



汕頭大學
SHANTOU UNIVERSITY

硕士学位论文

基于基因表达式编程的
题 目 随机森林在急性低血压预测中的应用

Prediction of Acute Hypotensive Episodes Using
英文题目 Random Forest Based on Gene Expression Programming

姓 名 左右祥 学 号 11309037

所在学院 工学院 导师姓名 范衡

专 业 信号与信息处理

入学日期 2013 年 9 月 答辩日期 2016 年 5 月

学位论文原创性声明

本论文是我个人在导师指导下进行的工作研究及取得的研究成果。论文中除了特别加以标注和致谢的地方外，不包含其他人或其它机构已经发表或撰写过的研究成果。对本文的研究做出贡献的个人和集体，均已在论文中以明确方式标明。本人完全意识到本声明的法律责任由本人承担。

作者签名：_____ 日期：_____年____月____日

学位论文使用授权声明

本人授权汕头大学保存本学位论文的电子和纸质文档，允许论文被查阅和借阅；学校可将本学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或其它复制手段保存和汇编论文；学校可以向国家有关部门或机构送交论文并授权其保存、借阅或上网公布本学位论文的全部或部分内容。对于保密的论文，按照保密的有关规定和程序处理。

作者签名：_____ 导师签名：_____

日期：_____年____月____日 日期：_____年____月____日

摘要

在重症监护室（Intensive Care Unit 简称 ICU）中，急性低血压（Acute Hypotension Episode 简称 AHE）的发生往往会对患者引起严重的后果，对器官造成不可逆的损坏，衰竭，严重时甚至危及患者的生命。临床上主要依据医生的经验进行预见性的判断，由于目前 ICU 病房中医疗资源紧张，人手严重短缺的问题，并且急性低血压发作前缺少可以直接观测到的有效征兆，导致医护人员难以及时发现急性低血压的病人，致使病人的存活率低下。

ICU 病房中存着大量有关患者的临床医疗数据，其中包括病人的实时体征数据，例如病人的血压，脉搏，呼吸等，在计算机技术飞速发展的今天，若能利用计算机辅助诊断，提前预测出急性低血压的发生，将会为医生的诊断赢得宝贵的时间，从而降低病人的死亡率。

为了实现急性低血压的提前预测，本文采取基于基因表达式编程（Gene Expression Programming，简称 GEP）的随机森林（Random Forest，简称 RF）算法进行预测。主要分四步：1. 首先对平均动脉血压（Mean Arterial Pressure, 简称 MAP）信号进行经验模态分解（Empirical Mode Decomposition，简称 EMD）分解；2. 然后对分解后本征函数（Intrinsic Mode Function，简称 IMF）信号进行特征提取；3. 通过随机森林对数据和特征进行分组，对每组数据和特征用 GEP 进化算法进行进化出一个预测函数；4. 最后建立随机森林分类预测模型，并通过随机森林投票进行最终的分类。最后，利用 5 折交叉验证对该算法的泛化能力进行了测试，得到准确率为 77.55%，灵敏度 80.55%，特异性 75.14%。实验表明，该方法可以为 ICU 监护中 AHE（急性低血压）的发生提前预测和提前干预提供技术支撑，具有重要的临床应用价值。

关键词：急性低血压，EMD 分解，基因表达式编程，随机森林

Abstract

In ICU (intensive care unit, ICU), occurrence of acute hypotension episode (AHE) tends to cause serious consequences to the patient and cause irreversible damage to organs, exhaustion, even endanger the patient's life. The predictability of judgment is mainly based on the doctor's experience in clinically, because of the shortage of ICU medical resources currently, a serious shortage of manpower and the lack of directly observed effective symptom before the occurrence of acute hypotension episodes, so the medical staff is difficult to find the acute hypotension patients timely, resulting in the patient's survival rate is low.

There is a large number of clinical data about patients in ICU, including the patient's signs in real time data, such as the patient's blood pressure, pulse, respiration, in the rapid development of computer technology today, if we make use of the computer aided diagnosis, predicting the occurrence of acute hypotension in advance, it will win precious time for the doctor's diagnosis, thereby reducing the patient mortality.

In order to realize the prediction of acute hypotension, this paper takes an algorithm based on the gene expression programming (GEP) and random forest (RF). Can be divided into four steps: 1. The mean arterial pressure (MAP) signals are analyzed by empirical mode decomposition (EMD); 2. The algorithm obtains features of the signal through the Intrinsic Mode Function (IMF) signal produced by applying empirical mode decomposition (EMD) to the arterial blood pressure (MAP) signal in step 1; 3. Then the feature sets and the data sets are grouped to evolve decision functions via GEP. 4. Finally, a random forest is formed and the classification result is obtained by voting. The achieved accuracy of the proposed method

is 77.55%, the sensitivity is 80.55% and specificity is 75.14% after the five-fold cross-validation.

Keywords: Acute hypotensive episode; Empirical mode decomposition; Gene expression programming; Random forest

目录

摘要.....	I
Abstract.....	II
第一章 绪论.....	1
1.1 研究的背景及意义.....	1
1.2 研究的现状.....	2
1.3 本文的研究内容和结构安排.....	3
第二章 AHE 数据特征提取.....	5
2.1 特征提取概述.....	5
2.2 特征提取.....	6
2.2.1 数据集介绍.....	6
2.2.2 EMD 经验模态分解.....	7
2.2.3 特征提取.....	9
2.3 本章小结.....	11
第三章 AHE 分类算法.....	12
3.1 AHE 分类算法综述.....	12
3.2 基因表达式编程算法.....	13
3.2.1 遗传算法.....	13
3.2.2 基因编程.....	15
3.2.3 基因表达式编程.....	18
3.3 随机森林算法.....	24
3.3.1 随机森林的分组策略.....	25
3.3.2 决策树的建立.....	26
3.3.3 投票机制.....	32
3.4 GEP-RF 算法.....	34
3.4.1 决策树的建立.....	34
3.4.2 GEP-RF 算法.....	35
3.5 本章小结.....	37
第四章 论文的实验.....	38
4.1 实验框架.....	38
4.2 实验参数及结果.....	40
第五章 总结与展望.....	44
参考文献.....	45
致谢.....	48
个人简历.....	49

第一章 绪论

1.1 研究的背景及意义

随着经济和科学技术的飞速发展，人们生活质量的提高和对自身健康的重视，现代医疗卫生事业得到迅猛的发展，在现代社会发展中占据重要位置，并逐步成为衡量社会发展的重要因素之一。在现在的医院中，重症监护室（ICU）成为必须的医疗单元，对急性的，危及生命的危重病人提供生命支持和医疗照顾。采用先进的医疗设备，维护病人器官的正常功能和内环境的相对稳定，无疑成为危重病人的最后一道防线。

近几年，医疗监护技术的快速发展，大多数 ICU 都装备了先进的监测仪器和相关设备。对病人的生理生化指标进行经常性或连续性的监测，从而获得大量的临床数据。常规的生理监测主要包括心电图，血压，呼吸，心率，血氧饱和度，体温等生理体征指标。医疗仪器的监护技术大大减轻了医护人员的劳动强度，提高工作效率，及时向医生反应病人病情的发展状况，便于医生做出科学准确的治疗方案，从而减少并发症，降低危重病人的死亡率^[1]。

急性低血压是危重病人术后在 ICU 监护中经常遇到的凶险并发症之一。因为迅速降低的血压对身体组织器官的供血供氧不足，对器官组织的生成运行造成严重的阻碍，所以急性低血压不仅会破坏治疗效果，如果未及时发现治疗的话，会造成器官衰竭，对器官造成不可逆的损坏，甚至死亡，严重威胁中病人的生命安全。

急性低血压是指在一个小时的检测时间内有任何 30 分钟或更长时间内有至少有 90% 的平均动脉血压小于或等于 60mmHg。

如下图 1，横坐标为时间序列，纵坐标为平均动脉血压值。从图 1 中我们可以看到虽然在 40 分钟左右的时刻，出现平均动脉血压值小于 60mmHg，但是该次出现并未持续 30 分钟，所以 40 分钟时刻出现的平均动脉血压小于 60mmHg，并不能称为急性低血压。然而在 150 分钟到 180 分钟之间出现了持续 30 分钟的平均动脉血压值小于 60mmHg，所以在认定在 150-180 分钟期间发生了急性低血压。

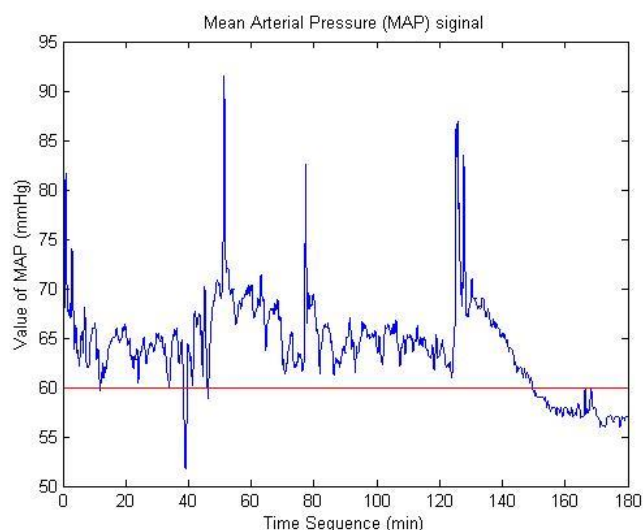


图 1. 含有急性低血压发生的平均动脉血压信号

根据美国复杂生理信号研究资源-PhysioNet 的数据库 MIMIC II (the Multi-Parameter Intelligent Monitoring for Intensive Care II, 即多参数智能重症监护数据库) 统计的 2008 年 12 月数据显示^[2], ICU 监护中记录了 1237 例患者的动脉血压, 其中术后监护中发生急性低血压的有 511 例, 死亡率达 37.6%。急性低血压发生后, 需要及时有效的干预, 如果不加以处理, 将会造成严重的后果。因此, 如何及时发现和处理急性低血压是术后监护过程中的一个重要问题。

目前临床上, 急性低血压的预测主要是靠具有丰富临床经验的医生预见性判断, 然而在医护资源紧张的 ICU 中, 这无疑降低医生的工作效率。随着计算机辅助诊断技术和信号分析处理方法的发展, 使得利用计算机根据临床数据进行预测急性低血压的发生成为一种可能。如果计算机能根据监测到的体征信号, 通过数据分析, 模式识别, 数据挖掘等手段, 进行预测急性低血压的发生, 将会为医生争取更为宝贵的干预时间, 对患者做出及时有效的干预措施, 极大减轻医生和医护人员的工作负担, 提高工作效率, 大大降低病人的死亡率。因此, 基于计算机的急性低血压预测的智能方法具有重要的临床应用意义和研究价值。

1.2 研究的现状

时序信号的预测是一个很难的问题, 并且在各个领域扮演着很重要的角色, 比如经济学中对股票的预测, 工程学对天气的预测, 以及生物医学对疾病的预测等。在生物医学领域, 对急性低血压预测的研究兴起于 21 世纪初期, 值得一提的是在 2009 年 PhysioNet/Computers 举办了一场关于 AHE 预测的竞赛, 该竞赛吸引大量学者在该领域的

研究。在 ICU 中，先进的医疗监测仪器能实时收集到大量的临床信息，包括血压（ABP），心率（HR），氧饱和度（SO₂）等^[3]，在这篇文章中，我们利用 ABP 信号来训练一个分类器去预测 AHE 的发生。

2001 年，J. Bassale 用参数非参数的方法表示和分析低血压之前的 ABP 信号，最终得出 ABP 信号的变化和形状能够预测急性低血压的发生^[4]。

2002 年，C. Crespo 假设 ABP 信号形态学的变化能够在急性低血压发生之前被检测到，发现 ABP 信号的方差和冲击波斜率的方差在预测急性低血压时是一些重要的因素^[5]，

2007 年，M. Saeed^[6]一种基于将时序信号数据转换为直观的符号表示的临时相似度衡量，他们用小波分解在多个时间尺度上进行表征时序信号。他们的方法用于辨别在 ICU 病人的血液动力学方面的时序信号之间的相似生理模式，还有一个潜在的用途就是能够进行预测血液动力学方面的疾病的发生。

2008 年 L. Lehman^[7]一种基于相似性搜索和模式匹配的算法来区分来自于大规模，多参数的数据库中有相似动态的时序信号。

2009 年，X. Chen^[8]窗口附近的 ABP 信号上提取出 6 个基本指数，包括 ABP 的均值和舒张压，利用这基本指数对急性低血压进行基本的分类。M. A. Mneimneh^[9]提出一个基于规则的方法，用动脉血压信号的均值作为预测急性低血压的指标。PA. Fournier^[10]用两个分布的信息散度，去找出最有识别能力的特征，并用这些特征去训练基于最近邻算法的分类模型。

2011 年，T. Rocha^[3]提出将多模神经网络应用在急性低血压的预测中，主要分两个阶段，第一阶段主要是训练模型，该模型是基于当前血压时序信号和收集到的历史血压信号模板之间的分析。第二阶段用多模态的结构算法在当前血压信号未来演化的基础上进行预测急性低血压。

2013 年，V. Awandekar^[11]提出一个算法，该算法是用 MAP 信号的概率分布和信息散度的方法去计算两个概率分布之间的统计距离。

从上述研究可知，对急性低血压的预测的研究思路大体上都是由先对时序信号进行特征提取，然后将提取好的特征输入到机器学习分类预测算法中进行分类模型的训练，最终得到对该时序信号的分类器。

1.3 本文的研究内容和结构安排

本文提出一个基于随机森林的预测算法，随机森林算法分类器是已被成功证明的强大

的集成算法之一。特别是该随机森林在模式识别和机器学习中的高维分类问题上的强大能力^{[12][13]}。随机森林中的生成决策树的方法是基于 CART 算法的方法^[12]。基因表达式编程是一种以自然进化为基础的演化算法，由于基因表达式编程强大的全局搜索能力，在本文中，随机森林中的决策树不再利用 CART 的方法生成，而是改用基因表达式编程的方法全局搜索最优决策树。正如本文实验结果中所得，该方法比传统的随机森林方法得到更高的预测准确率。

该算法（基于基因表达式编程的随机森林，简称 GEP-RF）主要包括两部分：第一部分，利用基因表达式编程自动进化并选出最有决策树；第二部分是通过得到的决策树进行投票来获得最终预测结果。

本文中样本特征的获取：首先将 MAP 信号通过经验模态分解（EMD），从得到的本征信号（IMF）中提取出的统计特征。EMD 是一种处理非平稳信号的方法，通过对原始信号的逐步分解获得信号的局部特征。

特征获取后，对数据样本和特征进行分组，然后通过基因表达式编程对每组数据和特征进行进化出一个最优的决策函数，众多决策函数组成随机森林，最终通过投票机制进行得到最终的预测结果。

本文各个章节的结构安排如下：

第一章：绪论。说明论文工作的背景意义，介绍国内外 AHE 预测的发展状况。

第二章：主要是 AHE 数据特征的提取以及数据集的介绍。

第三章：主要介绍本文的算法实现部分，包括基因表达式编程，随机森林以及二者结合而出的新算法。

第四章：本文所做的实验。

第五章：总结与展望。本章节会对本文所讨论的算法做出优缺点分析与小结，并分析在本文的基础上，还可以加深对某一些领域的展望。

第二章 AHE 数据特征提取

2.1 特征提取概述

特征提取在提高分类的准确性中起着非常关键的作用，然而特征提取的方法主要有四大类^[14]：

1. 基于基本统计方法的特征提取，就是提取数据波形的均值，方差，极值，过零点，功率谱，平均功率谱等基本统计特征来代表原有的时序数据。
2. 基于模型的特征提取，是指用模型去刻画时间序列数据，然后提取模型的系数作为特征矢量去进行分类器训练，比如对于平稳时间序列，可通过自回归模型等来进行拟合。
3. 基于变换的特征提取，通过变换的手段，使适合分类的特性突显出来，也是经常用到的特征提取方法。变换包括时频变换，线性变换等，比如通过傅里叶变换提取频域特征，小波变换等，提取特征进行训练分类模型。
4. 基于分形理论的特征提取，分形是指具有无限精细，非常不规则，无穷自相似和非整点的点集。在大自然中，海岸线，雪花，云雾这些不规则的形体都属于分形，即部分与整体有自相似性。基于分形理论的特征提取是用它的定量分析指标分维数作为特征矢量，进行数据分类，建立分类模型。

在急性低血压特征提取方面，2011 年王志刚等^[15]运用医学信息学理论，选取功率，谱幅度的中位数，平均值，最大值，标准偏差和极差等特征，运用支持向量机对急性低血压信号进行预测。陈朝虹等运用双谱分析提取动脉血压数据特征，构建特征向量，利用 BP 神经网络对急性低血压信号进行预测^[16]。赖丽娟等对发生和未发生的急性低血压信号两者之间的平均动脉压信号进行小波多尺度分解，并选取各层小波系数的统计特征参数中位数和最大值，用支持向量机对急性低血压信号进行分类预测^[17]。

本文是先用基于变换的特征提取方法，即对 MAP 信号进行经验模态分解（EMD），然后对分解之后对得到的本征信号（IMF）进行提取特征，提取的特征主要有统计特征，频域特征以及一些其他特征。

2.2 特征提取

2.2.1 数据集介绍

实验所采用的数据集来自美国 PhysioNet 网站的 MIMIC-II 数据库的数据，这些数据均为美国 ICU 病房病人的真实数据。在该数据库中收集了超过 30,000 个病重监护病人的临床数据。数据中包含了大量的医疗记录和生理数据，其中大概有 5000 条记录包含生理波形数据，例如：心电图，血压和呼吸等。

该文章利用 MIMIC-II 数据库中的 1599 条数据进行训练模型和测试，其中 799 条 AHE 记录的 ABP 信号，800 条正常 ABP 信号。该数据为以 125HZ 频率采集得到 ABP 的时序信号数据。然后再对该采样后的时序信号，求得其平均动脉血压（MAP）。MAP 由舒张压（DABP）和收缩压（SABP）组合求得如下公式：

$$MAP = DABP + \frac{SABP - DABP}{3} \quad (1)$$

根据公式求得平均动脉血压信号如下图 2：

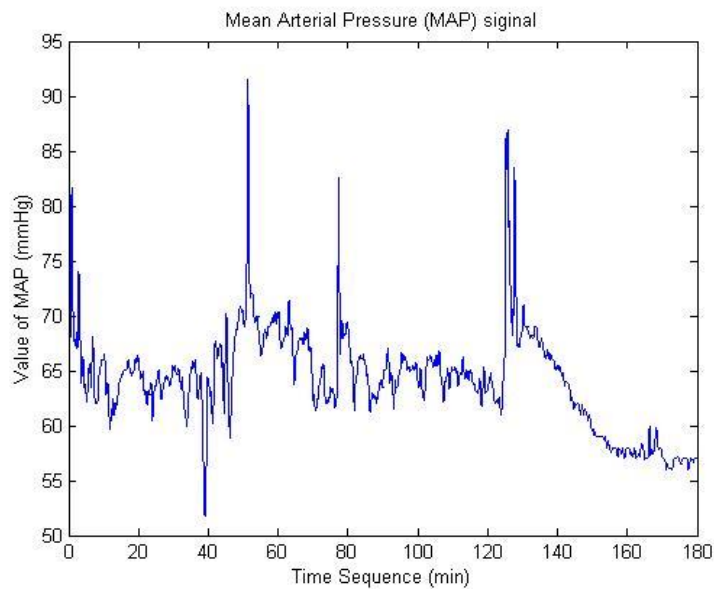


图 2. 平均动脉血压信号

获得 ABP 信号的平均动脉血压信号，是论文的第一步。在对 ABP 信号进行急性低血压的预测，实际上是对有 ABP 信号而来的 MAP 信号进行预测。论文中我们根据 2.5 小时的 MAP 信号，来预测在接下来的 30 分钟内是否会出现急性低血压。如下图 3：

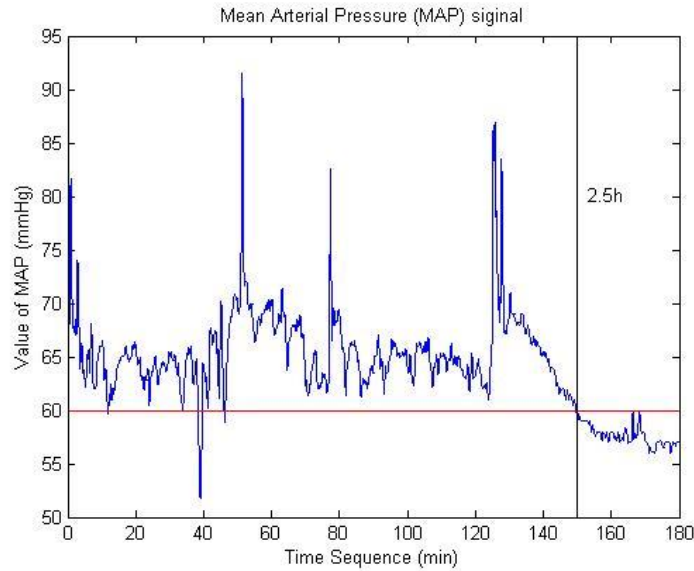


图 3. 含有急性低血压的 MAP 信号

从图 3 中，我们可以看到前 2.5 小时未发生急性低血压，而在 2.5 小时后发生了急性低血压。在该论文中，我们即根据前 2.5 小时的 MAP 信号中所含有的特征，对后 0.5 小时是否发生急性低血压进行预测。

2.2.2 EMD 经验模态分解

经验模态分解 (Empirical Mode Decomposition, EMD) 法是黄镌 (N. E. Huang) 在美国国家宇航局与其他人于 1998 年创造性地提出的一种新型自适应信号时频处理方法，特别适用于非线性非平稳信号的分析处理^[18]。文章中所用特征来自于对 MAP 信号进行 EMD 分解得到的 IMF 信号。EMD 是一个自适应的方法，将一个信号分解成本征模态信号^[19]。该方法由 N. E. Huang 在 1998 年作为一个非线性非平稳信号的处理工具提出，并应用在很多应用之中^[20]中。

EMD 将一个复杂的信号分解成几个本征模态信号 (IMF)，其中每一个 IMF 都满足一下两个条件^[19]：

1. IMF 信号的极值点的个数（极大值和极小值）和过零点的个数相等或者最多相差一个。
2. IMF 信号具有局部对称，并且上下包络线的均值为 0.

正如^[20]中所述，将原始信号分解成 IMF 是一个迭代算法。迭代的终止条件如下^[19]：

1. 残差信号小于一个预先设定的阈值。
2. 残差信号是一个模态函数，不能再分解成更多的 IMF 了。

经验模态分解的过程可以被总结为以下步骤：

- 1) 假设 $i=1$
- 2) 找到信号 $x(n)$ 的所有极大值和极小值点
- 3) 求得信号 $x(n)$ 的极小值和极大值的包络线： $e_{min}(n), e_{max}(n)$
- 4) 计算上下包络线的均值：

$$m(n) = \frac{e_{min}(n) + e_{max}(n)}{2} \quad (2)$$

- 5) 计算信号 $x(n)$ 与均值信号 $m(n)$ 的差值信号： $h(n) = x(n) - m(n)$

- 6) 将 $h(n)$ 作为一个新的信号继续步骤 1-4 的迭代，直到满足终止条件。假设

$c_i(n) = h(n)$, 然后将残差信号 $x(n) - h(n)$ 作为一个新的信号 $x(n)$ 继续该过程，并 $i+1$ 。

在 EMD 分解完成之后，原始信号可以写作其 IMFs 之和，如下公式：

$$x(n) = \sum_{i=1}^N c_i(n) \quad (3)$$

例如，下图 4 就是将一个急性低血压的 MAP 信号通过经验模态分解后得到的 IMFs。其中左上角第一个子图为原始 MAP 信号，后面的 11 个为其分解得到的 IMFs。

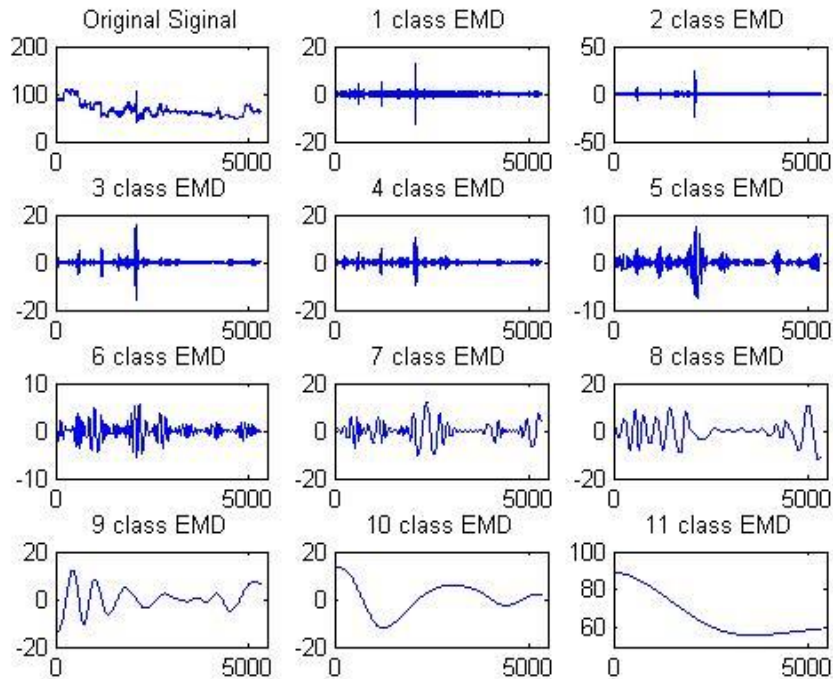


图 4. 原始信号和其 IMFs

从上图中，我们可以看出，每一个 IMF 信号均满足其两个条件：1. 极值点个数和过零点个数相差不超过 1, 2. 局部对称，上下包络线均值为 0。

为了保证每个信号提取出的特征个数相同，这里在处理 MAP 信号得到的 IMF 时用到了一个方法，因为每一个 MAP 信号经过 EMD 分解后得到的 IMF 个数不一定相同，我们要得到相同个数的特征就要保证每一个 MAP 信号分解的 IMF 个数相同。所以这里根据大多数 MAP 信号分解成 11 个 IMF 后分解停止，部分分解后的 IMF 个数大于 11，部分小于 11。将那些分解后 IMF 个数为 n 多于 11 的，保留第 n 个趋势信号，删除第 $n-1, n-2, \dots$ ，直至剩余 11 个 IMF。而将那些分解后为 m 少于 11 个 IMF 的信号，将第 m 个 IMF 信号进行复制直至凑够 11 个 IMF 信号。

2.2.3 特征提取

在每一个 MAP 信号经过经验模态分解得到本征信号之后，在原始信号和 IMFs 中提取出一些统计特性作为训练模型所用的特征。其中包括原始信号和 IMFs 的最小值，最大值，均值，中值和方差。

另外，一些其他的特征包括：

1. 每一个 IMF 信号的最大瞬时频率，最大瞬时频率由下公式 4 计算得到。

$$\max\left(\frac{d}{dt}(\angle \text{hilbert}(x(t)))\right) \quad (4)$$

这个特征表示 IMF 信号的瞬时变化量，其中 hilbert 是对一个复杂信号数学上的转换， $\angle$ 表示该复杂信号的相位^[19]。

2. 高频能量与低频能量的比值，在原始 MAP 信号被分解成 IMFs 后，原始信号 MAP 可以写为如下：

$$MAP = \sum_{i=1}^N c_i(n) \quad (5)$$

其中 $c_i(n)$ 为 MAP 信号的第 i 个本征信号，现在我们可以定义高频能量 E_1 和低频能量 E_2 如下：

$$E_1 = \sum_{i=1}^{\lfloor N/2 \rfloor} \varepsilon\{c_i(n)\} \quad (6)$$

$$E_2 = \sum_{i=\lfloor N/2 \rfloor + 1}^N \varepsilon\{c_i(n)\} \quad (7)$$

其中 $\varepsilon\{c_i(n)\}$ 表示 $c_i(n)$ 的能量，计算如下：

$$\varepsilon\{c_i(n)\} = \sum_{t=-\infty}^{\infty} c_i^2(n) \quad (8)$$

E1 和 E2 如此定义是合理的，因为每一个 IMF 的频率随着 EMD 的分解逐步降为 0。正如在图 1 中显示，第一个 IMF 信号相对于最后一个 IMF 信号含有更高的频率^[19]。而最后一个 IMF 信号的频率几乎为 0。

此外，还提取了最后一个 IMF 和原始信号的 12%分位数，以及最后一个 IMF 的偏斜，峰度和众数作为训练分类模型的特征。这里介绍下偏斜，峰度和众数。

a. 偏斜：

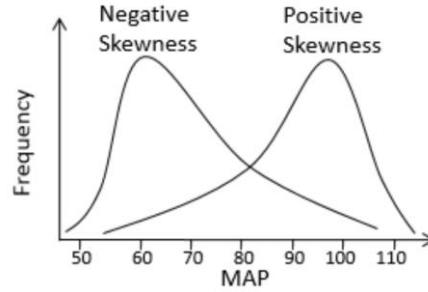


图 4. 偏斜

偏斜是对统计分布偏斜方向及程度的度量，正的偏斜说明样本数据中位于右侧的数据多余位于左侧的数据，而负的偏斜说明样本数据中位于左侧的数据多余右侧的数据。如果样本分布呈正态分布，则偏斜为 0。如上图 4：偏斜值的计算公式如下公式：其中 μ 为样本均值， σ 为样本标准差。

$$skewness = \frac{E(x - \mu)^3}{\sigma^3} \quad (9)$$

b. 峰度：

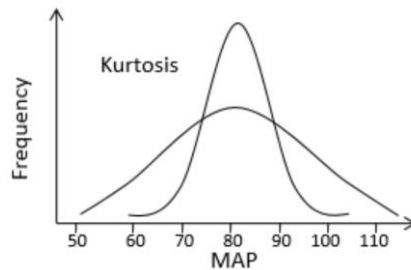


图 5. 峰度

峰度描述的是样本分布形态的陡缓程度。峰度值为 3 表示与正态分布相同，大于 3 表示比正态分布陡峭，小于 3 表示比正态分布更为平坦。峰度的计算公式如上图 5：其中 μ 为样本均值， σ 为样本标准差。

$$kurtosis = \frac{E(x - \mu)^4}{\sigma^4} \quad (10)$$

c. 众数:

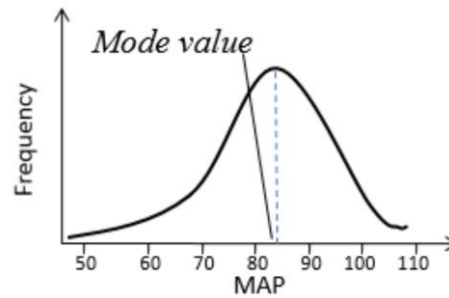


图 6. 众数

众数是 MAP 时间序列中出现次数最多的 MAP 值，在分布形态上表现为分布顶点对应的横坐标值如上图 6，如果分布中出现多个顶点，则取第一个。

至此，我们一共获得 77 个特征：原始信号和 11 个 IMF 信号的最小值，最大值，均值，中值和方差共 60 个，每个 IMF 的最大瞬时频率 11 个，高频低频能量比 1 个，原始信号的 12 分位数 1 个，最后一个 IMF 信号的 12 分位数，偏斜，峰度和众数共 4 个。总共 77 个特征，用来对分类模型进行训练。

2.3 本章小结

本章主要介绍了急性低血压方面的特征提取部分，简单介绍了国内外对急性低血压信号的特征提取，包括利用小波变换，双谱分析等方法进行特征提取，并简单介绍了数据集的来源。本章的重点在与经验模态分解的实现原理，将 EMD 的方法应用在急性低血压信号 MAP 上，对 MAP 信号进行分解，并详细介绍了对分解后的 IMFs 信号的特征提取。

第三章 AHE 分类算法

3.1 AHE 分类算法综述

分类是数据挖掘，机器学习和模式识别中一个重要的研究领域。解决分类问题的方法很多，单一的分类方法主要包括：决策树，贝叶斯，神经网络，k-近邻，支持向量机和基于关联规则的分类等，另外还有一些包含单一分类方法分集成学习算法，比如 AdaBoost，随机森林等集成算法。其中本文采取的分类方法为基于进化算法的随机森林分类算法。这里简单介绍一下以下几个常见的分类算法：

1. 决策树，决策树算法是用于分类和预测的主要技术之一，是一种非常常用的分类算法，决策树是一种树形结构，其中每个内部节点表示一个属性上的测试，每个分支代表一个测试输出，每个叶节点代表一种类别，根据测试属性值，通过对几种可能方案的选择，得到最终的解决方案^[21]。主要的决策树生成算法有 ID3，C4.5，CART 等算法。
2. 贝叶斯分类器，贝叶斯分类器是一种利用概率统计知识进行分类的算法，如朴素贝叶斯算法。这些算法主要贝叶斯定理来预测一个未知类别的样本属于各个类别的可能性，选择其中可能性最大的一个类别作为该样本的最终类别。
3. 神经网络，人工神经网络是在现代神经科学基础上提出的，类似于大脑神经进行信息处理的数学模型。在大脑神经中，信息的处理通过突触进行交换，而在这种模型中，存在大量的节点（类似突触）之间相互联接构成网络，即“神经网络”，从而达到处理信息的目的。神经网络的处理往往需要通过样本对模型进行训练，训练的过程就是模型进行学习的过程。训练通过改变节点的权值，从而使其具有分类的功能，经过训练的人工神经网络就可用于对象的识别。比如 BP 神经网络。
4. 支持向量机，支持向量机（SVM）是根据统计学习而提出的一种新的分类方法，其优点是结构风险最小化的准则，以最大化分类间隔为标准来构造最优的分类超平面，进而来提高学习机的泛化能力，优秀地解决了高维数，非线性和局部极小点等问题。对于分类问题而言，SVM 根据区域中的样本计算出该区域的决策曲面，并由此确定该区域中未知样本的类别。
5. K-近邻，k-近邻(kNN, k-Nearest Neighbors)算法是一种基于样本的分类方法。该方法就是找出与新的未知样本 x 距离最近的 k 个已知类别的训练样本，看这 k 个训

训练样本中大多数属于哪一类，然后就把 x 归为那一类。KNN 是一种懒惰学习方法，它并不需要进行分类模型的训练，只是简单的在分类时候进行样本距离的计算。这样若样本集比较复杂庞大，就可能会导致很大的计算开销，导致 KNN 算法无法应用到要求实时性很强的场合。

6. 随机森林，随机森林是一种集成学习方法。因为实际应用的复杂性和数据的多样性往往使得单一的分类方法无法达到预期有效的分类效果，因此诞生了多种分类算法融合的学习算法。本文中用的机器学习算法就是利用多颗决策树进行投票得出最终分类结果的集成算法，该算法克服了单颗决策树在该问题下分类准确率低下的问题。最终得出一个较为理想的分类器。

对于急性低血压的预测方面的分类算法，正如前面国内外研究现状中提及的，主要是从时序信号中运用不同的方法进行特征提取，然后基于机器学习算法进行训练模型。例如，利用 2011 年 Teresa Rocha 运用神经网络进行对急性低血压预测^[3]，2011 年赖丽娟运用支持向量机进行 AHE 的预测^[15]。

本文是基于基因表达式编程的演化算法的强大的全局搜索能力，寻找最优的决策树，通过对数据和特征的分组，实现在不同数据特征组中寻找出不同的决策树，最后众多决策树组合成随机森林，利用随机森林的投票机制进行得出最后的分类结果。

3.2 基因表达式编程算法

3.2.1 遗传算法

遗传算法类似于自然进化，通过在众多个体中寻找出最优的一个个体进行解决问题。它是 20 世纪 60 年代美国密执安大学 Holland J.H 教授提出^[22]，并随后由他和他的学生发展起来的。首先，通过随机产生一个种群，包含多个个体，个体之间通过遗传操作比如：交叉，变异，复制等进行产生新的个体。同自然界一样，遗传算法对求解的问题一无所知，在保持种群大小恒定的情况下，通过对每个新个体进行一个适应值的数值评价，淘汰适应值低的个体，选出适应值高的个体，让适应值高的个体再进行遗传操作产生新的个体，如此循环进化下去，直到种群收敛或者满足终止条件。这就是遗传算法的基本原理。

遗传算法的主要步骤如下：

- 1) 随机产生一个有固定长度的特征串组成的染色体的初始种群。
- 2) 对初始种群中的个体进行复制，交叉，变异等遗传操作，产生新的个体。

3) 对种群中的每个个体进行适应值评价, 在保持种群大小不变的情况下进行淘汰适应值低的个体。

4) 对种群一直执行步骤 2, 3 直至满足停止标准:

遗传算法一般流程图如下:

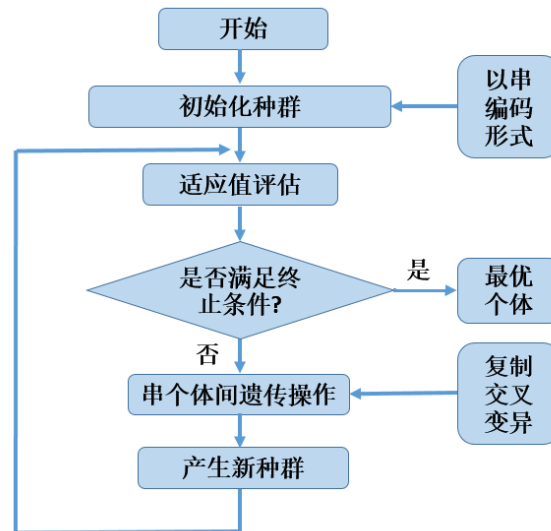


图 7 . 遗传算法流程图

从该流程图中, 我们可以看出, 首先是进行初始化种群, 该种群中的个体是以串编码形式体现。之后就是对种群中的个体进行适应值评估, 判断是否满足进化终止条件, 如果不满足则对种群进行遗产操作, 产生出新的种群, 直至满足终止条件, 输出最优个体, 则进化结束。

一般情况下, 种群的收敛标准有以下三种情况:

- (1) 种群收敛, 即种群中的所有个体都一样, 通过遗传操作产生不出新的个体。
- (2) 达到预先设置的进化代数, 可以预先设置一个进化代数, 达到之后停止进化。
- (3) 达到预先设置的评价次数, 可以设置进化评价多少次即可停止。

当进化停止后, 种群中的最优个体, 即是解决问题的最优方法。比如, 我们需要通过遗传算法进行从众多特征中选取比较好的特征, 此时先进行初始化种群, 种群中的个体都是随机产生的字符串如: 0101001101, 该个体在生物学中成为染色体, 该染色体由 10 个基因组成。这里的 0 表示该特征未被选中, 1 表示选中, 最后通过评价函数计算出该个体的适应值, 评价函数可以是将选中的特征输入到机器学习算法中比如支持向量机分类中得到的准确率, 最后进化结束得到的最优个体就是最终使分类准确率最高的个体, 对应的即是最优的特征。

遗传算法中个体之间的遗传操作主要包括复制, 交叉和变异。其中复制很简单, 即将

父代中的某个个体完整的保留在子代中，而交叉，则如下图 8：

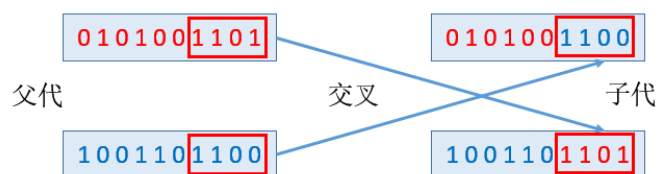


图 8. 遗传算法个体间的交叉操作

从上图我们可以看出，在两个父代的最后四个基因进行了互换，产生出了新的个体。交叉操作中交叉点（即从哪个基因开始进行互换）的选择是随机选择的。

在遗传操作中的变异操作则如下图 9：



图 9. 遗传算法中变异操作

在变异操作中，随机选择变异的基因，如上图将染色体上的第 6 个基因进行了变异。

总结起来，遗传算法中的个体是进行简单的串编码，通过简单的复制交叉变异等遗传操作进行进化，解决简单的问题，面对复杂的问题，比如进化决策树，遗传算法还是未能解决。

3.2.2 基因编程

基因编程是将遗传算法的思想用在以计算机程式为种群的进化计算^{[20][21]}。基因编程和遗传算法的主要区别是在于个体的编码方式上。在遗传算法中，个体的编码方式是进行串编码，然而，在基因编程中，个体的编码方式是树编码。例如下图 10：

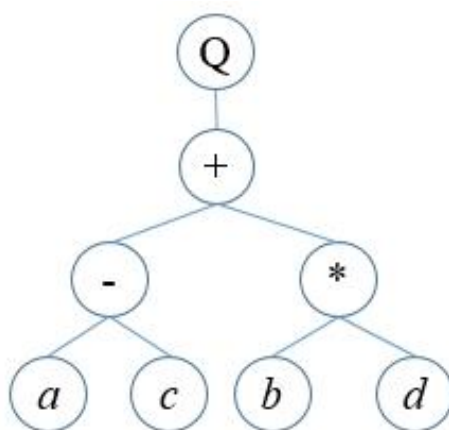


图 10. 基因表达式编程个体编码方式

在上面的树编码中，像 Q, +, -, *等在节点上的称为操作符，其中 Q 为执行开根号操作，+, -, *为算术运算符，而像 a, c, b, d 在叶子上的称之为参数。当然，像图 2 这样的树编码，解码成数学表达式如： $\sqrt{(a - c) + b * d}$

如此编码，与遗传算法比起来，采取了更加复杂的树结构编码，因此能解决更加复杂的问题^[21]。1992 年，J. Koza 在他的书《基因编程》中详细描述了此算法^[15]。

如上所述，基因编程与遗传算法的主要区别是在于对种群中个体的编码形式，在遗传算法中个体的编码方式为串编码，而在基因编程中是以树编码的形式形成个体。一个基因编程算法的流程图如下图 11：

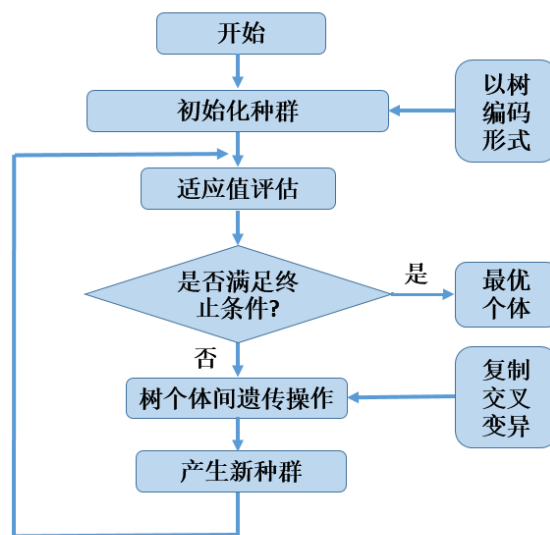


图 11. 基因表达式编程流程图

从该流程图上可以看出，算法的思想和遗传算法十分相似。区别仅在于个体的编码形式，当然由于编码形式的变化，导致遗传操作形式的变化。首先初始化种群，适应值评估，评估之后看下是否满足终止条件，如果不满足终止条件，则进行遗传操作，包括树个体之间的复制，交叉，变异等，来产生新个体，然后对新个体再进行评估等下一轮进化。直至满足终止条件，输出最优个体。

在这里，适应值评估函数是该个体在数据样本上的分类的准确率。这里的遗传操作交叉变异和复制，其中交叉是指两个子树进行互换，如下图 12，将父代 1 的右子树和父代 2 中的右子树进行互换，以产生出新的个体子代 1 和子代 2，如下图：

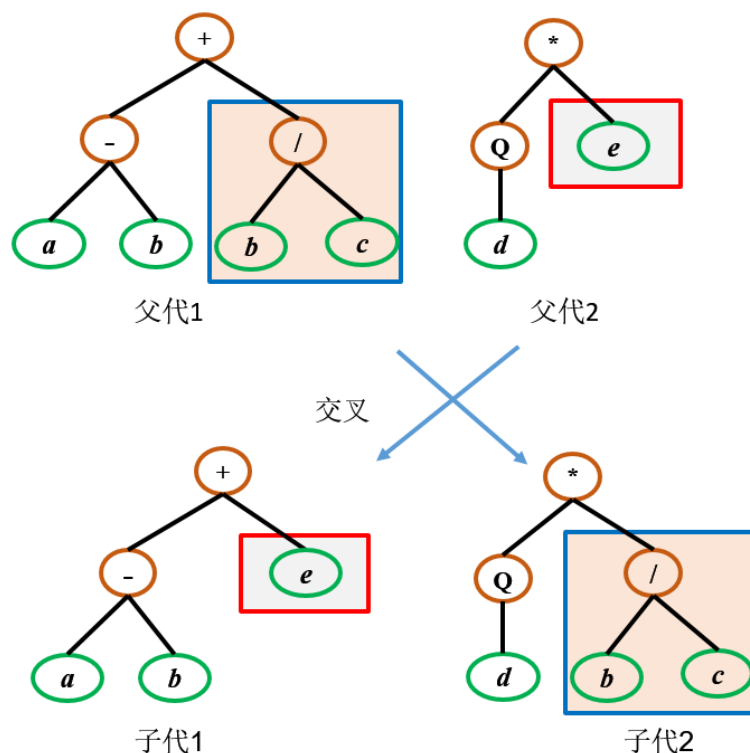


图 12. 基因表达式编程中交叉操作

而变异操作则如图 13，是将其中的某一个子树或者某一个节点（包括叶子）随机变成另一个节点或者叶节点，从而产生一个新的个体。图 13 中是将个体的右子树中的节点“Q”，替换为节点“/”，产生出一个子代。

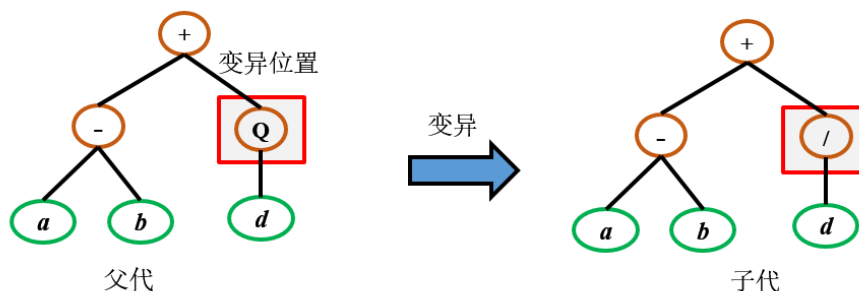


图 13 基因表达式编程中的变异操作

在树编码形式的个体之间进行遗传操作时，特别是进行交叉和变异操作时，在产生新个体后，需要检验新个体是否畸形，即检验新个体的有效性。比如在交叉变异后，特别是开根号操作和除法运算时有可能产生对负数开根号，或者除数为 0 的情况，在新个体出现畸形时，当即淘汰（剔除）该个体，重新生成。

总的来说，基因编程强大有效的全局搜索能力，能解决比较复杂的问题。但是其编码方式为复杂的树编码，所以在运行效率上较慢。

3.2.3 基因表达式编程

基因表达式编程是葡萄牙学者 Candida Ferreira^{[23][24]}于 2001 年首次提出，是一种基于生物基因结构和功能发明的一种新型自适应演化算法。基因表达式编程是从遗传算法和基因编程中发展而来，它在吸收二者优点的同时，又克服了二者的不足之处。其显著特点就是可以利用简单的编码来解决复杂的问题。

总体来说，基因表达上编程算法总共分三步：编码，解码，遗传操作。

1. 编码：

GEP 中个体的编码是实行有些类似于遗传算法中的串编码，有一串字符组成的字符串。其中的字符包括函数符号和终结符号。其中的函数符号比如：+，-，*，/，sin, cos, exp 等运算符号，而终结符号包括常数比如 a, b, c, d 等数值型常数，例如一个基因编码为：Q*+abcde（其中 Q 为开二次根号）。

个体中的基因包括头部和尾部两部分。头部中既可以含有函数符号，也可以含有终结符，而尾部只能含有终结符。基因头部的长度 h 可以根据具体情况而设置，而尾部的长度 t 则是由头部长度 h 和函数符号集中的最大操作数 n 决定，函数符号的操作数由函数本身决定，比如“+”，该函数符号具有两个操作数才能完成运算，例如 a+b。而“sin”函数符号具有的操作数就是 1 个，比如 sin(x)。基因的头部长度和函数符号集确定后，基因的总长度就确定下来了，其中尾部的长度 $t=h*(n-1)+1$ ，h 为头部的长度，n 为函数符号集中的符号的最大操作数。

比如：我们确定一个基因的头部长度为 4，函数符号集为 {Q, +, -, *, /}，终结符为 {a, b, c, d, e}。此时我们从函数符号集中可以看出最大操作数 n=2，则基因的尾部长度为 $t=h*(n-1)+1=4*(2-1)+1=5$ ，从而我们得知基因的总长度为头部长度+尾部长度为 9。根据编码规则，基因的头部可以有函数符和终结符，基因的尾部只能含有终结符，我们可以产生一个基因：

- a) 随机在函数集和终结符中抽取（可重复）4 个字符，依次排列成为基因的头部
- b) 随机在终结符中抽取（可重复）5 个字符，依次排列在头部的后面。

这样一个基因就产生了，比如我们抽取的四个头部依次为：+ a Q *，尾部依次为：c b d a e，则整个基因即为：+ a Q * c b d a e

然而，基因产生了，从这一串字符串上，我们看不出该基因的作用，就好

比我们人类的基因一样，我们无法从一串 DNA 上判断出哪里是我们的哪个部位一样。这就需要对基因进行解码。

2. 解码：

在上一步中，我们产生了一个基因： $+ a Q * c b d a e$ 。GEP 算法总是按照“层次遍历建树，后续遍历求值”的方式进行解码，

比如，我们将上基因解码成下编码树：

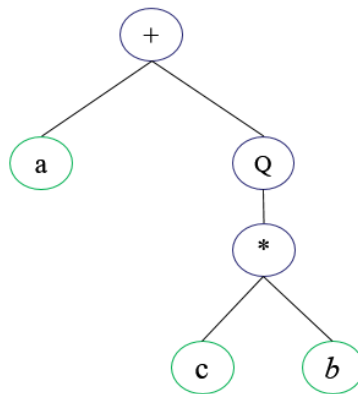


图 14. GEP 解码的编码树

解码后，可以看到上面基因中的后三个 $d a e$ 在该编码树种并未看到。这是因为每一个基因都有一个有效长度，基因的解码仅在有效长度范围内有效，基因的有效长度由基因的头部决定。在有效长度之外的基因均无效，即不在编码树中体现。

然而，该树编码解析成数学表达式为： $a + \sqrt{c * b}$ 。如果每次都按照层次遍历建树的方式进行解码，那么这种显式建树的方式，显然会拉低算法的运算效率。由于其编码串在编码规则限制范围内变动总有效，姜大志^[25]提出了一个称为基因阅读运算器（gene read & computer machine, GRCM）的个体适应值计算方法。该方法只需要对基因进行阅读遍历，不需要层次化构造树，就能实现对基因适应值的计算。该算法在运算的时候只计算基因的有效部分，以达到快速计算的效果。GRCM 算法的伪代码如下表：

表 1. GRCM 解码伪代码

第一步：计算基因的有效长度

第二步：在有效长度中找到最后一个非终结符，用指针 p 指向该位置

从非终结符开始运算：

if (p 指向的是单目运算符) then

从有效基因段尾部选取一位，赋予该运算符运算，将运算结果赋予该非终结符位置，并修改该基因位的属性为终结符；

基因的有效长度减1；

$p--$;

end if

if (p 指向的是双目运算符) then

从有效基因段尾部选取两位，赋予该运算符运算，将运算结果赋予该非终结符位置，并修改该基因位的属性为终结符；

基因的有效长度减2；

$p--$;

end if

if (p 指向的是终结符) then

$p--$;

end if

第三步：如果基因的有效长度为1，则停止运算，该终结符就保存着该基因的适应度值，如果不为1，则转向第二步。

也就是说，首先，我们需要计算该基因的有效长度，然后从有效长度中的最后一个非终结符开始运算，逐步运算与替代函数符，直至基因的有效长度为1，则停止计算，该非终结符就是基因的适应值。

3. 适应值函数：

所有的演化算法都需要对代表问题解答的个体进行一个环境适应能力的评估。个体的适应度是度量个体对环境适应能力强弱的指标，在选择个体进行遗传操作时，就是根据个体的适应值进行轮盘赌选择。一个个体的适应值函数的计算需要以下3步：

1. 对个体的编码串进行解码。比如，我们对上面编码中产生的个体进行解码成一个编码树。
2. 由解码后的个体计算出对应个体的函数值。在上面的例子中就相当于求出数遍码解析成数学表达式的值。
3. 根据优化问题的类型，由目标函数值按照一定的规则求出个体的适应度函数。假如上面问题的类型是一个符号回归的问题，那么该个体解析成数学表达式的值与训练样本的真实值之间的误差就可以作为适应值函数。比如：

T 为训练集集合，里面有 m 条数据， y_i 表示训练集中第 i 条数据的观测值， $\hat{y}_i$ 表示第 i 条数据在数学表达式上的值，即拟合值，那么该问题的适应值函数可以设置为如下公式：

$$fitness = \sum_{i=1}^m (M - |y_i - \hat{y}_i|) \quad (11)$$

如果问题是一个分类问题，那么个体在分类训练集上的准确率就可以作为适应值函数。总之，适应值函数的设置是根据具体的问题而定。

4. 遗传操作：

由于 GEP 采用了线性等长字符串编码，所以其遗传操作类似于遗传算法中的遗传操作，只要在进行遗传操作时满足“基因的长度不变，尾部只能出现终结符”，那么产生出来的新基因即为有效。所以在基因表达式中的遗传操作非常灵活，简单。

GEP 的基本遗传操作算子有选择，变异，倒串，插串，跟插串，单点重组，两点重组和基因重组 8 种^[26]。这些遗传算子使得种群的活力增强，避免种群出现早熟的现象。

1. 选择：

根据个体的适应度，进行选择父代个体进行遗传操作。在进行选择的时候采用“轮盘赌”的方法，适应值高的个体在选择的过程中选中的概率越大。也就是说适应值越高的个体越有可能作为父代进行遗传操作产生子代。其中轮盘赌的模型如下图：

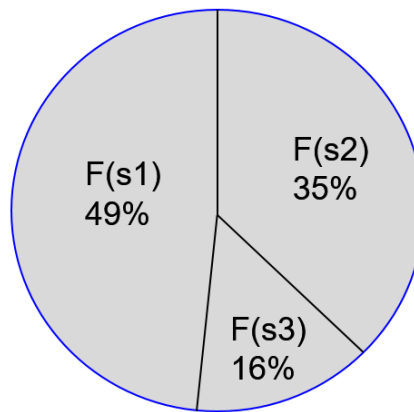


图 15. 轮盘赌模型

这里假如有三个个体 s_1, s_2, s_3 ，假如第一个个体占有所有个体总适应值的 35%，第二个个体占有总适应值的 16%，第三个个体占有总适应值的 49%。运用轮盘赌选择个体的时候，就产生一个 0-1 之间的随机数 x ，如果 x 在区间 $[0, 0.35)$ 中，则选择第一个个体；如果 x 在区间 $[0.35, 0.51)$ ，则选择第二个个体；如果 x 在区间 $[0.51, 1]$ 之间，则选择第三个个体作为父代参加到遗传操作中产生子代。

2. 变异：

变异作用在基因上，随机选取基因上的一位，重新产生该位置上的编码。如果变异位置在头部，那么可以再所有符号中重新选择一个，如果变异位置在尾部，那么久在终结符中选择一个，替换该位置。如下图，该基因发生变异的位置为第四个画横线的编码，由编码 b 变异为 a 。



图 16. GEP 中基因变异

3. 倒串：

倒串操作作用在基因的头部，它随机在基因的头部选择一段基因，然后使该基因段以其中间字符为中心，各字符顺序位置对调。下图演示了基因倒串的情况，在父代中选择了一段基因粗体下划线段，经过倒串之后产生的新子代，粗体下划线段与父代中顺序对调。



图 17. GEP 中基因倒串

4. 插串算子：

插串是基因表达式编程算法中所特有的遗传算子。它随机地在基因中选择一段子串，然后将该子串插入头部随机指定的一个位置（但不能是第一个位置），将头部的其他符号依次向后顺延，超过头部长度的编码将被截去。下图演示了父代基因经过插串操作后产生子代的过程。它选择了父代中粗体下划线段的基因，将其插入到头部第三个位置，父代基因中超出头部的编码被截去。

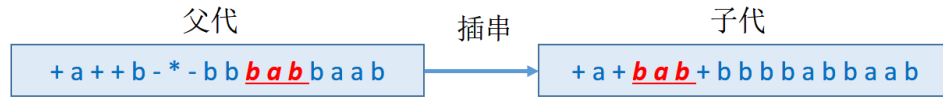


图 18. GEP 中插串操作

5. 跟插串算子:

在上面的插串操作中, 算子不允许将选择的子串插入到基因的第一个位置, 跟插串算子则是专门将选择的子串插入到第一个位置。但是这里对基因串的选择是有要求的: 从头部随机选择一个位置开始扫描, 找到第一个函数符, 然后从该位置起选择一段子串, 将该子串插入到基因的第二个位置, 头部编码依次后移, 超出头部的部分被截去。如果扫描的过程中, 遇不到函数符号, 则不做任何事情。这样做的目的就是为了保证基因的第二个位置一定是一个函数操作符号。如下图演示了一个基因经过跟插串后产生子代的过程。图中选择了粗体下划线部分的子串, 然后跟插串到基因的头部第二个位置, 其余部分后移, 超出部分截去。

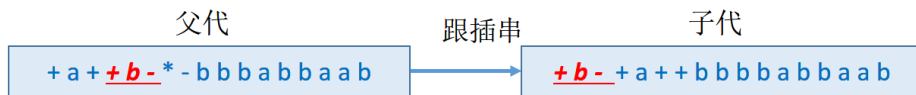


图 19. GEP 遗传操作中的跟插串

6. 单点重组算子:

单点重组算子类似于遗传算法中的交叉操作。单点重组发生在两个父代基因上, 随机选择一个交叉位置, 互换交叉点后面的基因部分, 得到两个子代基因。下图中显示了单点重组的发生过程, 选择交叉的位置为基因的第五个位置, 第五个位置后的基因两个父代互换, 从而产生出两个子代。

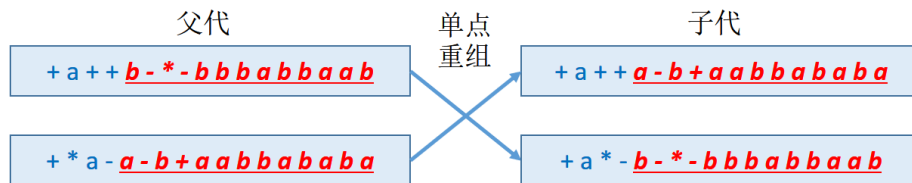


图 20. GEP 单点重组算子

7. 两点重组算子:

两点重组算子和单点重组算子有些类似, 也是作用在两个基因上, 它在两个父代基因上随机选择两个交叉点, 然后互换两个交叉点之间的基因段。如下图, 演示了两个父代染色体进行两点重组操作产生两个子代的过程。

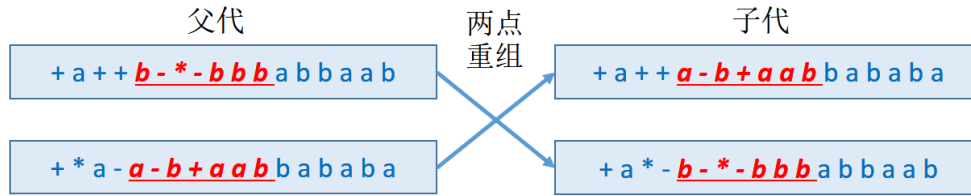


图 21. GEP 两点重组操作

8. 基因重组算子:

上面我们所说的 GEP 的遗传操作均在单个基因上进行，然而，一条染色体上也许不止一个基因，当一条染色体上存在着多条基因的情况下，就可能发生基因重组。它也是发生在两条染色体上，它是指随机选择一条染色体上的一个基因，然后同另一个父代相同位置的基因互换重组的过程。

很显然，这些算子的使用有很少的限制，得到的子代基因仍然符合 GEP 基因的定义，当然也能够解码成一个完整的编码树。经过这些遗传算子的作用，大大增加了种群的多样性。能够更大程度的避免陷入局部最优点。

基因表达式编程的流程图与上面的基因编程和遗传算法有些类似，如下图：

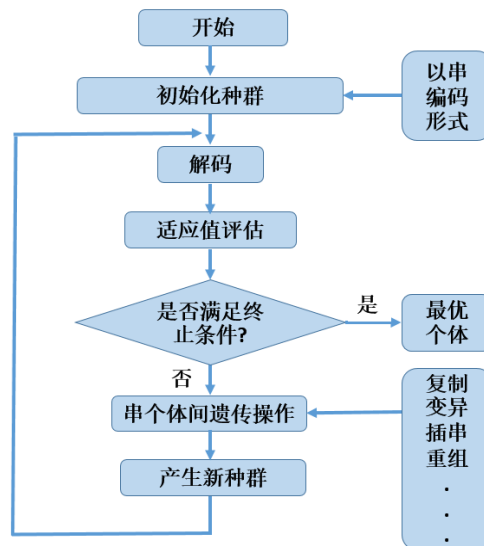


图 22. GEP 流程图

从流程图上可以看出，基因表达式编程与遗传算法和基因编程算法的最大区别就是 GEP 需要对基因个体进行解码和丰富多彩的遗传操作。

3.3 随机森林算法

随机森林一个已被证明了的成功的集成分类器，特别是用在多维分类问题上更是体现

出其强大之处^{[10][22]}。一个随机森林是一个决策树的集合，可以看作是一个分类器包括很多不同的决策树^[22]。整个算法包括三部分：特征和数据的分组，训练决策树，最后的结果投票

3.3.1 随机森林的分组策略

为了保持在随机森林中每个决策树的差异性，选择在生成决策树的时候选择不同特征集在不同的数据集上进行训练，生成最终的决策树。因此，我们需要对数据集和特征集进行分组，在分组的过程中，分别对数据集的分组和对特征集的分组。

在分组的过程中，采用基于 Bootstrapping Aggregation (Bagging)^[27] 分组策略的改进方法。在 Bagging 分组策略的方法思路如下：

1. 首先对个数为 n 的样本集通过重采样（有放回的采样）进行分组，每组大小为 n 个，分成 m 组。这样相当于是 m 个大小为 n 的样本集。
2. 在所有属性上，分别对 m 个样本集进行分类模型训练。这样训练完成后就得到 m 个分类器。
3. 利用 m 个分类器对数据进行分类。根据 m 个分类器的投票结果决定数据最终归结为哪一类。

然而，随机森林为了更能体现出其随机性，在 Bagging 分组策略的基础上进行了一些改进：不仅对样本集进行分类，还对特征集进行分类。具体如下：

1. 首先对个数为 n 的样本集通过重采样（有放回的采样）进行分组，每组大小为 n 个，分成 m 组。这样相当于是 m 个大小为 n 的样本集。
2. 对属性进行分组，属性的个数为 t ，在所有属性上无放回的随机选取 k 个属性 ($k < t$)，重复 m 次，这样共选取 m 组，大小为 k 的特征集。
3. 在 m 组属性集和 m 组样本集中，利用一组属性集在一组样本集上训练得到一个决策树分类器。这样训练完成后就得到 m 个分类器。
4. 利用 m 个分类器对数据进行分类。根据 m 个分类器的投票结果决定数据最终归结为哪一类。

首先通过重采样对数据集进行分组，将数据集分为 t 组。如图 24 所示：

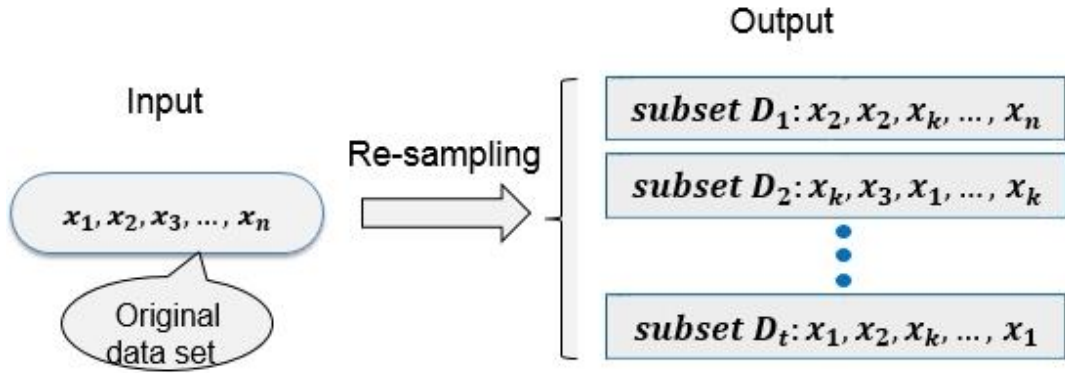


图 23. 对数据集进行分组

在该分组过程中，输入是原始的数据集，输出为分好组的子数据集 D_i ，其中子数据集的长度是和原始数据集的长度相同，值得注意的是为了保持数据集大小的一致和数据的多样性，这里采用重采样，所以在每一个子数据集中，允许有重复的样本出现。

对数据集进行分组后，对特征集进行分组，如图 25 所示：

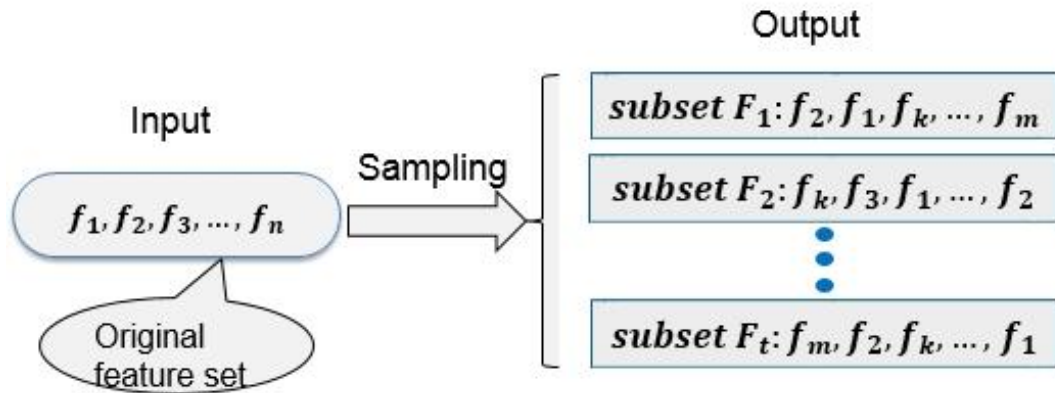


图 24. 对特征集的分组

对特征集的分组，如上图所示，输入是原始全部 77 个特征集。通过采样得到 t 组 F_i ，在对特征集进行分组的时候，是通过直接采样进行的，所以每个子特征集中的特征不会重复。这里特征集的长度应比原始特征集的长度小，如果等于原始特征集的长度，则在子特征集中特征不重复的情况下，每个子特征都一样，缺少了子特征集的多样性，从而限制了随机森林中，决策树的多样性。。

3.3.2 决策树的建立

决策树的建立，主要分两步：构建完全决策树和决策树剪枝。首先是通过一定的方法选择决策树节点，构建出一个未剪枝的决策树。然后，对该决策树进行剪枝来提高其分类的泛华能力。在构建完全决策树的时候节点的选择通常有三种方法，分别为 ID3 算法中所

用的信息增益方法，C4.5 中所用的信息增益率方法和 CART 中所用的 Gini 指标的方法：

1. ID3 算法构建完全决策树的信息增益法^[28]是一种利用贪心算法构建决策树，以信息熵的下降速度为选取测试属性的标准，即在每个节点选取还尚未被用来划分的具有最高信息增益的属性作为划分标准，然后继续这个过程，直到生成的决策树能完美分类训练样例。

ID3 构建决策树的规则是基于信息增益的原则，它选择具有最高信息增益的属性来作为节点 N 的分裂属性。该属性使结果划分中的元组分类所需信息量最小。对数据集 D 中的元组分类所需要期望信息计算如下：

$$Info(D) = - \sum_{i=1}^m p_i \log_2(p_i) \quad (12)$$

$Info(D)$ 又称为熵，其中 p_i 为第 i 类的概率值。现在假定按照属性 A 将数据集 D 划分为 v 个不同的类。在该划分之后，为了得到准确的分类还需要的信息由下面的式子度量：

$$Info_A(D) = \sum_{j=1}^v \frac{|D_j|}{|D|} \times Info(D_j) \quad (13)$$

信息增益定义为原来的信息需求（即仅基于类比例）与新需求（即对 A 划分之后得到的）之间的差，即

$$Gain(A) = Info(D) - Info_A(D) \quad (14)$$

一般来说，对于一个具有多个属性的元组，用一个属性将其完全分开几乎是不可能的。否则的话，决策树的深度就只能是 2 了，从这里可以看出，一旦我们选择一个属性 A，假设将元组分成了两个部分 A1 和 A2，由于 A1 和 A2 还可以用其它属性接着再分，所以又引出一个新的问题：接下来我们要选择哪个属性来分类？对 D 中元组分类所需的期望信息是 $Info(D)$ ，那么同理，当我们通过 A 将 D 划分成 v 个子集 D_j ($j=1, 2, \dots, v$) 之后，我们要对 D_j 的元组进行分类，需要的期望信息就是 $Info(D_j)$ ，而一共有 v 个类，所以对 v 个集合再分类，需要的信息就是公式 (13) 了。由此可知，如果公式 (13) 越小，就意味着我们接下来对 A 分出来的几个集合再进行分类所需要的信息就越小，而对于给定的训练集，实际上 $Info(D)$ 已经固定了，所以选择信息增益最大的属性作为分裂点。

但是使用信息增益的话有一个缺点，就是他偏向于具有大量值的属性。就是说在训练集中，某个属性所取的不同值的个数越多，那么越有可能拿它来作为分裂属性。例如一个训练集中有 10 个元组，对于某一个属性 A，它分别取 1-10 这十个数，如果对 A 进行分裂将会分成 10 个类，那么对于每一个类 $Info(D_j)=0$ ，从而式 (13) 为 0，该属性划分所得到的信息增益 (14) 最大，但是很显然，这种划分没有意义。

2. C4.5 算法构建完全决策树的信息增益率法。

正是因为基于信息增益的偏向具有大量知道属性的缺点，Quinlan^[29]提出利用信息增益率作为节点的选择。信息增益率使用“分裂信息”值将信息增益规范化。分类信息类似于 $Info(D)$ ，定义如下：

$$SplitInfo_A(D) = \sum_{j=1}^v \frac{|D_j|}{|D|} \times \log_2\left(\frac{|D_j|}{|D|}\right) \quad (15)$$

其中 j 表示属性 A 的第 j 个取值， $|D_j|$ 表示在 A 属性中取 j 值的样本个数， $|D|$ 表示样本集的个数。这个值表示通过将训练数据集 D 划分成对应于属性 A 测试的 v 个输出的 v 个划分产生的信息。信息增益率定义：

$$GainRatio(A) = \frac{Gain(A)}{SplitInfo(A)} \quad (16)$$

其中 $Gain(A)$ 是指上述公式几中的信息增益。这样，对每个属性进行计算器信息增益率，C4.5 通过选择具有最大增益率的属性作为节点分裂属性。

3. CART 算法构建完全决策树的 *Gini* 方法。

CART 算法利用 *Gini* 指标进行分裂节点的选择。*Gini* 指标度量数据划分或训练数据集 D 的不纯度，其中特征 A 的取值 A_i 的不纯度定义为：

$$Gini(A_i) = 1 - \sum_{j=1}^m p_j^2 \quad (17)$$

其中 i 表示取值 A_i 时样本的类别， j 为该取值下的类别，则特征 A 的不纯度表示为：

$$Gini(A) = \sum_{i=1}^n p_i * \left(1 - \sum_{j=1}^k p_j^2 \right) \quad (18)$$

其中 i 表示特征 A 所取的值， j 表示所分的类别。

选择不纯度最小的属性作为节点，进行决策树的构建。

这里我们给出一个小例子，来进行三种方法选择第一个节点，样本集是一个根据天气，温度，湿度和风速来判断是否举办活动的数据集，如下表：

表 2 构建决策树数据集

序号	天气	温度	湿度	风速	活动
1	晴	炎热	高	弱	取消
2	晴	炎热	高	强	取消
3	阴	炎热	高	弱	进行
4	雨	适中	高	弱	进行
5	雨	寒冷	正常	弱	进行
6	雨	寒冷	正常	强	取消
7	阴	寒冷	正常	强	进行
8	晴	适中	高	弱	取消
9	晴	寒冷	正常	弱	进行
10	雨	适中	正常	弱	进行
11	晴	适中	正常	强	进行
12	阴	适中	高	强	进行
13	阴	炎热	正常	弱	进行
14	雨	适中	高	强	取消

从上表中，我们可以看出，数据集有 4 个属性，即属性集为 $A=\{\text{天气，温度，湿度，风速}\}$ ，而标签集有两类 $\{\text{进行，取消}\}$ 。

数据集 D 包含 14 个训练样本，其中属于“进行”的样本有 9 个，属于“取消”的样本有 5 个。则计算数据集 D 的信息熵：按照公式 12 有：

$$\text{Info}(D) = -9/14 * \log_2(9/14) - 5/14 * \log_2(5/14) = 0.940$$

下面对属性集中的每个属性进行计算其信息熵，如下所示：

$$1 \quad \text{Info(天气)} = 5/14 * [- 2/5 * \log_2(2/5) - 3/5 * \log_2(3/5)] + 4/14 * [- 4/4 * \log_2(4/4) - 0/4 * \log_2(0/4)] + 5/14 * [- 3/5 * \log_2(3/5) - 2/5 * \log_2(2/5)] = 0.694$$

$$2 \quad \text{Info(温度)} = 4/14 * [- 2/4 * \log_2(2/4) - 2/4 * \log_2(2/4)] + 6/14 * [- 4/6 * \log_2(4/6) - 2/6 * \log_2(2/6)] + 4/14 * [- 3/4 * \log_2(3/4) - 1/4 * \log_2(1/4)] = 0.911$$

$$3 \quad \text{Info(湿度)} = 7/14 * [- 3/7 * \log_2(3/7) - 4/7 * \log_2(4/7)] + 7/14 * [- 6/7 * \log_2(6/7) - 1/7 * \log_2(1/7)] = 0.789$$

$$4 \quad \text{Info(风速)} = 6/14 * [- 3/6 * \log_2(3/6) - 3/6 * \log_2(3/6)] + 8/14 * [- 6/8 * \log_2(6/8) - 2/8 * \log_2(2/8)] = 0.892$$

根据上面的数据，我们可以计算第一个根节点所依赖的信息增益值，计算如下：

$$1 \quad \text{Gain(天气)} = \text{Info(D)} - \text{Info(天气)} = 0.940 - 0.694 = 0.246$$

$$2 \quad \text{Gain(温度)} = \text{Info(D)} - \text{Info(温度)} = 0.940 - 0.911 = 0.029$$

$$3 \quad \text{Gain(湿度)} = \text{Info(D)} - \text{Info(湿度)} = 0.940 - 0.789 = 0.151$$

$$4 \quad \text{Gain(风速)} = \text{Info(D)} - \text{Info(风速)} = 0.940 - 0.892 = 0.048$$

至此，我们如果按照 ID3 算法中采用的信息增益值进行节点选择的话，我们的根节点将会是湿度，因为湿度的信息增益值最大。

为了计算信息增益率，我们计算公式 15 的分裂信息度量 Splitinfo. 记为 $H(V)$ ：

在属性天气中，有 3 个取值，其中晴有 5 个样本，雨有 5 个样本，阴有 4 个样本，所以，我们有：

$$H(\text{天气}) = - 5/14 * \log_2(5/14) - 5/14 * \log_2(5/14) - 4/14 * \log_2(4/14) = 1.577$$

在属性温度中，有 3 个取值，其中炎热有 4 个样本，适中有 6 个样本，寒冷有 4 个样本，所以，我们有：

$$H(\text{温度}) = - 4/14 * \log_2(4/14) - 6/14 * \log_2(6/14) - 4/14 * \log_2(4/14) = 1.557$$

在属性湿度中，有 2 个取值，其中正常有 7 个样本，高湿度有 7 个样本，所以，我们有：

$$H(\text{湿度}) = - 7/14 * \log_2(7/14) - 7/14 * \log_2(7/14) = 1.0$$

在属性风速中，有 2 个取值，其中强有 6 个样本，弱有 8 个样本，所以，我们有：

$$H(\text{风速}) = -6/14 * \log_2(6/14) - 8/14 * \log_2(8/14) = 0.985$$

然后，我们根据公式 16 计算每个属性的信息增益率，记为 IGR。有如下结果：

$$IGR(\text{天气}) = \text{Info}(\text{天气}) / H(\text{天气}) = 0.246/1.577 = 0.156$$

$$IGR(\text{温度}) = \text{Info}(\text{温度}) / H(\text{温度}) = 0.029 / 1.557 = 0.019$$

$$IGR(\text{湿度}) = \text{Info}(\text{湿度}) / H(\text{湿度}) = 0.151/1.0 = 0.151$$

$$IGR(\text{风速}) = \text{Info}(\text{风速}) / H(\text{风速}) = 0.048/0.985 = 0.049$$

如果我们根据 C4.5 算法中的信息增益率算法进行选择决策树的节点，对该节点进行分裂。从上面的 IGR 可以知道属性天气的信息增益率最大，所以我们选择天气作为其根节点。

现在，我们计算每个属性的 Gini 指标：

$$\begin{aligned} \text{Gini}(\text{天气}) &= \frac{5}{14} \left(1 - \left(\frac{2}{5} \right)^2 - \left(\frac{3}{5} \right)^2 \right) + \frac{5}{14} \left(1 - \left(\frac{2}{5} \right)^2 - \left(\frac{3}{5} \right)^2 \right) + \frac{4}{14} \left(1 - \left(\frac{4}{4} \right)^2 - \left(\frac{0}{4} \right)^2 \right) \\ &= 0.3429 \end{aligned}$$

$$\begin{aligned} \text{Gini}(\text{温度}) &= \frac{4}{14} \left(1 - \left(\frac{2}{4} \right)^2 - \left(\frac{2}{4} \right)^2 \right) + \frac{6}{14} \left(1 - \left(\frac{4}{6} \right)^2 - \left(\frac{2}{6} \right)^2 \right) + \frac{4}{14} \left(1 - \left(\frac{3}{4} \right)^2 - \left(\frac{1}{4} \right)^2 \right) \\ &= 0.4405 \end{aligned}$$

$$\text{Gini}(\text{湿度}) = \frac{7}{14} \left(1 - \left(\frac{4}{7} \right)^2 - \left(\frac{3}{7} \right)^2 \right) + \frac{7}{14} \left(1 - \left(\frac{6}{7} \right)^2 - \left(\frac{1}{7} \right)^2 \right) = 0.3673$$

$$\text{Gini}(\text{风速}) = \frac{8}{14} \left(1 - \left(\frac{2}{8} \right)^2 - \left(\frac{6}{8} \right)^2 \right) + \frac{6}{14} \left(1 - \left(\frac{3}{6} \right)^2 - \left(\frac{3}{6} \right)^2 \right) = 0.4286$$

从四个属性的 **Gini** 指标上，我们可以看出，在 CART 算法选择 Gini 指标最小的属性作为根节点，此时，选择天气作为根节点进行分裂。

当决策树构建完全之后，由于数据中的噪声点和离群点。许多分枝反应的是训练数据集中的异常。剪枝方法是用来处理构建决策树后的这种过拟合的问题。通常的剪枝方法都是使用统计度量的方法，即剪去最不可靠的分枝。

剪枝一般分为两种方法：先剪枝和后剪枝

先剪枝的方法是通过提前停止树的构造（比如决定在某个节点不再分裂）而对树剪枝，一旦停止，这个节点就编程树叶，该树叶可能取它持有的子集最频繁类作为自己的类。先剪枝有很多方法，比如当决策树达到一定的高度就停止决策树的生长；或者到达此节点的实例个数小于某个阈值的时候也可以停止树的生长，不足之处是不能处理那些数据量比较小的特殊情况。先剪枝有个缺点就是视野效果问题，也就是说我们很难确定基于先剪枝生成的决策树是否是最优的决策树。

更常用的是第二种方法后剪枝。它由完全成长的树剪去子树而形成，通过删除节点的分枝并用树叶来替换它。树叶一般用子树中最频繁的类别来标记。这个方法称为基于误判的剪枝。这个思路很直接，因为完全决策树过拟合，所以通过一个测试数据集来纠正它。对于完全决策树中的每一个非叶子节点的子树，我们尝试着把它替换成一个叶子节点，该叶子节点的类别我们用子树所覆盖训练样本中存在最多的那个类来代替，这样就产生了一个简化决策树，然后比较这两个决策树在测试数据集中的表现，如果简化决策树在测试数据集中的错误比较少，并且该子树里面没有包含另外一个具有类似特性的子树（所谓类似的特性，指的就是把子树替换成叶子节点后，其测试数据集误判率降低的特性），那么该子树就可以替换成叶子节点。该算法以 bottom-up 的方式遍历所有的子树，直至没有任何子树可以替换使得测试数据集的表现得以改进时，算法就可以终止^[30]。

在我们的随机森林算法构建决策树的时候，就是在对数据集和特征集分组之后，将子特征集和子数据集分组进行训练分类模型：

$$G = \{G_1, G_2, \dots, G_i, \dots, G_t\}, \text{ where } G_i = \{D_i, F_i\} \quad (19)$$

在传统的随机森林算法中，决策树的建立是通过 CART 算法进行训练决策树模型。CART 算法的基本原理是通过由测试变量和目标变量构成的训练数据集的循环分析，而构成的二叉树形式的决策树^[23]。CART 算法在最佳测试变量和分割阈值准则上采用经济学中的基尼系数 *Gini*，*Gini* 系数的定义上。通过 CART 的方法对每组 G_i 进行构建决策树。这样我们就可以构建 t 颗决策树。

3.3.3 投票机制

投票机制是指根据投票者的选择以选出最终结果的一种常用方法。其中主要分以下几种：

1. 一票否决制（一致表决）：一票否决制是指在投票的过程中，只有在大家都同意 A

结果的情况下，最终的结果才为 A，任何一票不同意，则最后的结果均不为 A。

2. 少数服从多数制：是指在投票的过程中，最终的结果是大多数人认可的结果，少数人须服从多数人的意愿。
3. 阈值表决制：在投票的时候，为每个类设置阈值，当投票结果达到某一类的阈值时，即选取该类为最终结果。
4. 加权投票制：即指在投票的过程中，某些人投票分量比较重，在 GEP 中，可以看做适应值高的个体投票份额较大。
5. 贝叶斯投票机制：简单的投票法假设每个分类器都是平等的，在实际生活中，我们听取一个人的意见，会考虑这个人过去的意见是否有用，从而加大或减少权值。也就是说贝叶斯投票机制是基于每个分类器在过去的分类表现设定一个权值，然后按照这个权值进行投票。

随机森林的决策树的传统方法通过使用 CART 的方法训练决策树，在根据 t 组数据和特征获得 t 棵决策树后，需要利用每一棵决策树对测试集进行决策出一个结果。这样就会对一个测试样本得到 t 个测试结果。最简单的，这里每颗决策树投一票，根据少数服从多数的原则，得到该测试样本的最终分类结果。如下图 8：

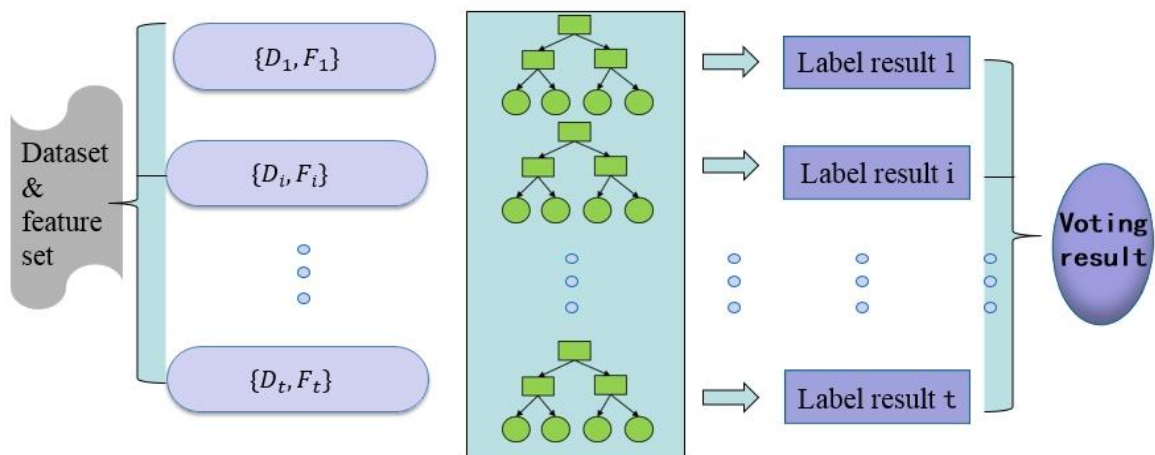


图 25 随机森林流程图

整个随机森林的流程图如上图所示。首先对数据集和特征集进行分组，然后对每一组数据集和特征集通过 CART 算法进行决策树的构建，对于一个测试样本，每一个决策树都会得出一个决策结果，最后根据少数服从多数的原则进行投票得出最终是分类结果。

3.4 GEP-RF 算法

3.4.1 决策树的建立

正如前面所讨论的，在 GEP-RF 算法中，决策树的建立采用有基因表达式编程进化出来的决策时代替传统方法由 CART 算法得到的决策树。由于决策树在 GEP 算法中是以基因串的形式展现进化，所以在进化决策树阶段，对决策树的建立也就是对基因串的建立，主要分以下两步：

1. 定义决策树参数：

如同前面 GEP 中所述，首先我们先定义基因的函数符号集： $\{Q, +, -, *, /, \sin, \cos, \tan, \cot\}$ ，终结符为 $\{f_1, f_2, f_3, f_4, \dots, f_{75}, f_{76}, f_{77}\}$ ，终结符为提取的特征集。其中决策函数的节点为数学运算符号，而叶子节点为特征集的属性值。其中基因的头部，我们能从函数符号集和终结符集中进行选取，而基因的尾部，我们能只能从终结符中进行选取。定义基因的头部长度，从基因的头部长度，我们可以计算出基因的总长度。

2. 定义过基因的一些参数之后，我们就可以通过产生随机数进行构建基因串。

此时，假如基因的头部长度为 5，可以得到基因的总长度为 11。我们可以按照规则，从函数符号集和终结符集中随机选择 5 个字符： $+ \sin * / f_2$ ，生成该基因的头部，再从终结符集中随机选取 6 个字符： $f_5 f_3 f_7 f_1 f_9 f_{11}$ ，作为基因的尾部。这样头部尾部组合到一起就产生出一个基因： $+ \sin * / f_2 f_5 f_3 f_7 f_1 f_9 f_{11}$ 这样经过基因解码（翻译成数学表达式）就能得到一个函数，我们可以根据该函数进行预测分类。可以将该基因解码成一颗编码树，例如下图：

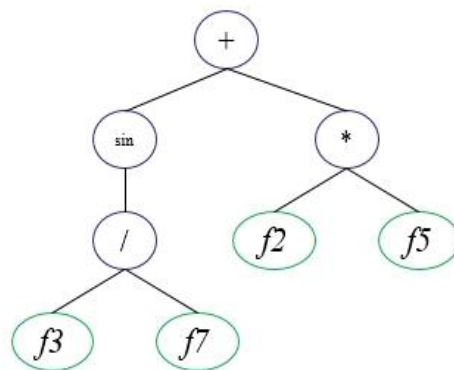


图 26 进化决策树中的一个树编码个体

在图 26 这个个体中， $+$ ， $\sin$ ， $*$ ， $/$ 是来自于符号集中的一些数学运算符号， f_2 ， f_5 ， f_3 ， f_7

是来自于特征集中的一些特征值。例如， f_2 表示特征集中的第二个特征，在该文章中，所有的特征均是以数值的形式存在。当然，如果对上图 9 进行解码，或者按照前面我们所讲的 GRCM 解码算法，则可以将其解码为下面一个函数式：

$$y = \sin\left(\frac{f_3}{f_7}\right) + f_2 * f_5 \quad (20)$$

现在，当一个新的数据输入该决策树之后，将该数据的特征 2, 3, 5, 7 带入函数 11，我们可以根据函数得到一个 y 值。至此，用户可以设置一个阈值来进行一个决策分类。比如我们设置 $y > 0$ 表示发生急性低血压，而 $y < 0$ 表示该记录未发生急性低血压。实际上，我们可以把该函数作为一个决策函数。

3.4.2 GEP-RF 算法

进化过程中的适应值函数为决策树分类的准确率，比如计算图 26 个体的适应值时，我们可以将训练集的样本输入到该决策树中，进而可以得到一个该决策树对训练集的一个准确率，通过该准确率的高低，在整个种群进化的过程中优胜劣汰。

至此，我们已经介绍完了树编码和解码，因此第一步，我们可以按照图 11 的流程图进行种群进化，首先，先初始化一个种群，种群中包含 n 个解码后类似于图 26 的个体。利用样本的训练集对该种群中的个体进行适应值评估，即计算每一个个体对训练集中的所有样本进行分类的准确率。根据轮盘赌法则，从种群中选出个体作为父代，然后让父代个体之间进行遗传操作，变异，插串，重组等遗传操作，对操作产生的新个体，再利用 GRCM 解码方式进行适应值评估，在保持种群大小不变的情况下根据适应值大小进行优胜劣汰的筛选。即淘汰适应值低的个体，选中适应值高的个体。让优秀的个体有更大的机会进行产生下一代。如此直至进化结束。

进化算法收敛的条件一般分为三种：

1. 种群进化收敛，也就是说种群中的个体再也进化不出适应值更高的个体来了，至此，进化结束。
2. 人为设定进化代数，让种群进化 m 代之后，停止进化，比如 5000 代，等根据情况而定。
3. 人为设定进化的评价次数，让种群达到个体的评价次数后，停止进化。

在这里，我们选择种群收敛和进化代数相结合的终止进化的方法：即在种群收敛和进化 m 代，二者满足其一就终止进化。

进化随机森林分类器的伪代码如下表；

表 3 GEP-RF 伪代码

GEP-RF Algorithm
Input: -t: the number of trees, -D: the training data set, -F: the feature set, -m: the size of subset of the feature -g: the generation of GEP's evolution Output: -A random forest RF Method: 1. For $i=1$ to t do 2. Grouping the data set as in Figure 6. 3. Grouping the feature set as in Figure 7. 4. $Tree(i) = \text{evolveTrees}(D_i, F_i, g);$ 5. End for 6. RF is obtained with the set of trees with the number of t Function evolveTrees() 1. Initialize the population 2. For $i=1$ to g do 3. Evaluate the fitness of individuals in population 4. Keep the best individual through elitism strategy 5. Apply genetic operations 6. Whether the population is convergence 7. End for 8. Obtain a tree from the best individual of all the generations

从该伪代码中，可以看出 GEP-RF 算法的输入是 t -随机森林中决策树的个数， D -训练数据集， F -特征集， m -子特征集的长度， g -进化的代数。算法的输出是一个随机森林分类器。首先按照图 23 对数据集进行分组，得到子数据集 D_i ，其中 D_i 的长度和原始数据集一致。然后对特征集按照图 24 进行分组，得到子特征集 F_i ，其中 F_i 的长度比原始特征集要小。分完组后利用基因表达式编程对分好组的数据集和特征集进行全局搜索出一个最优的决策树。在进化决策树的过程中，首先初始化一个种群，然后对种群中的个体进行适应值评估，通过遗传操作产生出新的个体，在保持种群大小不变的情况下实施优胜略汰精英保留策略。这里进化终止条件为达到 g 代或者收敛之后停止计划。进化完成之后便得到一个

类似于上面所说的一个基因串，可以将该基因串解码成图 26 的决策树。在整个算法运行完成之后，最终输出一个含有 t 个可解码成决策树的基因串的随机森林。

3.5 本章小结

本章主要介绍了：

1. 基因表达式编程算法，遗传算法的编码解码，基因表达式编程的编码解码，基因表达式编程与基因编程，遗传算法之间的不同之处。
2. 随机森林算法，随机森林算法如何分组，如何获得决策树，以及投票机制。
3. GEP-RF 算法，基因表达式编程算法与随机森林算法的结合，如何建立决策树以及算法的流程。

第四章 论文的实验

4.1 实验框架

论文的实验部分是基于 VS2010 平台实现的，论文的实验主要分 3 个部分，首先是对原始数据集进行 EMD 分解得到 IMF 信号，并对 IMF 信号进行特征提取获得特征集 F 。然后进行随机森林分类器模型的建立，最后利用分类器对测试集进行投票得出最后结果。

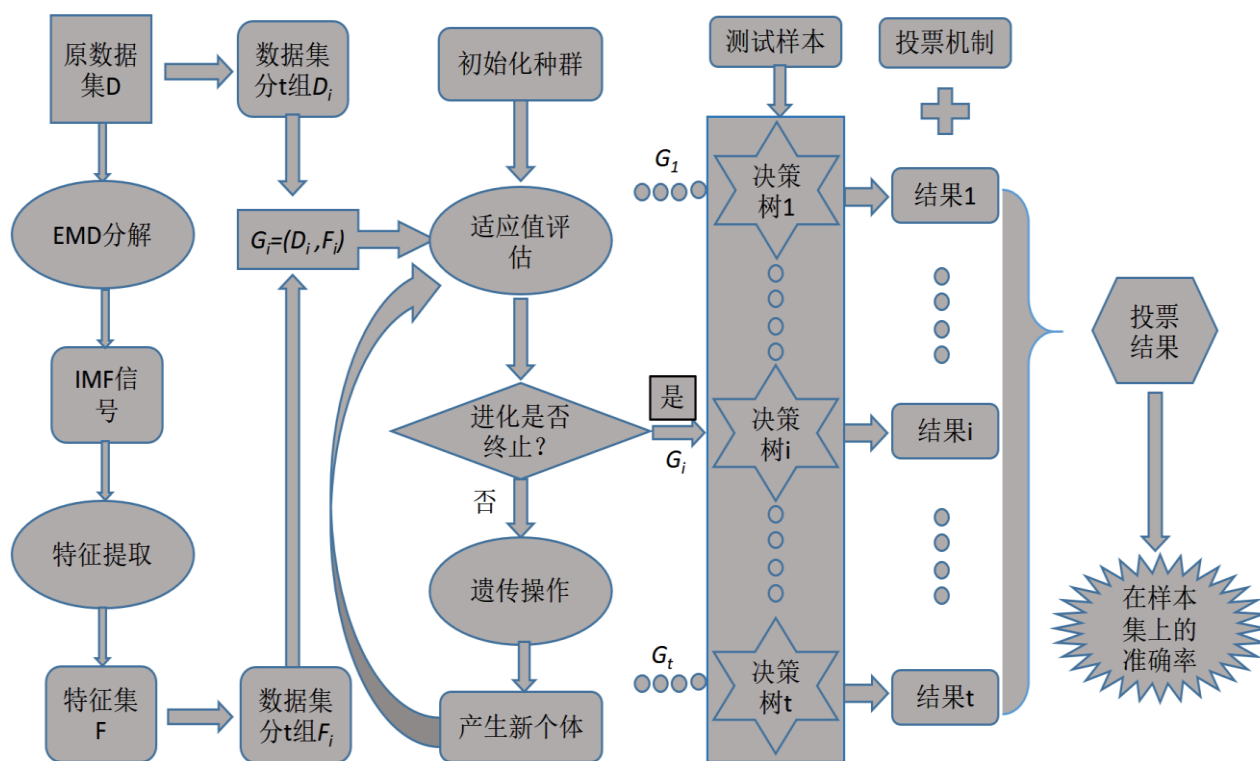


图 27 论文实验流程图

算法的最终验证采取交叉验证的方法。交叉验证是一种评估统计分析，机器学习算法对独立于训练数据的数据集的泛化能力。常见的验证算法的方法有三种：

1. 保留样本法：保留样本法是将原始数据集随机分为两组，一组用来做训练集，训练分类模型，一组作为验证集，用来验证分类模型的分类能力，利用其在验证集上的性能指标作为衡量分类器的标准。这种方法比较简单，只需将数据分为两组即可。但是这种验证方法得到的结果其实并不具有说服力，主要原因是训练样本集太少，通常不足以代表母体样本的分布。
2. K 折交叉验证：k 折交叉验证是常用的验证算法泛化能力的方法，其主要思想是将原始数据分成 k 组（一般是均匀分），然后将每个子集数据分别做一次验证集，其

余 $k-1$ 组做训练集，这样就会得到 k 个模型，用这 k 个模型最终的在各自验证集上的分类性能的平均指标作为此算法的性能指标。 K 一般大于 2. 一般来说 k 取 5 或者 10（经验参数）。 K 折交叉验证可以有效的避免过学习以及欠学习状态的发生，最后得到的结果也比较具有说服力。

3. 留一手交叉验证：其实该交叉验证法是 k 折交叉验证的一个特殊情况。就是将这个 k 值取为数据集样本的总数。假如，数据集中总共有 m 个数据，那么这个 k 就取为 m ，这样，每一次只由 1 个样本用来作为测试集，其余的 $m-1$ 个样本作为训练集来训练分类模型，同 k 折交叉验证类似，取最终 m 个模型在各自验证集上的分类性能的凭据作为此算法的性能指标。当然，该验证方法的缺点也是比较明显的，那就是计算成本过高，实现起来比较困难。

在本算法中，我们用 5 折交叉验证的方法对该算法进行泛化能力的评价，五折交叉验证的流程如下图：

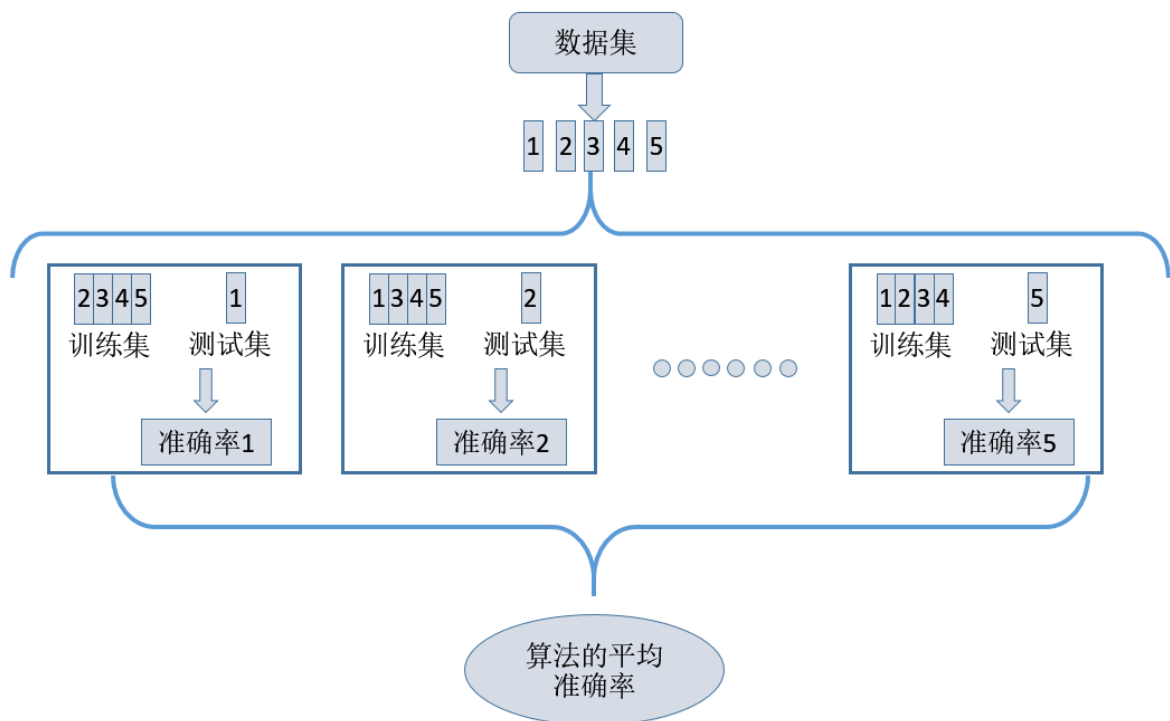


图 28. 5 折交叉验证流程图

5 折交叉验证是先将数据集分成 5 份：1, 2, 3, 4, 5 个子数据集，如上图。首先用 2, 3, 4, 5, 数据集进行训练分类模型，用 1 子数据集测试模型的分类性能，得到准确率 1；然后类似，用子数据集 2 作为测试集，其余的为训练集，得到准确率 2；依次对每一个子数据集都进行测试，其余的用来训练。最后得到的 5 次准确率求平均值即得到算法的最终平均准确率。

4.2 实验参数及结果

实验数据集来自于来自美国 PhysioNet 网站的 MIMIC-II 数据库的数据，实验中共共 1599 条数据，其中 799 条 AHE 记录的 ABP 信号，800 条正常 ABP 信号。首先将该信号转换为 MAP 信号。然后在对 MAP 进行 EMD 分解提取特征。实验中采用 5 折交叉验证来验证算法的泛化能力，随机森林中决策树的个数为 5，从 IMFs 和原信号中提取的特征 77 个，在随机森林分组阶段，子特征集的个数为 30。进化决策树的参数如下：

表 4. 决策树进化参数设置

基因个数	1	插串概率	0.1
种群大小	100	跟插串概率	0.1
头部长度的	10	单点重组概率	0.3
进化代数	1000	双点重组概率	0.3
变异概率	0.1	倒串概率	0.1
函数符号集	+, -, ×, /, sin, cos, tan, cot, exp		
终结符集	特征 f1, f2, ..., f76, f77		

实验结果以准确率 ACC，灵敏度 SE，特异性 SP 表示。其中 ACC, SE, SP 表示如下：

$$AC = \frac{TP+TN}{TP+TN+FP+FN} \quad (21)$$

$$SE = \frac{TP}{TP+FN} \quad (22)$$

$$SP = \frac{TN}{TN+FP} \quad (23)$$

其中 TP, FP, TN, FN 分别为真阳性，假阳性，真阴性，假阴性。对于一个二分类问题来说，会出现四种情况。如果一个实例是正类并且也被预测成正类，即为真正类（True positive），也成为真阳性（TP）。如果实例是负类被预测成正类，称之为假正类（False positive），也成为假阳性（FP）。相应地，如果实例是负类被预测成负类，称之为真负类（True negative），又叫真阴性（TN），正类被预测成负类则为假负类（false negative），

也称假阴性 (FN)。

我们对原始数据集进行 5 折交叉验证后, 其中训练集中的 ACC, SE, SP 的结果如下表, 其中最后一列为 5 次交叉验证最终得到的平均指标:

表 5 交叉验证在训练集的结果

Training set						
Time Precision	First validation	Second validation	Third validation	Forth validation	Fifth validation	Average validation
ACC	0.7852	0.7648	0.8016	0.7727	0.7945	0.7837
SE	0.8225	0.8146	0.8029	0.7910	0.8094	0.8081
SP	0.7553	0.7292	0.8003	0.7575	0.7798	0.7644

从上表中, 我们可以看出, 对数据集进行五折交叉验证后在训练集上的平均准确率为 78.37%, 灵敏度为 80.81%, 特异性为 76.44%。

5 折交叉验证中测试集的 ACC, SE, SP 的结果如下表, 其中最后一列为 5 次交叉验证最终得到的平均指标:

表 6 交叉验证在测试集的结果

Testing set						
Time Precision	First validation	Second validation	Third validation	Forth validation	Fifth validation	Average validation
ACC	0.7524	0.8245	0.7524	0.7962	0.7524	0.7755
SE	0.7635	0.8786	0.7607	0.8786	0.7462	0.8055
SP	0.7427	0.7821	0.7436	0.7318	0.7566	0.7514

从上面的结果上可以看出, 该算法最终在测试集上得到的最终准确率为 77.55%, 灵敏度为 80.55%, 特异性为 75.14%。

基于基因表达式编程的随机森林算法同只用基因表达式编程的方法训练决策树的方法进行比较, 以及传统用 CART 算法构建决策树的随机森林的方法比较。5 折交叉验证后训练集上的结果如下表。

表 7 本文算法与其他算法在训练集上的比较

Trainingset			
Algorithm \ Precision	GEP	GEP-RF	RF
ACC	0.7803	0.7837	0.6938
SE	0.8089	0.8081	0.6870
SP	0.7575	0.7644	0.7009

在训练集上，我们可以看到基于基因表达式编程的随机森林算法在 ACC 和 SP 指标上均高于 GEP 算法，在与随机森林算法的比较上看，三项指标均高于随机森林算法

在测试集上，GEP-RF 算法在 ACC, SE, SP 三个指标上均高于基因表达式编程和随机森林算法。如下表：

表 8 本文算法与其他算法在测试集上的比较

Testing set			
Algorithm \ Precision	GEP	GEP-RF	RF
ACC	0.7705	0.7755	0.7289
SE	0.7975	0.8055	0.7297
SP	0.7478	0.7514	0.7284

从上面结果来看，我们可以得出基于基因表达式编程的随机森林算法在本文数据集上的准确率 77.55%，灵敏度为 80.55%，特异性为 75.14%，并且高于基因表达式编程和随机森林算法。

三种方法为了有一个更好的比较，这里做了一个 T-测试，T 测试是用于小样本（样本容量小于 30）的两个平均值差异程度的检验方法。它是用 T 分布理论来推断差异发生的概率，从而判定两个平均数的差异是否显著。

其结果如表下下。

p-value of T-test result		
Algorithm \ Precision	GEP-RF & GEP	GEP-RF & RF
ACC	0.3864	0.02251
SE	0.4175	0.03923

SP	0.3876	0.1211
----	--------	--------

表 9 T-测试结果

从表 9，我们可以看出，基于基因表达式编程的随机森林在准确率和灵敏度上明显优于随机森林算法，在准确率和灵敏度上略优于基因表达式编程算法。

总的来说，本文算法基于基因表达式编程的随机森林算法在准确度，灵敏度和特异性上均高于基因表达式编程算法和传统的随机森林算法。在临床上对急性低血压的预测有着重要的指导意义。

第五章 总结与展望

在本文中，我们提出了一个新的算法叫做 GEP-RF，该算法融合了基因表达式编程和随机森林算法。这里随机森林主要用其强大的分类能力来进行分类，而基因表达式编程算法主要用其强大的全局搜索能力来训练决策树。在文章的实验部分，利用 5 折交叉验证对该算法进行了泛化能力的验证，在验证阶段得到准确率为 77.55%，灵敏度 80.55%，特异性 75.14%的结果。通过比较得出该算法优于基因表达式编程和随机森林算法在该数据集上的性能。

在接下来的工作中，会再次设置机制进行提升该算法的分类准确率以及将该算法用在图像处理领域中。

参考文献

- [1] 庞兴梅. 基于数据融合的心率估计算法及监护仪器错误报警一直的研究[D]. 山东大学硕士学位论文. 山东: 山东大学. 2008
- [2] <http://www.physionet.org/challenge/2009/>.
- [3] Teresa Rocha, Simao Paredes, “Prediction of acute hypotensive episodes by means of neural network multi-models”, *Computers in Biology and Medicine* 41 (2011) 881 – 890.
- [4] J. Bassale, “Hypotension Prediction – Arterical Blood Pressure Variability”, Technical Report, 2001.
- [5] C. Crespo, et al., “Precursors in the arterial blood pressure signal to episodes of acute hypotension in sepsis”, *Proceedings of the 16th international EURASIP Conference BIOSIGNAL*, vol. 16, 2002, pp. 206 – 208
- [6] M. Saeed, “Temporal Pattern Recognition in Multiparameter ICU Data.” Doctoral dissertation, Department of Electrical Engineering and Computer Science, MIT, Cambridge, MA, 2007
- [7] L. Lehman, M. saeed, “Similarity-based searching in multiparameter time series databases”, *Computers in Cardiology* 35(2008) 653–656
- [8] X Chen, D Xu, G Zhang, R Mukkamala, “Forecasting Acute Hypotensive Episodes in Intensive Care Patients Based on a Peripheral Arterial Blood Pressure Waveform.”, *Computers in Cardiology*, Sept. 2009, pp. 545–548.
- [9] MA Mneimneh, RJ Povinelli, “A rule-based approach for the prediction of acute hypotensive episodes,” *Computers in Cardiology*, Sept. 2009, pp. 557–560.
- [10] PA Fournier, JF Roy, “Acute hypotension episode prediction using information divergence for feature selection, and non-parametric methods for classification,” *Computers in Cardiology*, Sept. 2009, pp. 625–628.
- [11] Vaibhav Awandekar, A.N. Cheeran, “Predicting Acute Hypotensive Episode by Bhattacharyya Distance.” *International Journal of Engineering Research and Application*, 3(2):370–372, 2013.
- [12] L. Breiman, Random forests, *Mach. Learn.* 45 (2001) 5–32

- [13] Ahmad Taher Azar, et al. “A random forest classifier for lymph disease”, Computer Methods and Programs in Biomedicine 113 (2014) 465-473
- [14] 林珠, 邢延. “数据挖掘中适用于分类的时序数据特征提取方法”, 计算机系统应用, 2012 年第 21 卷第 10 期
- [15] 王志刚, 赖丽娟. “基于 AR 模型和支持向量机的记性低血压预测”, 中国生物医学工程学报, 2011 年 4 月
- [16] 陈朝虹. “基于双谱分析-BP 神经网络的 ICU 急性低血压发生预测模型研究”, 中国医学物理学杂志, 2011 年 9 月
- [17] LAI Li-juan, “Study on Predicting Method for Acute Hypotensive Episodes Based on Wavelet Transform and Support Vector Machine”. CHINESE J. DIAL. & ARTIF. ORGANS VOL. 22 NO. 1 MAR. 2011
- [18] http://baike.baidu.com/link?url=7g44ag6q5HUNeogT6y35CQjbFBkU_VYRdWdLMYoDdglRA_9AUpeXnKsPOIUrmzfwimKenV32kmaHmYtlgEHshyf0MjBPgGUnuci juVuOU1a
- [19] Abdollah Arasteh, et al, “Application of Empirical Mode Decomposition in Prediction of Acute Hypotension Episodes”, ICBME2010, 3-4 November 2010
- [20] Huang Norden E, et al., ” The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis”, Proc. R. Soc. Lond. A, 1998
- [21] <http://baike.baidu.com/view/589872.htm>
- [22] Holland J H. Adaptation in natural and artificial systems. Univ of Michigan Press, Ann Arbor Mich, 1975
- [23] FERREIRA C. Gene expression programming: a new adaptive algorithm for solving problems [J]. Complex Systems, 2001, 13 (2) : 87-129
- [24] FERREIRA C. Gene expression programming[M]. Portugal: Angora do Heroismo, 2002.
- [25] 姜大志, 吴志健, 康立山等. 基因表达式程序设计的 GCRM 方法. 系统仿真学报, 2006, 6 (18): 1466-1468
- [26] 元昌安, 彭昱忠, 覃晓等. 基因表达式编程算法原理与应用, 北京. 科学出版社, 2010

- [27] Breiman, Leo. "Bagging predictors". Machine Learning 1996.24 (2): 123 - 140.
- [28] <http://baike.baidu.com/link?url=qEukzfVuZ5rUN209nS5M69noAWsiA4gYiQqkFk1ReKWzq1fbklrZxz2YYigIkPm4YtW0Kx1uY2HHFt1EtJRkKK>
- [29] Quinlan, J. R. C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, 1993.
- [30] (美)吴信东, 库马尔. (译者: 李文波) 数据挖掘十大算法. 清华大学出版社, 2013-5

致谢

时光荏苒，岁月如梭。转眼就到了跟我的研究生生涯说再见的季节。回忆起这三年来在学习和生活中的点点滴滴，感触良多。对于这里的一切我感到不舍与难忘。在这段美好时光里，我得到很多老师的点拨、同学的帮助以及家人的关怀与支持。我对于他们表示深深的感激与感谢。

首先，我要感谢我的导师范衡教授，副导师姜大志副教授，导师们不仅为我提供了优秀的学习环境和条件，更重要的是老师严谨的治学态度，脚踏实地的工作作风以及广博的知识无时无刻不在激励着我。使我沿着正确的道路不断前进。我在研究生生涯的每次成长和进步都离不开老师的帮助与支持，在此我对我的导师表示最崇高的敬意和最衷心的感谢。

同时，我要感谢电子工程系的各位老师，感谢你们对我理论知识的指导，不仅为我之后的研究和论文撰写打下理论基础，而且扩展了我的知识面，增长了见识。

感谢实验室小组的李文姬师兄、余泽峰师兄、林惠标师兄、赵雷师弟、谢红辉师弟、肖杨师弟以及实验室的所有成员，是你们把实验室凝聚成团结合作，共同进步的集体，我非常荣幸能够成为其中一员。感谢同年級的容毅标，邱超乐，陈列慧，李芳，谢光晨等同学，感谢你们在学习中给予我的帮助和生活中给予我的关心与照顾，使我度过了三年美好的时光。

特别感谢我的家人，世上最大的恩情莫过于父母的养育之恩。感谢他们给了我宝贵的生命、健康的体魄和执着的精神。感谢弟弟对我学业上的支持，同时也感谢亲爱的女朋友在我受到挫折和失败的时候不断支持鼓励我，陪伴我。正是你们的不断支持与鼓励，使我顺利完成学业，在此表示衷心的感谢。

最后，感谢评审我论文的各位老师以及出席本次论文答辩的各位教授和专家。

个人简历

左右祥，男，1990 年 11 月出生于河南省周口市，2007 年就读于江西理工大学电子科学与技术专业，2011 年 6 月获得工学学士学位；2013 年 9 月至今就读于汕头大学信号与信息处理专业，硕士研究生，研究方向为医学信号处理。

1. 2015 年发表论文：“基于 EnFCM 的海量图像聚类分割算法的并行研究”，《微型机与应用》，2015 年第 34 卷第 15 期。
2. 2015 年发表 IEEE CEC 国际会议论文：“Prediction of Acute Hypotensive Episodes Using Random Forest Based on Genetic Programming”.
3. 2016 年发表 IEEE WCICA 国际会议论文：“Design of A Diamond Adsorption Detection System Based on Machine Learning Techniques”
4. 2015 年参加并完成莫斯科 Skoltech 大学为期两个月的国际交流项目，国家：俄罗斯。
5. 2015 年参加并完成以色列理工学院为期一个月的 Summer School，国家：以色列。
6. 2015 年参加由 IEEE 主办的 CEC 国际会议， 并做相关报告， 国家： 日本。
7. 2015 年，荣获 2015 年度研究生国家奖学金。
8. 2015，汕头市残联学校义教一个月
9. 2013-2016 年，担任电子系班长， 并荣获优秀班干部荣誉称号。
10. 2014 年，获得 2014 年全国电子设计大赛三等奖。
11. 2014-2015，担任广东省数字信号与图像处理重点实验室助管。
12. 2013-2014，担任数字图像处理课程助教。
13. 2013 年，担任 2013 级研究生迎新晚会主持人。