部分的结合，用户能够方便地上传文件、预处理数据、调整参数后选择算法，直观地查看到多种算法的处理结果。

上传文件后，右侧将展示用户上传的数据，此处以uci中的鸢尾花数据集为例

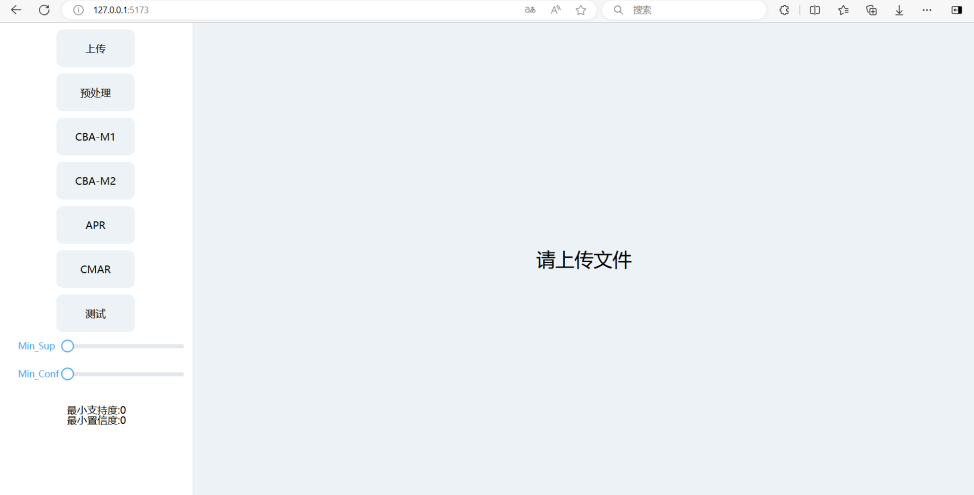


图4.1 软件初始界面

用户点击“上传”按钮后，可以选择本地csv文件进行上传，文件数据将在操作栏右侧展示，此处以UCI机器学习数据库中的鸢尾花（iris）数据集为例。



图4.2 上传iris数据集结果展示

点击“预处理”按钮后，服务器将返回预处理之后的数据。

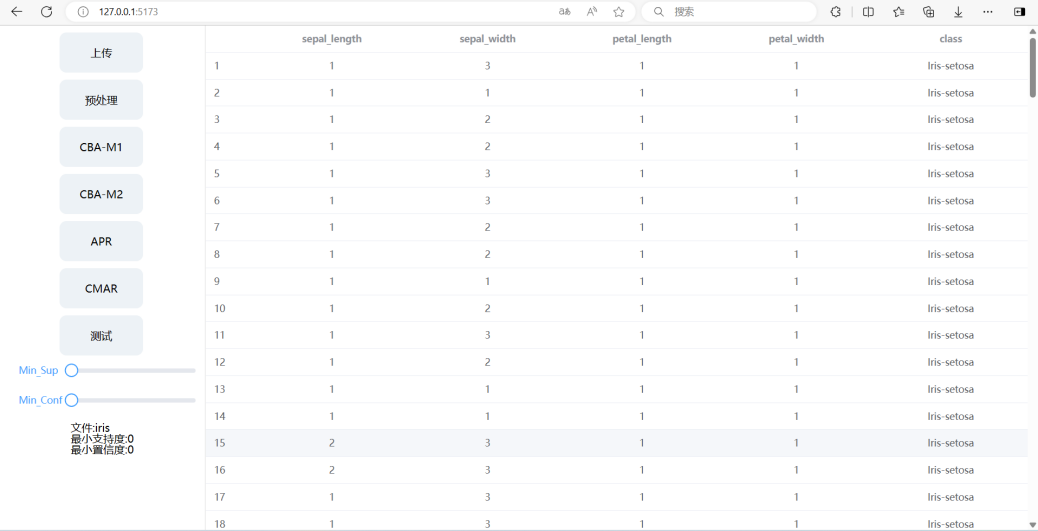


图4.3 预处理后的数据

使用鼠标操作最小置信度和最小支持度滑动栏，将参数值分别设置为为0.01和0.5，依次点击“CBA-M1”、“CBA-M2”、“APR”、“CMAR”按钮，右侧将展示对应的算法运行结果，包括分类准确率、算法运行时间、分类器中的默认类和分类器中类关联规则及其数目。见图4.4.，4.5，4.6，4.7

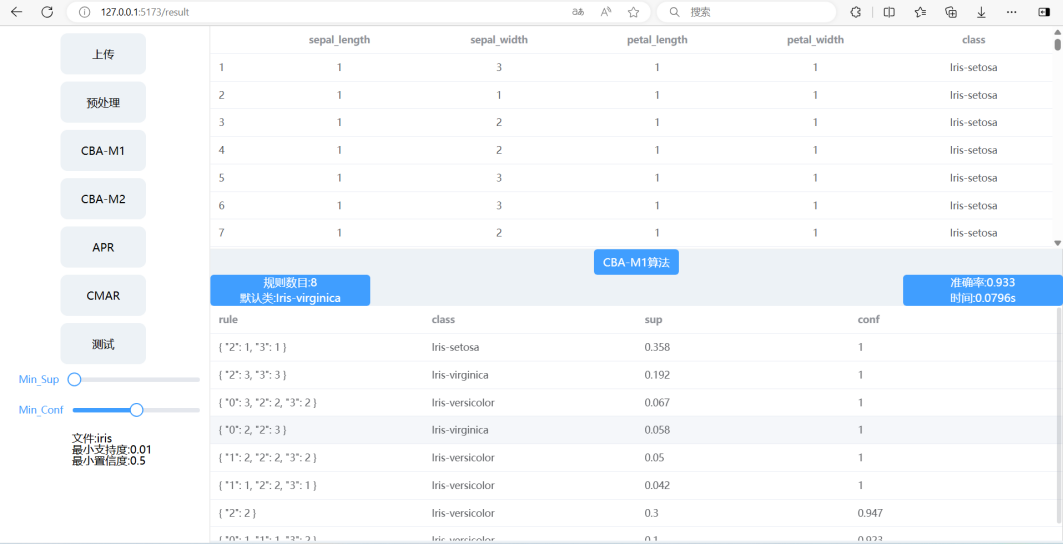


图4.4 CBA-M1算法运行结果

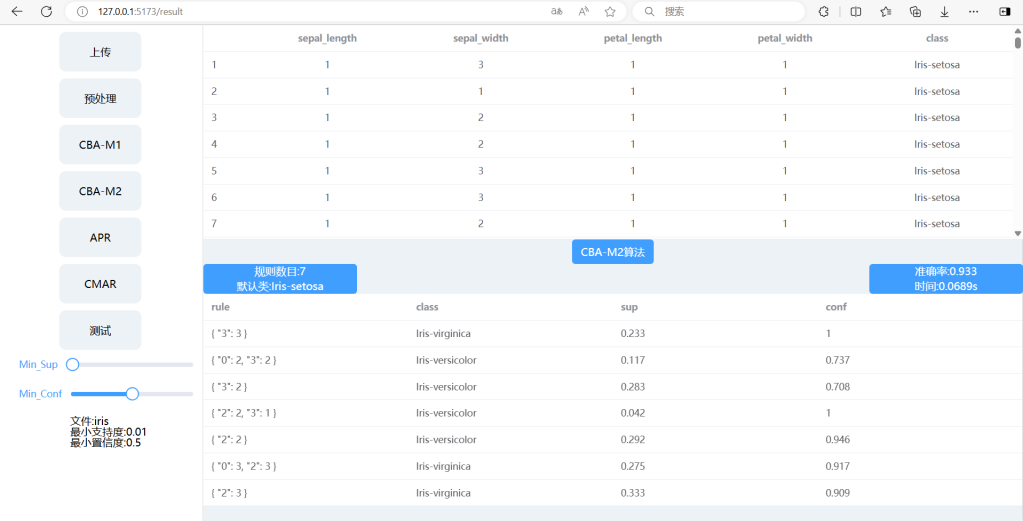


图4.5 CBA-M2算法运行结果

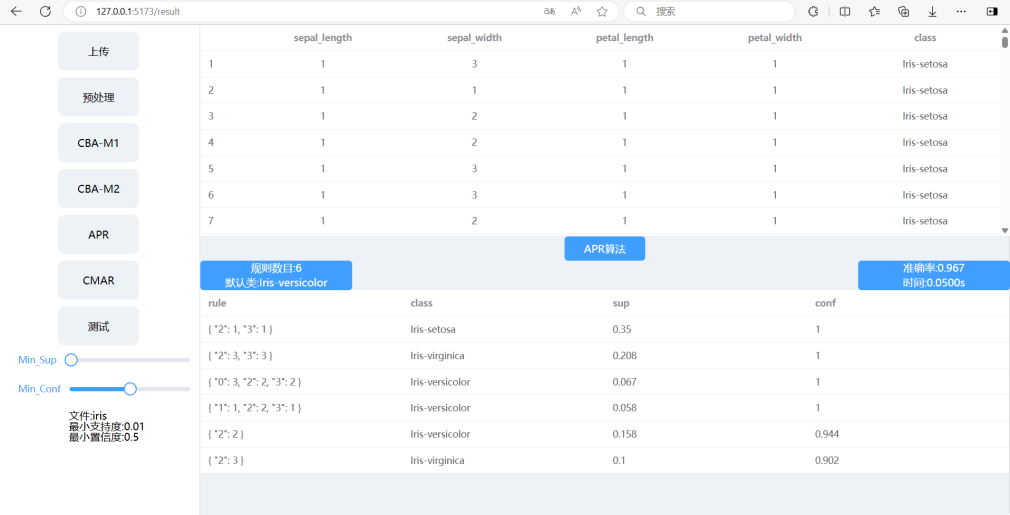


图4.6 APR算法运行结果

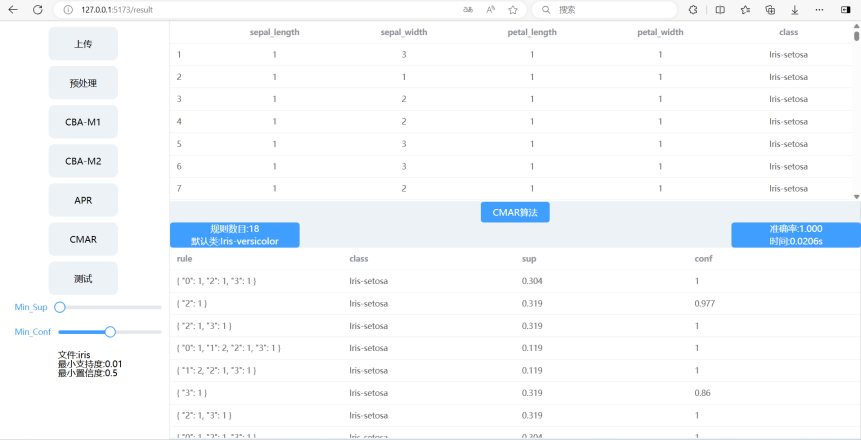


图4.7 CMAR算法运行结果

为了直观地比较不同算法，点击“测试”后，页面中将展示四种算法及其运行结果的图表，包含准确率、规则数目、时间开销三种指标。见图4.7

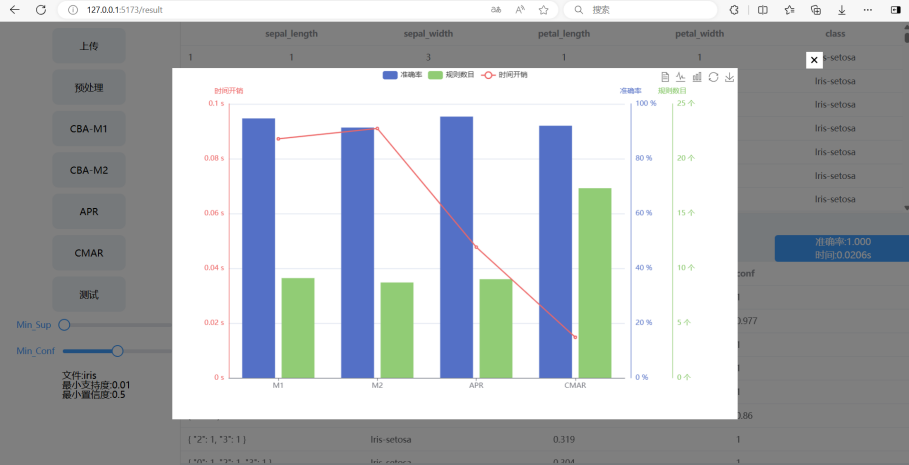


图4.8 四种算法测试运行结果

同时，对于测试结果，有多种不同的展示方式。见图4.9

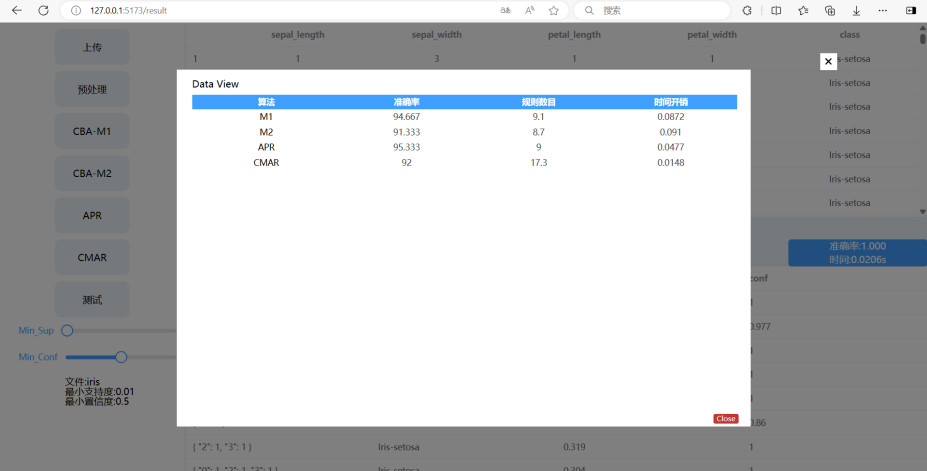


图4.9 结果总览

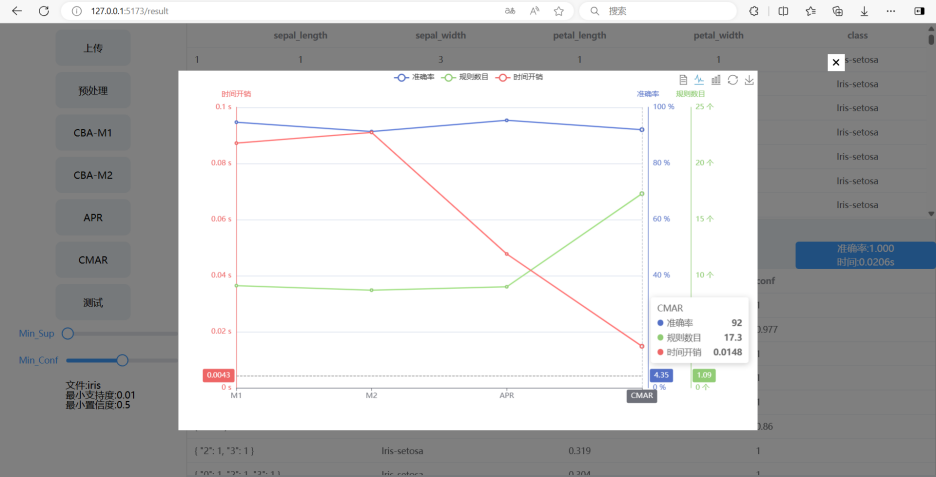


图4.10 结果折线图



图4.11 结果柱形图

4.2页眉

# 第5章 科学道德与学风

仅参考此格式进行排版。

# 第6章 参考文献的标注和要求

仅参考此格式进行排版。参考文献具体格式及排版见后面“参考文献”部分。

# 第7章 总结与展望

仅参考此格式进行排版。

学位论文应有结论，可以从论文的主要工作、创新点和后续的研究工作等方面进行总结。

## 7.1 主要工作与创新点

学位论文的结论是最终的、总体的结论，不是正文中各段的小结的简单重复。结论应该观点明确、严谨、完整、准确、精炼。文字必须简明扼要。可以从论文的主要工作、创新点和后续的研究工作等方面总结。

如果不可能导出应有的结论，也可以没有结论而进行必要的讨论。

可以在结论或讨论中提出建议、研究设想、仪器设备改进意见、尚待解决的问题等。不要简单重复罗列实验结果，要认真阐明本人在科研工作中创造性的成果和新见解，在本领域中的地位和作用，新见解的意义。对存在的问题和不足应作出客观的叙述。应严格区分自己的成果与他人（特别是导师的）科研成果的界限。

一般应按四级标题的方式给出，根据需要设置数量。如本文主要工作和创新点如下：

1）阐述第一个创新工作。不要把阅读文献当成创新工作。

2）阐述第二个创新工作。

3）阐述第三个创新工作。

4）阐述第四个创新工作。

特别提醒，不应简单和中文摘要内容相互拷贝。同一段文字或句子在本文中原则上只出现一次。

## 7.2 后续研究工作展望

针对工作不足或问题，说明更下一步深入的研究。如内容较多，也应用四级标题方式列出。

# 参考文献

参考文献是毕业设计(论文)不可缺少的组成部分，它反映论文作者的科学态度和毕业设计(论文)的取材来源、广博程度和可靠程度，同时能方便地把作者的研究成果与他人的成果区别开来。一份完整的参考文献也是向读者提供的一份有价值的信息资料。**参考文献应列入主要的中外文献，数量不少于15篇，教科书和硕士论文不多于5本，英文文献不少于5篇。**按论文中参考文献出现的次序，用中括号以数字连续编号，格式参考毕业设计（论文）模版。

1. 中国标准研究中心等. GB/T7713.1-2006. 学位论文编写规则[S]. 北京: 中国标准出版社, 2006.
2. 汪继祥. 作者编辑手册[M]. 北京: 科学出版社, 2004: 109, 118, 146-148.

# 致谢

致谢二字一级标题：黑体3号字居中，段前17磅，段后16.5磅，1.5倍行距，致谢二字与致谢内容之间不空行。致谢内容正文样式：宋体小四号，1.5倍行距。

可以从下列方面致谢：协助完成研究工作和提供便利条件的组织或个人；在研究工作中提出建议和提供帮助的人；给予转载和引用权的资料、图片、文献、研究思想和设想的所有者；其他应感谢的组织或个人。

主要感谢导师和对论文工作有直接贡献及帮助的人士和单位。学位申请人的家属及亲朋好友等与论文无直接关系的人员，一般不列入致谢的范围。

致谢辞应谦虚诚恳，实事求是，切忌浮夸与庸俗之词。

# 附录A 科技写作中非学术性低级错误的主要表现

对于一些不宜放入正文中，但作为毕业设计（论文）又不可缺的组成部分或具有重要参考价值的内容，可编入毕业设计(论文)的附录中，例如，公式的推演、源程序代码、附图等内容。附录的内容为备选项目，作者可根据内容的需要决定附录的项目数，用附录A、附录B方式编号。附录的篇幅不宜太多。附录与主体部分一起编制页码。若附录部分有手工制作或复印件，手工制作或复印件部分要装订在内但可以不计页码。附录的文字按照正文格式进行排版。