

# 程序报告

姓名：叶睿宸

学号：22551338

学院（系）专业：软件学院人工智能

## 1 算法描述

### 1.1 算法历史演变

规则引擎是一种基于规则的推理系统，广泛应用于专家系统、诊断系统和推荐系统等领域。其发展经历了从简单的硬编码规则到数据驱动的混合推理的演变。以下将介绍规则引擎的历史演变过程：

#### 1. 硬编码规则阶段

早期的规则引擎主要依赖领域专家手工定义规则，规则以“如果-那么”（IF-THEN）的形式存在。这种方法的特点是：

**优点：**规则明确、可解释性强。

**缺点：**规则维护成本高，难以扩展，且对复杂问题的覆盖有限。

在本项目中，它直接通过检测症状中是否同时包含“痰湿相关”、“气虚相关”和“发热相关”的关键词，来判断是否属于“痰湿内蕴兼气虚发热”复合证型。

这种规则明确且易于理解，但需要手动维护关键词列表，难以应对复杂的症状组合。

#### 2. 数据驱动的朴素贝叶斯阶段

随着数据量的增加，规则引擎逐渐引入统计学习方法。朴素贝叶斯是一种经典的生成式模型，假设特征之间相互独立，通过计算条件概率来进行分类。在本项目中通过 RuleIndex 类实现了基于朴素贝叶斯的分类器：

**先验概率：**通过类别文档计数计算，使用加一平滑避免零概率。

**条件概率：**通过特征词在类别中的出现频率计算，使用加法平滑。

**后验概率：**通过先验概率和条件概率的对数累加得到。

这种方法的特点是：

**优点：**通过数据驱动生成规则，减少了手工定义规则的工作量；加法平滑提高了模型的鲁棒性。

**缺点：**朴素贝叶斯假设特征独立性，可能无法捕捉复杂的特征交互关系。

#### 3. 混合推理阶段

现代规则引擎结合了硬编码规则和数据驱动方法，形成了混合推理框架。硬编码规则用于处理高置信度的简单场景，数据驱动方法用于复杂场景的推断。

这种方法的特点是：

**优点：**结合规则和数据驱动的优势，既保证了高置信度场景的准确性，又提高了复杂场景的覆盖率。

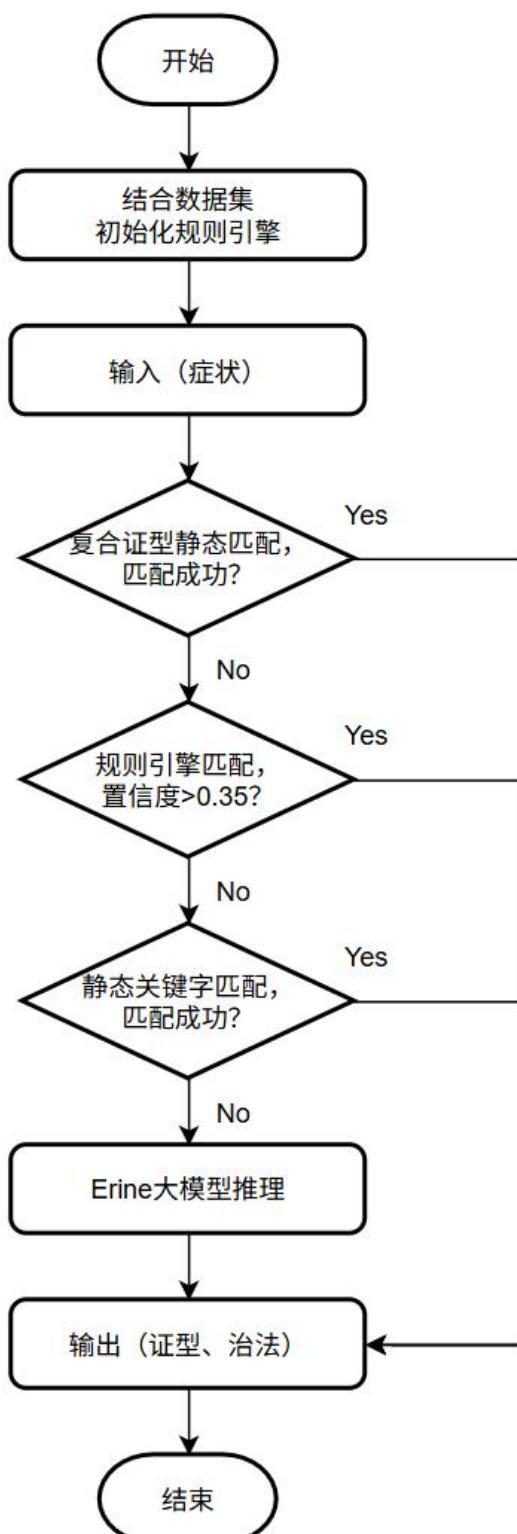
**缺点：**混合框架的实现较为复杂，规则和数据驱动部分需要协调。

规则引擎算法从硬编码规则到数据驱动，再到混合推理，经历了以下演变：硬编码规则：明确、可解释，但难以扩展。数据驱动（朴素贝叶斯）：减少手工规则，提升泛化能力，但假设简单。混合推理：结合规则和数据驱动，兼顾准确性和覆盖率。在本项目中，规则引擎通过硬编码规则、朴素贝叶斯分类器和静态规则的结合，形成了一个高效的中医证型推断系

统。这种混合推理框架在实际应用中具有很强的实用性。

## 1.2 算法流程图

如下展示的是带有规则引擎算法的中医辨证系统运行流程



## 2 算法性能分析

在本项目中，我首先尝试的是不带规则引擎版本，仅依赖大模型输出。其次尝试了不同版本的规则引擎，并对最后的计算结果进行性能分析。对比的算法主要如下所示：

**不带规则引擎：**仅依赖大模型（ERNIE）进行证型和治法的预测。使用 `call_large_model` 函数调用大模型，直接返回预测结果。

**基于硬编码规则引擎：**使用硬编码规则（如 `detect_combo`）对特定复合证型进行预测。对未覆盖的情况调用大模型进行预测。

**基于朴素贝叶斯的规则引擎：**在硬编码规则的基础上，加入基于朴素贝叶斯的分类器（RuleIndex）。使用数据驱动方法生成规则，减少手工定义规则的工作量。

**混合推理规则引擎（最终版本）：**结合硬编码规则、朴素贝叶斯分类器和大模型，形成混合推理框架。优先使用规则引擎处理高置信度场景，数据驱动方法补充，最后静态规则匹配和大模型兜底。

以下针对上述四种算法从准确性、效率、鲁棒性和可解释性四个维度对比分析。

### 2.1 准确性

不带规则引擎的情况，大模型具有强大的语义理解能力，能够处理复杂的症状描述。但是大模型并不具备数据集的先验知识，没有经过后训练对下游任务进行适配，因此回答的准确性可能会远远偏离预期答案。并且由于该方法是实验提供的默认版本，并且四种算法最后都由大模型输出作为兜底，因此这个准确性作为 `baseline`，可以视为最低准确率。

基于硬编码规则引擎版本，硬编码规则对高置信度的复合证型预测准确性高。但是该方式会完全依赖训练集导致过拟合，对于没有见过的规则，或是规则覆盖有限的范围，难以处理复杂或未见症状，泛化性能较差。其准确率略高于 `baseline`。

基于朴素贝叶斯的规则引擎，结合数据驱动方法，能够覆盖更多复杂症状，提高预测准确性。并且能避免完全人为的规则，导致忽略数据内在联系以及数据概率分布的情况。但是朴素贝叶斯假设特征独立性，可能无法捕捉复杂特征交互。最终实验情况，准确性高于 `baseline`。

混合推理规则引擎，结合规则和数据驱动的优势，能够处理大部分症状描述。但是规则和数据驱动部分的协调可能影响部分边界情况的预测。最终结果该算法在四种算法中准确性最高。

### 2.2 效率

不带规则引擎，大模型调用时间较长，尤其在高并发场景下性能较差。并且后续将分词转化为词向量也需要涉及远端 api 的调用，在调用过程受限于网络条件或网络带宽的影响，因此该种算法效率较低。

基于硬编码规则引擎版本，硬编码规则运行效率高，但未覆盖的情况仍需调用大模型。效率中等。

基于朴素贝叶斯的规则引擎，运行效率高，能在端侧处理大部分情况，减少了大模型调用频率。因此最终运行效率较高。

混合推理规则引擎，同理减少了大模型调用频率，但由于混合框架的实现复杂度较高，在部分数据集下的效率不如朴素贝叶斯的规则引擎，平均情况下效率最高。

## 2.3 鲁棒性

不带规则引擎，大模型对噪声数据的鲁棒性较好，但可能生成不符合领域知识的结果。四种算法中鲁棒性处于中等。

基于硬编码规则引擎版本，硬编码规则对噪声数据敏感，可能因关键词缺失而无法匹配。鲁棒性较低。

基于朴素贝叶斯的规则引擎，数据驱动方法对噪声数据的鲁棒性较硬编码规则更强。鲁棒性较至前两者更高。

混合推理规则引擎，规则引擎和大模型的结合提高了对噪声数据的鲁棒性。但规则引擎部分仍可能对关键词缺失敏感。总体情况下该方法的鲁棒性依旧是最高。

## 2.4 可解释性

不带规则引擎，大模型的预测结果缺乏明确的推理过程，用户难以理解预测依据。该算法的可解释性同等于大模型的可解释性，因此目前可解释性偏低。

基于硬编码规则引擎版本，硬编码规则具有较高的可解释性，因为它是完全按照用户能在数据集中捕获到的规律得到的规则，用户可以从编码中清楚了解预测依据。因此具有最高的可解释性。

基于朴素贝叶斯的规则引擎，朴素贝叶斯分类器具有一定的可解释性，但不如硬编码规则直观。可解释性中等。

混合推理规则引擎，按照人为预设的多重规则实现，先经过硬编码静态规则的匹配，当硬编码规则无法满足要求时，再考虑基于朴素贝叶斯的规则匹配。当规则引擎还是无法得到目标答案时，最终还有大模型参与兜底，完全符合人的认知，是一个依赖设计者惯用设计方案的算法，具有较高的可解释性。

## 3 研究展望

### 3.1 提升规则引擎的鲁棒性

当前规则引擎对关键词的依赖较强，可能对拼写错误、不完整描述或同义词缺乏鲁棒性。展望：引入模糊匹配算法（如编辑距离、Jaccard 相似度）来增强关键词匹配能力。使用预训练语言模型（如 BERT、ERNIE）对输入症状进行语义解析，生成更规范化的输入。构建更全面的同义词库，覆盖更多中医术语的变体。

### 3.2 优化混合推理框架

前混合推理框架通过规则引擎、朴素贝叶斯和大模型结合，虽然性能较好，但实现复杂度较高。可以通过引入强化学习优化规则引擎和数据驱动部分的协作，动态调整规则优先级。使用知识图谱（Knowledge Graph）增强规则引擎的推理能力，将中医领域知识结构化。在混合推理框架中加入更多的上下文信息（如患者病史、体检结果），提高推理的准确性。

### 3.3 提高大模型的效率与降低调用成本

大模型（ERNIE）在处理复杂症状时表现出色，但调用成本高且可解释性较低。考虑使

用轻量化模型（如蒸馏后的小型 Transformer）替代大模型，降低计算成本。在大模型输出中加入推理路径或依据（如关键词权重、语义匹配得分），增强可解释性。结合规则引擎和大模型，探索“可控生成”（Controllable Generation）技术，使大模型输出更符合领域知识。

### 3.4 扩展数据驱动方法

当前朴素贝叶斯分类器假设特征独立性，可能无法捕捉复杂的特征交互。可以引入更复杂的机器学习模型（如随机森林、XGBoost）或深度学习模型（如 RNN、Transformer）替代朴素贝叶斯。使用迁移学习技术，将其他中医数据集的知识迁移到本项目中，提升模型的泛化能力。增加数据标注量，构建更大规模、更高质量的中医证型数据集。

### 3.5 提升算法的可解释性

当前规则引擎和朴素贝叶斯分类器具有一定的可解释性，但大模型的预测结果缺乏明确的推理依据。可以采用“推理路径展示”，即在预测结果中加入规则匹配或概率计算的详细信息，增强用户对结果的信任。输出规则匹配日志，向用户展示规则引擎和大模型的推理过程。领域知识标注，在预测结果中标注与领域知识的对应关系，提升结果的可信度。