大连理工大学本科毕业设计（论文）

随机化贪心特征选择方法的设计与应用

Design and Application of Randomized Greedy Feature Selection Method

学 院（系）：电子信息与电气工程学部

专　　业：计算机科学与技术

学 生 姓 名：陈曦

学　　号：201485049

指 导 教 师：孟军

评 阅 教 师：林晓慧

完 成 日 期：2018.06.10

大连理工大学

Dalian University of Technology

摘　要

由于基因基因表达数据具有“高维度”、“小样本”、“高冗余”的特点，为了解决这类样本的分类的问题，传统的解决方案是首先进行特征选择，其次训练集成分类器模型,其中对于集成分类器的训练往往都需要有一个集成剪枝的步骤。然而，无论是特征选择还是集成剪枝，通常使用贪心算法。众所周知，贪心算法的缺点是，搜索空间相对于整个特征空间过于狭小，很多特征都没有被考虑到，其次是非常容易陷入局部最优得不到全局最优解。

本文提出了一种随机化贪心算法，首先将特征空间分成若干等份，每一次随机化贪心特征选择的第一步都在对应的等分特征空间随机选取一个特征，之后面对整个特征空间执行传统的贪心算法，如此进行若干次，最后对所得若干个特征集合进行整合。

这样一来就一定程度上地解决了贪心搜索的局限性，而且时间复杂度仅仅是传统算法的常数倍不变，避免了枚举搜索爆炸的缺点，是一个良好的在枚举搜索算法与贪心算法之间的折衷。并且进过实验数据表明，本文所提出的基于基因表达数据的随机化贪心特征选择方法的性能相比一些传统的算法有明显的提升。

关键词：随机化；特征选择；集成剪枝；基因表达数据；贪心算法

Design and Application of Randomized Greedy Feature Selection Method

Abstract

As gene expression data has high dimension, small sample and high redundancy, to stress the classification problem based on this kind of data is a hard problem. There is an old method. Firstly, because there is redundancy features, choosing the best feature subset is needed. Secondly, we need a proper model to train a good classifier, for ensemble classifier can integrate weak classifiers, we choose the ensemble classifier model where a pruning algorithm can't be ignored. However, both Greedy Feature Selection algorithm and the pruning algorithm is based on greedy algorithm. As we all know, the greedy algorithm has two significant disadvantages, on the one hand, it's search space compared the whole feature set is too small which leads to the problem that many feature cannot be considered, on the other hand, it is often easy to fall into a local optimum although there is a global optimal solution.

Therefore, we promoted a randomized greedy feature select algorithm. Firstly, we split the feature into several feature subsets. Secondly, in each time of randomized greedy feature selection, we choose the first feature in the corresponding feature subset, then rest step of each randomized greedy feature selection is the same as the old greedy feature selection algorithm. After several times of randomized greedy feature selection, we can get a few feature subsets. Finally, we can use specific methods to combine the feature subsets we got in the last step.

Thus, to a certain extent, we can solve the disadvantages of traditional greedy feature selection algorithm. Also, the time complexity of the new method is just a constant time of the traditional method. What is more, the new randomized algorithm can solve the disadvantage of enumerating search explosions, it is really a good election. Via the Experimental data, it can be illustrated that the new randomized greedy feature selection algorithm based on gene expression data is much better than the traditional greedy feature selection method.

Key Words：Randomized；Feature Selection；Pruning Algorithm；Gene Expression Data；Greedy Algorithm

目　　录

1 绪论

1.1 研究背景及意义

随着计算机科学与技术以及生物技术的高速发展，生物学方面相关的知识的研究和发现已不再单单依靠生物学方面的观察和实验。综合了计算机科学与技术、生物学、数学等等多门学科的交叉性学科——生物信息学，正在快速地发展，并且反过来推动着生物学相关知识的探索与发现。由于各个领域技术的发展以及迅速革新，从生物信息学研究中获取的数据的规模与日俱增，对生物大数据的知识挖掘与分析是当前生物信息学领域的一个重点研究课题[1]。

大多数植物在生长的过程中经常会遭受到各式胁迫因素的影响，这些胁迫因素影响着植物的生长和发育，影响着粮食的产量，影响着生态环境的稳定性和多样性。植物胁迫因素可以大致归纳为两大类：一类是非生物胁迫因素，主要源自于全球气候的变化以及发展工业所造成的环境污染，例如干旱、重金属、盐胁迫因素等等；另一类是生物胁迫因素，主要是指病原菌、病毒的感染和寄生植物、食草类昆虫的侵略，例如植物茎锈病、蚜虫、致病疫霉等。目前，仍旧有大量的生物方面的学者正在致力于研究植物对胁迫响应的机制与原理[2-5]，这些研究对于作物育种、农业生产、环境保护和林业等多方面都具有非常重要的意义。研究数据表明，基因组的表达信息是分析植物胁迫响应机制与原理的关键[6]。因此，分析基因表达数据并寻找关键性的差异表达基因，即基因选择，或称基因提取，是这一类研究的重要步骤。

高通量测序技术和基因微阵列芯片等生物技术的高速发展，为研究者提供了海量的基因表达数据。这些技术从转录水平的角度来讲，可以同时展示出所测样本对应于基因组中的成千上万条基因的表达情况，为基于基因表达数据的差异性表达基因的分析提供了有利条件[6,9]。基因表达数据已经广泛地应用于各个领域，例如疾病诊断领域[10,11]、癌症和肿瘤分类领域[9,12-14]、植物胁迫响应分析[4-6]等领域。然而，技术的发展也为数据分析和处理工作带来了新的挑战。对于基因选择这个问题，基因表达数据具有“高维度”、“小样本”和“高冗余”的特点，“高维度”是指检测的基因种类繁多（通常万的数量级），“小样本”是指由于进行一次样本检测获取数据的成本较高而造成的数据样本含量极少（通常仅为十或百的数量级），“高冗余”是指检测基因中包含大量与样本差异处理无关的冗余基因。

特征越多，训练模型的时间越长，同时模型的推广能力也越差。因此由于基因表达数据“高冗余”的特点，如何基于基因表达数据，从众多的基因中选择出差异表达的特异基因，是基因表达数据分析的研究热点和难点。

为了提高对基因表达数据的处理能力，许多基于机器学习的特征选择方法被用于基因选择，并取得了一定成果[5,15,16]。但是由于基因表达数据“小样本”的特点，即使进行了特征选择，样本与特征之间的比例也无法满足训练出一个强分类器的要求，所以使用单分类器模型并不能的达到令人满意的效果。由于一个集成分类器是由很多基分类器构成的，它将这些基分类器的结果结合在一起，可以比单分类器模型有更好的效果。将功能互补的基分类器结合起来可以使集成分类器的分类能力比每一个基分类器的分类能力强。如果两个分类器产生不相关的误差，则认为它们是互补的。因此，当互补分类器集成到一个集成中时，正确的决策会被集成过程放大[17-20]。有大量的研究工作证明集合学习经常可以提高分类和泛化性能（例如 bagging[3]、boosting[7]和 stacking[8]）。所以对于基因表达数据的分类问题，采用多分类器模型，即集成学习模型，将会达到较好的分类和泛化能力。

虽然集成分类器的性能卓越，但集合学习有两个严重的缺点。一是通常需要组合大量的基类分类器来确保分类错误率收敛到它的渐近值。这导致了巨大的内存需求和相当大的计算成本。二是集成分类器中分类效果不好的基分类器以及相互冗余的基分类器也会对最终的效果产生不良的影响。正因为这两个原因，在本文的基于基因表达数据的分类器模型的的训练的问题中，在特征选择和基分类器训练结束之后，对于大量的基分类器的集成剪枝也是需要解决的一个重要的问题。

1.2 研究现状

1.2.1 特征选择方法研究现状

如果想从一个特征集合中选取一个对于数据分类最有益处的，最能够彰显两种不同类别数据的特点的特征子集，并且没有相关领域的知识作为基础的话，即对于数据属性背景等相关知识一无所知没有任何的可以作启发式信息的知识的话，那就只能够采取枚举的方法遍历所有的数据子集了，其中数据子集的数目与数据属性的数目指数相关，这是非常可怕的。假如初始的特征集合有 300 个特征，特征子集的数目就有 2^300 个，即上亿个，这是没有办法接受的，尤其是本文的针对于基因表达数据的问题，特征子集的数目有上万个，所以穷举选择特征子集的方法是万万行不通的。所以采用贪心算法来进行近似最优特征子集的选取。

在子集搜索的过程中，给定特征集合$\{a_1,a_2,.....,a_d\}$，初始化特征子集为空集，在每一轮的迭代中，对于所有的没有被选入过特征子集的特征进行考虑，考虑将该特征加入到当前特征子集里，并对根据加入前后特征子集对于数据集合的区分的能力进行打分，全取打分最高的那个特征，确定地将它加入到集合当中。

这样一个一个地增加当前评价最优的特征的策略成为“向前搜索”(SFS)。类似地，如果从包含所有的特征的集合开始，一个一个地减少当前评价最差的特征，直至达到某一个终止的条件，这样逐渐减少特征的策略称为“向后搜索”(SBS)。如果将向前搜索策略和向后搜索策略结合起来，每一轮次，从待选集合中删除评价最差的特征并且向当前集合加入评价最好的特征，这样的搜索策略叫作 ”双向搜索策略”(BDS)。

常见的评价函数有相关性(Correlation)、距离(Distance Metrics)、信息增益(Information Gain)、一致性(Consistency)、分类器错误率(Classifier error rate)等等。

常见的特征选择方法可以划分为两种：过滤式、包裹式。过滤式方法是先进性特征初选，利用选择好的特征子集处理数据，再进行分类器的训练，这就是说特征子集的选择和分类器的训练是分开的步骤。例如 Relief 是一种著名的特征选择方法[Kira and Rendall,1992]；包裹式方法和过滤式方法恰恰相反，包裹式方法以最终所训练的分类器的性能作为评价指标，来对于特征进选择。换而言之，包裹式方法就是为分类器选择最有利于其性能，为其“量身制作”的特征子集。如 LVS(Las Vwgas Wapper)[Liu and Setiono,1996]是一个典型的包裹式特征选择方法。两类方法均有各自的优势和局限性，过滤式法操作简单，但是过分突出了单个基因的性能，而忽略了基因之间的相互影响，因此选出的基因子集的整体性能不稳定；包裹式法能够考虑选出的基因子集的整体性能，但是获得的这种较优的性能对分类器较敏感，同时这类方法的时间代价通常较高。

对于基于向前搜索的贪心特征选择方法 ZHAO 和 WU 等人提出过将加权本地模块化（WLMGS）作为评价指标来基于基因表达数据对于是有癌症判断预测[27]，这种评价指标是首先利用基因表达数据的各个属性的值构建加权网络模块，在这个网络中，控制同一种疾病的基因的距离近，控制不同类别的基因的距离远。然后计算加权本地模块化（WLMGS）的值，这个评价指标就是就算如果将某一基因加入当前基因子集后，新的子集对于癌症的鉴别能力增强程度的一个指标，这个越大，那么这个基因子集鉴别癌症的能力就更加强烈。相比于一些经典的评价指标，比如 t 检验家族中的 Z-score，贝叶斯评价家族中的贝叶斯 t 检验指标等，这个指标考虑到了当前基因子集内部的基因的相关性，避免了冗余。然而，这仅仅是对于评价指标的改变，并不能够改变先前搜索贪心算法的缺点。

1.2.2 集成剪枝技术研究现状

集成学习(ensemble learning)通过构建并结合多个分类模型(通常被叫做基分类器)来完成分类任务，有时也被称为多分类器系统(multi-classifier system)、基于委员会学习(committee-based learning)等。集成学习通过将多个分类模型结合，常常获得比单一分类模型更加优越的泛化性能（例如 bagging[3]、boosting[7]和 stacking[8]）。这些基分类器可以是同种算法组成的分类器，也可以是由不同算法组成的分类器。

然而，尽管它的性能卓越，但集合算法有一个严重的缺点，即通常需要组合大量的基类分类器来确保分类错误收敛到它的渐近值。这导致了巨大的内存需求和相当大的计算成本。因此集成剪枝技术成为近年以来的研究热点。

在多种集合修剪算法中，贪心集成剪枝技术（GEP）也称定向山爬山集成修剪技术（DHCEP），已经达到了良好的分类效果，因此引起了研究人员的关注[20-25]。 这种算法决定了特定的基分类器的选择和它们基于贪心选择策略被整合到集成分类器的顺序。他们从一个空的初始集合（或全集）开始，通过用一个模型迭代地扩展（或删除）初始集合来探索不同子集的空间。贪婪的选择是通过基于预测性能或者所替代子集多样性的评价指标来实施的。

对于集成剪枝算法，Partalas 等人提出了一种通过定向爬山（DHCEP）进行集合剪枝的不确定加权准确度指标（UWA），这种方法考虑了当前子集的决策的不确定性[12]。他们通过丰富的仿真实验验证了他们提出的算法的有效性[12]；Lazarevic 和 Obradovi[26]提出了一种首先利用 k-means 聚类方法对于基分类进行聚类，再在每一个类别中删除冗余分类器的方法来进行集成剪枝。进而达到删除冗余基分类器，完成集成剪枝的目的，其中，利用 k-means 聚类方法聚类的类数的取值是一个关键。然而，无论基于 UWA 的 DHCEP 算法还是利用 k-means 聚类方法删除冗余基分类器的集成剪枝算法，虽然具有较好的性能和较高的效率，但它仍然属于一种贪心算法。众所周知，贪心算法总是会选择当前最优的情况。 它得出局最优的选择，希望这种选择可以得出全局最优的解决方案。 尽管对于许多问题贪心算法产生最佳解决方案，但它并不总是实现这一目标。DHCEP 算法通常会产生集成剪枝问题的次优解，因为它相对于整个解的搜索空间仅仅考虑到一个较小的子区域。

1.3 本文的研究内容及结构

针对于本文所提出的基于基因表达数据的分类的问题，上文已经提到，因为基因表达数据的“高维度”、“小样本”、“高冗余”的特点，所以需要进行特征选择和集成分类器的训练和剪枝两个步骤。在这两个步骤中可以发现，贪心特征选择和集成分类器

的剪枝都是基于贪心算法的，都是先初始化空集，再向前搜索的，都是具有贪心算法的缺点。对于集成学习这个过程，基分类器是集成系统的一个成员，可以换一个角思考，将它看作为集成分类器的一个属性。那么特征选择那一个步骤和集成剪枝那一个步骤，就可以看作为贪心算法在不同的领域的应用了，并且这两个步骤唯一不同的地方就是中间每一次迭代时的评价指标。

既然本文的针对于基因表达数据的分类问题的两个主要的步骤是基于贪心算法的，都有贪算法的缺点，搜索范围相对于整个搜索空间相对狭小，可能获得局部最优解而非全局最优解。那么本文探讨的内容就是如何改进贪心算法的框架与结构，使得算法的特征搜索空间范围得到提高，尽量避免搜索陷入局部最优。

为了解决贪心算法的搜索空间相对于整个搜索范围过于狭小的问题，人们常常通过引入随机性来扩大搜索范围，避免搜索过早陷入局部最优。例如，随机产生序列选择算法(RGSS, Random Generation plus Sequential Selection) 通过随机产生一个特征子集，然后在该子集上执行 SFS 与 SBS 算法来扩大搜索范围，模拟退火算法(SA, Simulated Annealing) 以一定的概率来接受一个比当前解要差的解，因此有可能会跳出这个局部的最优解，达到全局的最优解，遗传算法(GA, Genetic Algorithms) 借鉴生物进化论，遗传算法将要解决的问题模拟成一个生物进化的过程，通过复制、交叉、突变等操作产生下一代的解，并逐步淘汰掉适应度函数值低的解，增加适应度函数值高的解。这样进化 N 代后就很有可能会进化出适应度函数值很高的个体，以及等等。

上述随机化心算法的启发，针对于本文的基于基因表达数据的分类的问题的特征选择和集成分类器的训练的两个步骤，本文设计了一种新的随机化贪心特征选择方法。在特征工程那一个步骤，首先，在特征选择的那一个步骤，先随机选择一个特征，作为初始的集合，再进行先前搜索，直至特征子集选取结束，如此重复进行 K 次，根据这 K 个特征子集训练基分类器。其次，在集成训练和剪枝那一个步骤，先随机选择一个基分类器，作为初始的集合，再进行先前搜索，直至基分类器集合取结束，如此重复进行 L 次，根据 L 个已经选取好的基分类器集合进行投票，最终投票数目超过某一个阈值的分类器将被选进最终的集成分类器。

提出本文这种针对于基因表达数据的分类的随机化贪心特征选择方法的动机有以下几点。首先，通过在经典的 SFS 算法中引入随机性，算法相对较窄的搜索空间被适当地扩展，虽然没有扩展很多。其次，本文提出的随机化贪心选择算法的计算复杂度只是经典 SFS 算法的常数倍，但是仍然比枚举算法快得多。因此，简言之，经典 SFS 算法的随机化让集成剪枝技术在有效性和效率性之间有了一个适当的折衷。本文提出的随机

化贪心选择算法通常具有比传统 SFS 有更好的性能，并且它的时间复杂度仍旧和后者相当。

此外，本文提出的算法的另一个重要优点是选择到最终集成分类器的基分类器由投票决定，这是一个非常符合我们直觉的特征。集成分类器的每个基分类器都可以被视为决策支持系统中的一名专家。这是非常合理的，决策支持系统中的所有专家都是根据某些特定标准的结果然后投票来选择的，这就证明了这些专家们的决策能力。

最终,本文提出的算法很自然地继承了随机算法通常具有的两个优势[22]。首先，在大多数情况下，其运行时间或空间要求比那些性能良好的确定性集成贪心算法要小得多。其次，这个算法非常易于理解和实施。

本文的组织结构如下：

第一章绪论介绍了本文课题的研究背景和意义综述了特征选择和集成剪枝技术研究现状和存在的问题。

第二章介绍了传统的贪心特征选择与集成剪枝技术的基本概念（如产生过程、评价函数、停止准则、验证过程等）与实现细则。

第三章详细阐述了本文的基于基因表达数据的分类问题的随机化贪心特征选择法的设计与应用。

第四章报告并且讨论了针对本文提出的方法的实验，验证方法，验证过程等等，并且讨论了一些参数的选择和设定。

第五章总结了这篇文章。

2 基于贪心策略的特征选择方法和集成剪枝算法

2.1 基于贪心策略的特征选择方法

首先介绍香农熵以及信息增益两个概念，然后介绍传统的基于贪心算法的特征选择方法。

2.1.1 香农熵以及信息增益

香农熵，简称为熵(Entropy)是评价数据集合杂乱程度的一个指标。数据集合的杂乱程度越大，熵的值就越大。对于给定数据集合 D，假定数据集合 D 中有$|y|$种样本，数据集合 D 熵的定义为：

$$Ent(\mathrm{D}) = -\sum_{i=1}^{|y|} p_k \log_2 p_k \tag{2.1}$$

其中，p_k (k=1,2… …,$|y|$)表示第 k 类样本在数据 D 中所占的比例。

信息增益(Information Gain)是衡量以某一个属性的某个指标划分数据集合 D 后，数据集合杂乱程度的变化，也就是划分前后香农熵值的变化。为了便于展示，假定数据集合 D 的属性是离散的。假定属性 A 划分数据集合 D，数据集合 D 被划分为了$\{\mathrm{D}_1, \mathrm{D}_2, ..., \mathrm{D}_v\}$这 v 个数据子集（在每一个数据子集 D_i (i=1,2……,v)中，属性 A 的取值都是相同的），此时信息增益定义为:

$$Gain(\mathrm{A}) = \mathrm{Ent}(\mathrm{D}) - \sum_{v=1}^{V} \frac{|\mathrm{D}_v|}{|\mathrm{D}|} \mathrm{Ent}(\mathrm{D}_v) \tag{2.2}$$

其中，$|D|$表示数据集合 D 的大小，$|D_v|$表示被属性 A 划分后的数据子集 D_i (i=1,2……,v)的大小。

2.1.2 基于贪心算法的特征选择方法

对于本文的针对基因表达数据的分类的问题，由于样本的种类是一个离散的变量，所以选用信息增益作为基于贪心策略的特征选择方法的评价指标。

对于拥有 n 个属性$\{a_1, a_2, ..., a_n\}$的数据集合 D：

D=$\{(x_1, y_1), (x_2, y_2), ..., (x_m, y_m)\}$

首先，将属性集合 A=$\{a_1, a_2, ..., a_n\}$中的每一个属性看作为有个候选子集，假定以每一个属性来划分数据集合 D，将信息增益(Information Gain)作为评价指标对各个属性进行评价，假设$\{a_2\}$最优，将$\{a_2\}$作为初始的特征子集；然后，对于那些没有被选入的

候选属性$\{a_1,a_3,.....,a_n\}$，在上一轮的基础上，尝试将其中的一个a_k加入上一轮的集合中，这样就有了特征子集 $\{a_2,a_1\},\{a_2,a_3\},.....,\{a_2,a_n\}$，用新加入的属性$a_k$再一次的划分上一轮划分过数据集合，计算新的信息增益，选出最优的那一个属性，假设a_4最优，则这一轮所选出的特征子集即为$\{a_2,a_4\}$；……假定第 k 轮数据集合的熵为 0 或者集合里的特征数目达到某一个阈值，那么将这个 k 特征集合作为特征选择的结果。

其流程图如图 2.1 所示。

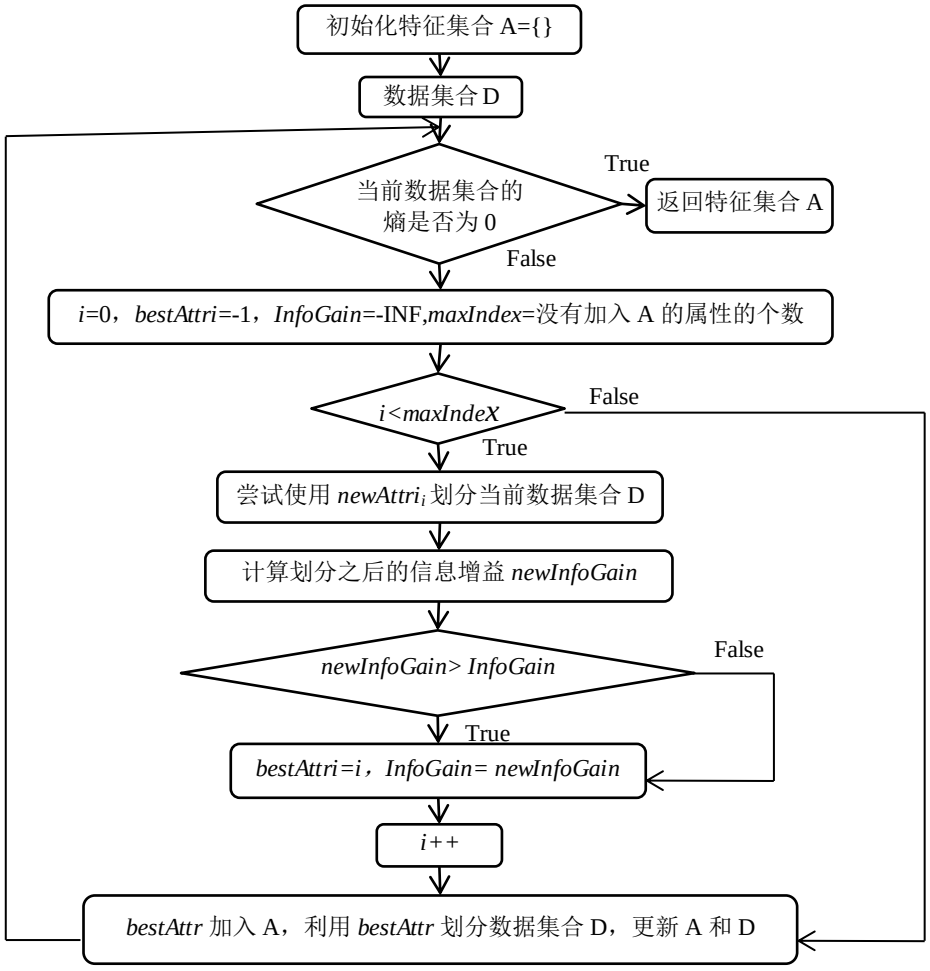

图 2.1 基于先前搜索的贪心特征选择方法

2.2 基于贪心策略的集成剪枝算法

在 2.2 这一节中，本文将介绍传统的基于贪心策略的集成剪枝算法，即定向爬山算法。至于定向爬山算法的评价指标，本文将选用较为 Partalas 等人提出的不确定加权准

确度指标(uncertainty weighted accuracy，UWA)[18]，这种评价指标考虑到了当前子集的决策的不确定性，并且他们通过丰富的仿真实验验证了他们所提出的这个指标的有效性。

Partalas等人提出了一种通过定向爬山（DHCEP）进行集合剪枝的不，该标准考虑到当前集成分类器S的不确定性，并对这种不确定性进行了精确的度量。并且他们通过丰富的仿真实验验证了他们提出的算法的有效性[18]。

因此这一节将有两部分内容，一是对于不确定加权准确度指标（UWA）以及相关概念的描述，二是对于传统的基于贪心策略的集成剪枝算法定向爬山算法的描述。

2.2.1 确定加权准确度指标（UWA）以及相关概念

首先集成剪枝这一个步骤是针对于由 L 个不同的基分类器组成的初始状态 H={ h_l ,l=1,2,……,L}开始的。对于初始状态集合 H 的集成剪枝过程利用了由 N 个样本构成的剪枝集合 $\mathrm{Pr}=\{(x_i,y_i),i=1,2,,...,N\}$ 进行的，其中其中 x_i 代表样本的特征向量，y_i 代表相应目标变量的值。

对于任意实例（x_i，y_i）$\in \mathrm{P_r}$，分别采用四种事件来描述基分类基分类器 h 与当前已经选好的剪枝集合 S 在（x_i，y_i）上的 4 种不同的分类结果[17]，即

$$e_{00}(h,S,x_i,y_i):h(x_i)\neq y_i \mathrm{and} S(x_i)\neq y_i \tag{2.3}$$

$$e_{01}(h,S,x_i,y_i):h(x_i)\neq y_i \mathrm{and} S(x_i)= y_i \tag{2.4}$$

$$e_{10}(h,S,x_i,y_i):h(x_i)= y_i \mathrm{and} S(x_i)\neq y_i \tag{2.5}$$

$$e_{11}(h,S,x_i,y_i):h(x_i)= y_i \mathrm{and} S(x_i)= y_i \tag{2.6}$$

其中 $e_{00}(h,S,x_i,y_i)$ 表示 h 没能正确地分类了实例（x_i，y_i），并且当前已经选好的剪枝集合 S 也没有能够正确分类实例（x_i，y_i）。

$e_{01}(h,S,x_i,y_i)$ 表示 h 没能正确地分类了实例（x_i，y_i），而当前已经选好的剪枝集合 S 可以正确分类实例（x_i，y_i）。

$e_{10}(h,S,x_i,y_i)$ 表示 h 可以正确地分类了实例（x_i，y_i），而当前已经选好的剪枝集合 S 没有能够正确分类实例（x_i，y_i）。

$e_{11}(h,S,x_i,y_i)$ 表示 h 可以正确地分类了实例（x_i，y_i），并且当前已经选好的剪枝集合 S 也正确分类实例（x_i，y_i）。

基分类器 h 对于当前已经选好的剪枝集合 S 的的不确定加权准确率指标 UWA 可以定义为：

$$UWA(h,S)=\sum_{i=1}^{M}(I\left(e_{10}\left(h,S,x_i,y_i\right)\right)NT_{\mathrm{i}}-I\left(e_{01}\left(h,S,x_i,y_i\right)\right)NF_i \\ +I\left(e_{11}\left(h,S,x_i,y_i\right)\right)NF_i-I\left(e_{00}\left(h,S,x_i,y_i\right)\right)NT_{\mathrm{i}}) \tag{2.7}$$

其中 I(true)=1, I(false)=0。对于任意一个样本（x_i，y_i）$\in \mathrm{P_r}$，NT_i 表示对于集合 S 中分类正确的基分类器所占的比例，对应地，$\mathrm{NF_i}$ 表示对于（x_i，y_i）集合 S 中分类正确的基分类器所占的比例，可以知道 $NT_i+NF_i=1$。在 $e_{10}\left(h,S,x_i,y_i\right)$ 和 $e_{00}\left(h,S,x_i,y_i\right)$ 中，NT_i 比 NF_i 的值小；然而在 $e_{11}\left(h,S,x_i,y_i\right)$ 和 $e_{01}\left(h,S,x_i,y_i\right)$ 中，NT_i 比 NF_i 的值大。UWA 表达式就是在计算，在已知当前集合 S 的情况下，h 对于 S 的重要程度，越重要，UWA 值越大。

在上述定义中，UWA 值与 $e_{10}\left(h,S,x_i,y_i\right)$ 以及 $e_{11}\left(h,S,x_i,y_i\right)$ 成正比，与 $e_{01}\left(h,S,x_i,y_i\right)$ 和 $e_{00}\left(h,S,x_i,y_i\right)$ 成反比。更细致一点讨论：对于（x_i，y_i）$\in \mathrm{P_r}$，当 $\mathrm{I}(e_{00}\left(h,S,x_i,y_i\right))=1$ 时，h 和 S 都对（x_i，y_i）分类错误，所以在 S 中加入 h 是使得 S 性能对于未知样本正确分类的能力更差的，所以 UWA 的值减去 NT_i；当 $\mathrm{I}(e_{01}\left(h,S,x_i,y_i\right))=1$ 时，S 对于（x_i，y_i）分类正确，h 对于（x_i，y_i）分类错误，若果 h 加入 S 会降低 S 的性能所以 UWA 的值减去 NF_i；当 $\mathrm{I}(e_{10}\left(h,S,x_i,y_i\right))=1$ 时，S 对于（x_i，y_i）分类错误，h 对于（x_i，y_i）分类正确，如果将 h 加入 S，那么加入 h 后的 S 很有可能对（x_i，y_i）的份分类结果就正确了，所以 UWA 的值加上 NT_i；当 $\mathrm{I}(e_{11}\left(h,S,x_i,y_i\right))=1$ 时，h 和 S 都对（x_i，y_i）分类正确，计入 h 后 S 的性能将更优，所以 UWA 的值加上 NF_i。这就是 UWA 值衡量新的基分类器 h 对于当前集合 S 的重要程度的原理。

2.2.2 传统的基于贪心策略的集成剪枝算法

集成剪枝算法是基于向前搜索的贪心算法的。对于拥有 L 个不同的基分类器的集合 H={h_l,l=1,2,……,L}，首先将最终的及成分类器集合初始化为空集 S。然后对于每一个没有被选中过的基分类器 h_l，尝试将它加入 S，计算这个基分类器相对于当前集合 S 的重要程度，即不确定性加权评价指标 UWA，对于 UWA 值最大的那个基分类器，将它真正的加入 S。一直到集成分类器集合 S 可以将所有的数据样本正确分类，或者 S 的大小到达特定阈值。

其流程图如图 2.2 所示。

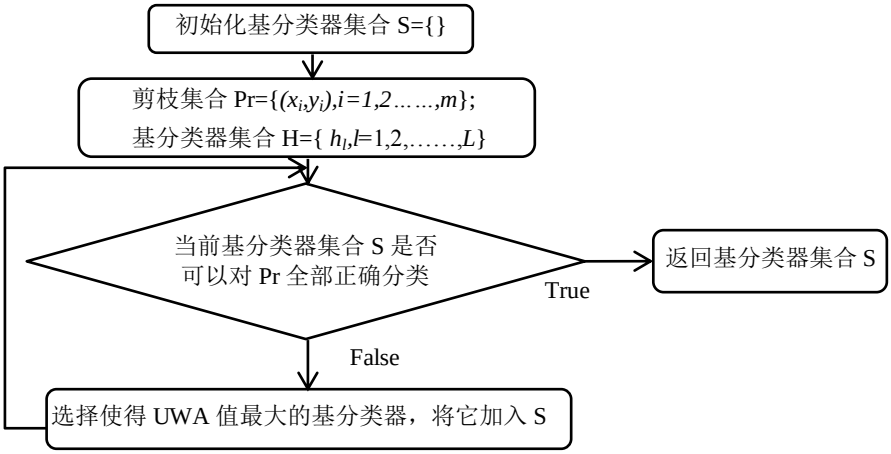

图 2.2　传统的基于贪心策略的集成剪枝算法

3 随机化贪心特征选择法的设计与应用

在第 2 章中已经阐述了传统的基因表达数据的分类问题的最主要的两个步骤，特征选择与集成剪枝那连个步骤，这两个步骤也是本文着重需要改进的两个步骤。在本章中将整体详尽地描述针对于基因表达数据的分类问题如何进行模型的训练，并且介绍基于基因表达数据的分类问题的随机化贪心特征选择法的设计与应用。

本章的展开方式如下。第一节，利用威尔克逊秩和检测(Wilcoson Rank Sum Test)原理，目的是将上万维的数据降低至几百到一千维。第二节，计算每一个属性的 gini 系数，目的是将数据离散，化，为后续的基于信息增益的贪心特征选择做准备。第三节，改进的随机化贪心特恒选择算法，目的是选出几百个数据子集，为后续的基分类器的训练做准备。第四节，基分类器的训练，主要讲的是利用上一步的特征子集，处理数据，并且训练基分类器。第五节，随机化贪心基分类器选择与投票，目的是利用随机化贪心基分类器选择与投票策略进行集成剪枝，选出最终的集成分类器。

3.1 威尔克逊秩和检测

虽然贪心算法采取了一定的策略以有可能陷入到局部最优为代价解决了特征选择组合爆炸的情况。但是对于本文的基于基因表达数据的分类问题来讲，首先，样本的基因表达常常是上万维的，由于贪心特征选择算法的每一次迭代都需要对于没有被选入的特征进行评估计算，这样计算下来所需要的时间复杂度也是极其大的。对 0 个特征子集选下来，所需要的时间就为 225,000,000* f()，以最常见的配置为于本文所针对的的拥有上万个属性的基因表达数据(大多情况下数据的维度为两万到三万)来讲，假设数据的维度是 2.5 万，每一次选择 30 个特征，总共选择 300 个特征子集，假设使用一个属性对于数据集合进行划分并且计算信息增益的时间为 f()，那么使用 Core i5 处理器,4GB 内存，64 位操作系统而言，使用 python 语言编写代码，进行这样一次 300 个特征子集的选择需要八九个小时。其次由于基因表达数据大多具有小样本的特点，在后续的过程中需要十折交叉验证，这样做所需的时间太长，是无法忍受的。

针对于特征初选，有很多种方法，其中最常见的一种就是特征排序法，即为每一个特征与样本类别的相关性打一个分数，根据分数对所有的特征进行排序，最后选择排序靠前的基因，至此初选过程结束。

但是，这种操作有一个显著的缺陷。这种操作只考虑到了与样本类别相关性强的单个特征，忽略了某些较弱的特征，当这些较弱特征出现在某一个特征子集里面，作为一个整体出现，时可以体现出很强的样本分类的能力。初始选择和样本类别相关性大的特

征进行降维和尽可能考虑更多的特征以防止上述缺陷的产生，这是一个 NP 难的问题，选择有一个良好的折衷的点是一个关键。

言归正传，继续介绍如何进行特征排序，特征初选。常见的方法有 T 统计检验、Relief 算法、χ^2 统计检验、Fisher 判别法、Wilcoxon 秩和检验和 Kruskal-Wallis 秩和检验等。T 统计检验、χ2 统计检验、Wilcoxon 秩和检验和 Kruskal-Wallis 秩和检验方法都属于假设检验统计方法，这些方法可以评估两类样本的差异。其中，T 统计检验和χ2 统计检验都需要样本满足正态分布（即高斯分布），而秩和检验属于非参数假设检验，对样本的分布没有要求。由于对于小样本的基因表达数据，并不了解他们满足什么分布，因此选用秩和检验方法来评估特征与样本类别的相关关系。又由于 Kruskal-Wallis 秩和检验方法适用于多分类问题，而本文的针对于基因表达数据的分类问题，如植物胁迫响应，这类问题普遍都是二分类的，所以本文采取威尔克逊秩和检测来评估特征与样本类别的相关关系。

3.1.1 秩与秩和的定义

对于基因表达数据来讲，可以将样本在某一个特征上的表达值，作为这个样本在这个特征上边的观测值。针对于本文的基因表达数据的问题，这样对于某一个特定的特征来讲，基因表达数据可以分为两类，即存在两组观测值，分别可以表示为：$X=\{x_i \mid i=1, 2, \ldots, n_1\}$和 $Y=\{y_j \mid j=1, 2, \ldots, n_0\}$，其中 X 中的样本均属于 $class_0$，Y 中的样本均来自于 $class_1$，令 $n= n_1 + n_0$。对这个样本按照特定特征的观察值进行升序排序，样本排在第几位，样本的秩就是几。但是有一种特殊的情况需要注意，有时，在样本中可能会有若干个样本在这个特定特征上的观测值是一致的。这是需要对样本的秩进行一些微小的调整。具体的调整方法为：对于 k 具有相同观测值的样本，假设他们原来的秩为$\{rank_1, rank_2, \ldots, rank_k\}$，对这些秩值取均值，再赋回去，即调整之后这个 k 个对应特征观测值相等的样本的秩都为$(rank_1+rank_2+ \ldots + rank_k)/k$。至此，对于秩值的计算讨论结束。

秩和的定义则较为简单，属于 $class_0$ 的样本集合 X 的秩和即为$\{x_i \mid i=1, 2, \ldots, n_1\}$在混合编秩过程中所获得的秩的和。同样，类似地，属于 $class_1$ 的样本集合 Y 的秩和即为$\{y_j \mid j=1, 2, \ldots, n_0\}$在混合编秩过程中所获得的秩的和。$W_0$ 表示 $class_0$ 的秩和，W_1 表示 $class_1$ 的秩和。

3.1.2 威尔克逊秩和检测方法

威尔克逊秩和检测方法是一种假设检验方法，给出两个假设检验。H0：X=Y，H1：X≠Y，H0 表示 X 和 Y 在某个特定特征上的分布是相同的，H1 表示 X 和 Y 在某个特定

特征上的分布是不相同的。当 n_1 和 n_0 的值都小于等于 12 时，可以通过计算秩和 W_1、W_0 与查威尔克逊秩和检测表的方法找出 P(X=Y)的概率，进而进行推断。但是，对于本文的针对于基因表达数据的分类问题上，基因表达数据虽然都是小样本的，但是样本的大小都是几十到几百的，并不能满足 n_1 和 n_0 的值都小于等于 12 的条件。对于这种情况威尔克逊表示，此时 W_0 服从均值为 μ_0 方差为 σ_0 的正态分布，即 W_0~Normal（μ_0，σ_0），并且 $\mu_0=n_0*(n_0+n_1+1)\div 2$；$W_1$ 服从均值为 μ_1 方差为 σ_1 的正态分布，即 W_1~Normal（μ_1，σ_1），并且 $\mu_1=n_1\times(n_0+n_1+1)\div 2$，$\sigma_0=\sigma_1=\sqrt{(n_1+n_0)\times(n_0+n_1+1)\div 12}$。更加准确地，$p(W_0\geq w_0)\approx p(Z\geq z)$。

其中：$z=\dfrac{w_0-\mu_0}{\sigma_0}$。

对于假设检验来说 $p=P\left(|Z|>|Z_g|\right)$ 的值越大，就有越大的可能拒绝 H0，接收 H1，即有更大的可能说明 X 和 Y 在某个特定特征上的分布是不相同的。在这里，Z_g 是对应基因 g 的 Z 统计量值。

因此基于基因表达数据，对于所有的所有的基因特征进行秩和检测，p 值越大的特征，在两个类别中的分布就越不相同，这个特征就越与样本的类别相关。

在本文的针对于基因表达数据的分类问题中，选取 p 值较大的 *numOfFeature* 个特征，作为基因初选的结果，以后的所有的操作都是针对于这 *numOfFeature* 个特征的。

3.2 数据离散化处理

在后面的随机化贪心特征选择的过程中，以信息增益作为某一个特征是否选入当前特征子集的的评价指标，但是信息增益的计算是针对于离散型的数据输入的，本文的针对于基因表达数据的分类问题的属性值都是连续的，所以需要对数据进行离散化处理。

离散化技术主要分为两大类—无监督数据离散化和有监督数据离散化。无监督离散化过程没有使用到样本的类别，而有监督离散化技术使用到了。常见的无监督离散化方法有等宽(equal-width)、等频(equal-frequency)技术，但是这样的等宽或者等频离散化方法有重大的缺点，例如如果间隔或者边界选取得不好的话，会造成实例分布不均匀以及多种类别的样本混合在一起的情况，这样会造成一些不良的后果。因此本文选用一种监督式数据离散化方法，使用 Gini 系数对数据进行离散化处理。

因此，这一小节的展开方式是先讲述 Gini 系数的定义，再讲寻找 Gini 系数的方法。

3.2.1 Gini 系数的定义

Gini 值(Gini Index)，更准确地说是 Gini 不纯度，是一种指标，它衡量了这样一个情况，如果从一个集合里随机选出一个样本并且根据样本子集中样本的分布情况标注，这个样本被标注错误的几率。直观地来讲，Gini 值反映了从数据集合 D 中随机抽取两个样本，其类别标记不一致的情况。Gini 不纯率的计算表达式如下：

$$Gini(p)=\sum_{k=1}^{K}p_k(1-p_k)=1-\sum_{k=1}^{K}p_k^{\,2} \tag{3.1}$$

其中，K表示样本类别的数目，p_k表示第k个类别的样本在整个数据集合中的概率，即统计频率。更加具体的，对于个给定的样本 D,假设有 K 个类别, 第 k 个类别的数量为 C_k，则样本 D 的基尼值表达式为：

$$Gini(\mathrm{D})=1-\sum_{k=1}^{K}(\frac{|\mathrm{C}_k|}{|\mathrm{D}|})^2 \tag{3.2}$$

其中 D 表示数据集，$|D|$表示数据集合的大小，C_k表示样本标签均为 $class_k$ 的数据子集，$|C_k|$表示数据子集C_k的大小。Gini 值越大，数据集合 D 的不纯度越高。

因此对于基因表达数据集合 D，某一个特征 a 的基尼指数定义为

$$Gini\_Index(\mathrm{D},a)=\sum_{v=1}^{V}\frac{|D_v|}{|D|}Gini(\mathrm{D}_v) \tag{3.3}$$

3.2.2 利用 Gini 寻找数据离散化对应的阈值

对于拥有 m 个样本的数据集合 D，在某一特征 a 下的取值有 m 个，将他们从小到大排列为$\{a_1,a_2,.....,a_m\}$。现在取相邻样本值的中位数，一共取得 m-1 个划分点，其中第 i 个划分点T_i的表达式为T_i=（$a_{i-1}+a_i$）/2。对于这 m-1 个划分点，分别利用该点对数据集合进行划分，计算以该点为划分点时所对应的 Gini 值。选取对应 Gini 值最小的那个划分点作为这个属性的离散化的阈值。

曾经有过质疑，对于数据离散化时，问什么是仅仅找寻一个值对于数据进行离散划分为两堆，为什么不多找寻几个划分点，将某属性下对应的数据划分为好几堆。在本文的针对于基于基因表达数据的分类问题上，本文的解释是，因为为了有较好的离散化的效果，选取的使用 Gini 系数进行离散化的方法是一种监督式离散化方法。在每一次的 Gini 值，也就是数据杂乱程度评价指标，的计算的过程中，所依赖的是数据的类别，由于本文所针对的问题的数据仅仅是二分类的，将数据多分成好几堆是毫无意义的。所以在数据离散化的过程中，仅仅选取一个阈值将数据离散化为两堆。

利用 Gini 寻找数据离散化对应的阈值的这一个步骤所对应的伪代码如下：

输入：数据集合 D=$\{(x_1,y_1),(x_2,y_2),.....,(x_m,y_m)\}$；

过程：函数 FindThreshold(D)

1:**while** 还有特诊没有计算 Gini 系数 **do**

2: i=1

3: **while** i<m **do**

4: T_i=（$a_{i-1}+a_i$）/2

5: 计算 $Gini(\mathrm{D})=1-\sum_{k=1}^{K}(\frac{|C_k|}{|D|})^2$

6: 找到对应 Gini 值最小的 T，将它作为该特征的阈值，为后边的数据化做准备

7: **return**

3.3 随机化贪心特征选择

在特征选择这一个步骤当中，曾经有过疑问，为什么不采用主成分分析法（PCA），而是采用特征选择。主成分分析法（PCA）是一种降维方法，它应用的场景是，样本所有的属性或者说特征都都与样本的类别相关，但是样本的维度太高，对于模型的训练不利。主成分分析法（PCA）的目的是，将高维度的样本映射到低维度当中（保持低维度空间的线性无关性，即低维度单位向量正交），并且使得数据信息的丢失最小；然而，在本文的基因表达数据集中，并不是所有的基因都与样本的类别相关，比如控制植物颜色的基因是与植物是否容易受到干旱影响毫不相干的，如果使用了降维方法主成分分析法（PCA），反而是将无关的属性学习到了，就会产生过拟合的现象。因此，本文采用的是特征选择方法，而非数据降维方法。

传统的贪心特征选择方法的两个缺点是，搜索范围相对于整个搜索空间相对狭小，可能获得局部最优解而非全局最优解另一个缺点是，每次仅仅考虑当前评价最优的那一个特征，忽略了如果此次选择不那么优秀的特征，最后总的一起体现的效果会更好的那一种情况。针对于缺点的两个方面以及后边要使用到需要多个基分类器的集成算法，便有了以下的解决方案。

首先第一步，并不是选择当前评价最优的那个特征，而是随机选取一个特征，随后的步骤和传统的贪心特征选择的步骤一致。

在这里可能会产生疑问，第一步随机选择一个特征，随后和传统的贪心特征选择一致，这样随机初始化的一步反倒是增加了算法的不确定性，何来增大搜索空间范围这一说呢。上述疑问确实是有道理的，为了解决这个问题，我们把随机初始化的贪心特征选择算法执行 *TimeOfRanFeaSelec* 次，并且每一次所进行随机化贪心特征选择的范围 U_i (*i*=1,2,……, *TimeOfRanFeaS*elec)都是不相同的，并且 U_i (*i*=1,2,……, *TimeOfRanFeaSelec*) 满足 $U_1 \cup U_2 \cup ..... \cup U_{TimeOfRanFeaSelec}$ 为全集并且 $U_1 \cap U_2 \cap ..... \cap U_{TimeOfRanFeaSelec} = \varnothing$。

这样一来，第一步 *TimeOfRanFeaSelec* 次随机初始化贪心特征选择所选择的特征就遍布了整个特征搜获空间，防止了某些特征一直得不到考虑的现象的产生，解决了搜索空间相对于整个特征搜索空间过于狭小的问题。其次随机初始化第一个特征这一步，就相当于是没有贪心地选择当前评价最优的那一个特征，这样的话就避免了贪心的缺点，就有可能会使得当前评价不是那么优秀但是总体评价很优秀地特征得到了选择。

此时，又有另一个疑问产生，对于这 *TimeOfRanFeaSelec* 次随机初始化贪心特征选择所选择出来的 *TimeOfRanFeaSelec* 个特征子集该如何处理，因为每一个特征子集仅仅它自己作为一个集合出现时才对于数据具有较好分类能力地子集，才具有意义。由于后续的过程中需要大量的基分类器的训练，特征选择所选择出来的 *TimeOfRanFeaSelec* 个特征子集，正好可以是每一个特征子集对于数据集合进行处理，然后利用处理好的数据集合训练基分类器。

以上便很充分地揭示了 *TimeOfRanFeaSelec* 次随机初始化贪心特征选择的原理与意义。下面详细地介绍随机化贪心特征选择的评价指标和算法。

对于评价指标，仍旧使用衡量数据程度变化的指标信息增益(Information Gain)。对于信息增益的具体介绍请参考本文的 2.1.1 小节。

随机初始化贪心特征选择的具体步骤如下：

首先，对于所有的特征进行编号，分成均匀大小的 *TimeOfRanFeaSelec* 序号区间段，使得每一个区间段所对应的特征集合 U_i (*i*=1,2,……, *TimeOfRanFeaSelec*) 满足 $U_1 \cup U_2 \cup ..... \cup U_{TimeOfRanFeaSelec}$ 为全集并且 $U_1 \cap U_2 \cap ..... \cap U_{TimeOfRanFeaSelec} = \varnothing$。然后独立地进行 *TimeOfRanFeaSelec* 次随机化贪心特征选择操作。在第 *i* (*i*=1,2,……, *TimeOfRanFeaSelec*)次随机化贪心特征选择中，第一次在之前划分好的特征子集 U_i 中随机选取第一个特征，之后和传统的贪心特征选择方法一样以真个特征集合为特征搜索空间，以信息增益为特征评价指标，迭代地一步一步的选择特征，直至以当前特征子集划分之后的数据集合的熵为 0 或者选取的特征的数目到达上限。该算法的伪代码如下：

输入： 训练集 D=$\{(x_1,y_1),(x_2,y_2),.....,(x_m,y_m)\}$;

经过秩和检测之后的属性集合 A=$\{a_1,a_2,.....,a_n\}$

进行随机初始化贪心特征选择的次数 *TimeOfRanFeaSelec*

过程： 函数 chooseAttributionSet(D,A, *TimeOfRanFeaSelec*)

1:将特征集合 A=$\{a_1,a_2,.....,a_n\}$ 划分为 *TimeOfRanFeaSelec* 份，其中每一份 U_i (*i*=1,2,……, *TimeOfRanFeaSelec*) 满足 $U_1 \cup U_2 \cup ..... \cup U_{TimeOfRanFeaSelec}$ 为全集并且 $U_1 \cap U_2 \cap ..... \cap U_{TimeOfRanFeaSelec} = \varnothing$

2:*i*=1

3:**while** *i*<=*TimeOfRanFeaSelec* **do**

4: 初始化 AttiSubSet={}

5: 从 U_i 里随机选出一个特征 $firAttr_i$ 加入 AttiSubSet

6: 以 $firAttr_i$ 划分数据集合 D，并且计算划分之后的熵

7: **while** $A \neq \{\} \& \mathrm{Ent} \neq 0$ **do**

8: **for** $newAttr_i$ in A **do**

9: 使用 $newAttr_i$ 划分当前数据集合 D，令每一个数据子集 D^k 对于 $newAttr_i$ 的取值都是一样的；

10: 计算新的数据集合 D^* 的熵，以及对应的信息增益 $Gain_i$;

11: **end for**

12: 选取使得 *Gain* 最大的那一个 *newAttr* 作为加入 AttiSubSet 的新的属性

13: 将使用 *newAttr* 划分后的数据集合作为新的数据集合，替 D

14: 将新的数据集合 D 的熵 *Ent* 作为新的 *Ent*

15: **end while**

16: 记录下此次随机初始化贪心特征选择的结果

17: **end while**

18: **return**

3.4 基分类器的训练

这一个步骤的目的是利用上一步得到的 *TimeOfRanFeaSelec* 训练 *numOfBaseClf* 个基分类器，其中 *TimeOfRanFeaSelec=numOfBaseClf*。

在集成算法中，所有的基分类器可以是由不同的算法构成的，也可以是由相同的算法构成的。常见的分类算法有 C4.5 决策树算法、BP 神经网络、KNN 等等。由于传统 SVM

模型是解决二分类问题的，可以解决非线性的分类问题，因此在本文的针对于基因表达数据的设计与分类的问题上，拟采用 SVM 算法训练若干个模型作为基分类器。有的人会质疑 SVM 算法的性能会不会太好，因而显现不出最后的集成算法的优点。本文给出的解释是这样的，针对于本文的基于基因表达数据的分类问题而言，由于成本问题，对于大多数的植物来说，可以的到的基因表达数据样本的数量仅仅有几十或者上百，即使使用当前性能比较好的，研究比较成熟的 SVM 算法来训练模型，最后测试得到所得的模型的 ACC 值也仅仅只有百分之五十流六十(以拥有 22810 个特征数据集合大小为 42 的 ArabidopsisNitrogen 样本来说，对它采取十折交叉验证训练 SVM 模型所得出的准确率仅仅为 0.4762，这个准确率甚至比不上随机猜测所得到的准确率，这样的模型也肯定是一个弱分类模型)，这样的模型的性能仅仅说是一个弱分类器的模型，远远达不到掩盖基分类器模型的性能的能力。

如果数据是线性可分的话，支持向量机的目的是找到一个找到一个最优的平面，是它可以正确地划分数据集合，并且能够最好泛化于其他的未知的数据样本，即找到一个平面使得它与离它最近的点的间隔最大；如果数据并不是线性可分的话，就需要找到一个合适核函数将数据映射到更高维度的空间，使得数据在更高的维度线性可分，从而继续地找到一个最优的超平面使得这个平面与离它最近的点的间隔最大。

训练支持向量机的模型有很多种，本文拟采用最小序列化(Sequential Minima Optimization)SMO 模型进行训练。本节将粗略地介绍 SMO 算法的原理。

首先定义间隔为数据集合中所有的点都分隔面最小间隔的 2 倍，其中到分隔面最近的点被称为支持向量(support vector)。如果将分隔面写成W^TX+b，则间隔的值就为$|W^TA+b|/||W||$。然后，本文来讨论分类器工作原理，对于未知样本，使用海维赛德阶跃函数(即单位阶跃函数)对于W^Tx+b进行处理，即$f(W^Tx+b)$，若W^Tx+b大于 0，$f(W^Tx+b)$的值为 1，，若W^Tx+b小于 0，$f(W^Tx+b)$的值为-1,。对于未知模型的训练，任务自然而然就是找出超平面W^TX+b,即寻找系数W和b使得间隔值$|W^TA+b|/||W||$最大。为此，必须找到最小间隔的数据点，即找到支持向量(support vector)。这样一来，就是寻找$\arg\max_{w,b}\{\min_n(f(W^Tx+b)*(W^Tx+b))\bullet\frac{1}{||W||}\}$，解决以上表达式，即对于乘积进行优化，是一个很难的问题，为此使得$f(W^Tx+b)*(W^Tx+b)$的值大于等于 1，此时仅需最小化||W||就可以找到$\arg\max_{w,b}\{\min_n(f(W^Tx+b)*(W^Tx+b))\bullet\frac{1}{||W||}\}$了，约束条件就是$f(W^Tx+b)*(W^Tx+b)>=1$。由于所有的约束都是基于数据集合中一个个的数据点的，超

平面可以写成基于数据点的形式(假设数据集合 D 的大小为 m，并且最终可以优化为 $\max_{a}[\sum_{i=1}^{m}\alpha=1-\frac{1}{2}\sum_{i,j=1}^{m}lable^{(i)}\bullet lable^{(j)}\bullet a_i\bullet a_j\left\langle x^{(i)},x^{(j)}\right\rangle]$，其中的约束条件为 C>= α >=0 和 $\sum_{i=1}^{m}lable^{(i)}\bullet a_i=0$。

SMO 是一种训练 SVM 模型的算法，它的含义是序列最小化，它的思想是分而治之，将大的复杂的优化问题分解成为小的多个小的易于求解的优化问题来一一求解，逐个攻破，并且这里满足分而治之之后的求解和整体求解的结果一致。SMO 算法的目的就是要求出所有的 α，然后根据 a 去求 W 和 b。

SMO 算法的主要思想是，由于 α 的数量很多，先要一起全部期初最优的 α 值是非常不容易的，因此在每一次循环之中找到两个 α 进行优化处理，其他的值看作为常数。其中这里的一对对 α 需要满足这一对 α 中的任意一个需要在间隔的边界之外并且没有进行过区间优化处。一旦找到一对 α 就对其中一个进行增大，对于另一个进行减小，使得满足 KTT 条件。直至所有的 α 都满足条件，最终求出 W 和 b，此时就找到间隔面了。

利用随机初始化贪心特征选择的那一个步骤所得出的 *TimeOfRanFeaSelec* 特征子集处理过初始数据集合得到 *TimeOfRanFeaSelec* 个数据集合，利用它们分别相互独立训练出 *numOfBaseClf* 个基分类器*(TimeOfRanFeaSelec=numOfBaseClf)*，这 *numOfBaseClf* 既可以作为下一步集成剪枝的输入。

3.5 基于随机化贪心特征选择与投票的集成剪枝算法

基于随机化贪心特征选择与投票的集成剪枝算法与随机初始化贪心特征选择算法是极为相似的，区别仅仅在于投票那一个步骤。

对于这个改进了的基于随机化贪心特征选择与投票的集成剪枝算法的每一步贪心迭代地评价指标仍旧选择可以衡量新的基分类器对于当前集合的重要性的不确定加权准确度指标（UWA），对于不确定加权准确度指标（UWA）的具体描述与解释可以参考本文的 2.2.1 那一小结。

基于随机化贪心特征选择与投票的集成剪枝算法的具体步骤如下：

首先对于所有的 *numOfBaseClf* 基分类器 $clfToBeChos_i$ (i=1,2,……, *numOfBaseClf*)进行编号，并且对它们进行均匀分堆操作，分成 *TimToChos* 堆，使得 $U_1\cup U_2\cup.....\cup U_{TimToChos}$ 为全集(即所有的待选择的基分类器)并且 $U_1\cap U_2\cap.....\cap U_{TimToChos}=\varnothing$。然后独立地进行 *TimToChos* 次随机化贪心特征选择操作。在第 i(i=1,2,……, *TimToChos*)次随机化贪心特征选择中，第一次在之前划分好的特征子

集 U_i 中随机选取第一个基分类器，之后和传统的贪心特征选择方法一样以整个基分类器集合为特征搜索空间，以 UWA 为特征评价指标，一步一步地，直至当前基分类器子集可以对剪枝集合中的所有数据进行分类或者分类的准确率达到收敛。

因为最后的集成分类器模型必须选出一个最终的分类器集合，所以本文打算采取模拟人的投票策略。按照搜有的被选择出来的 *TimToChos* 基分类器子集而言，按照编号对于所有的基分类进行投票，对于某一个基分类器而言，如果他出现在某一个特征子集一次，就为他投上一票，对于所有的基分类器的票数进行统计，所得票数大于某一阈值 *ThresholdOfBase* 的基分类器被选入最终的集成分类器里。其中对于投票阈值 *ThresholdOfBase* 的设定，如何使得它最优从而让最终的集成分类器的效果最好，目前对于这个值的设定没有给出太好的方法只能通过反复尝试的方法（在本文的针对于基因表达数据的分类的问题中，这个投票阈值 *ThresholdOfBase* 的取值一般是大于 10 的）。这个投票的一个步骤是非常符合人们行为的一个方法而且便于理解。

该算法的伪代码描述如下：

输入：剪枝集合 $\mathrm{Pr} = \{(x_i, y_i), i = 1, 2, , \ldots, N\}$；

基分类器集合 H={ h_l ,*l*=1,2,……,*L*}

过程：函数 chooseBaseClfSet (P_r ,H,*TimToChos*, *ThresholdOfBase*)

```
1:i=1
2:while i<=TimToChos do
3:        初始化基分类器集合 S={},并且在 U_i 随机选择一个基分类器加入 S
4:      while |S|<Threshold and S 不能对所有(x_i, y_i)∈P_r 正确分类 do
5:            for newH in H and newH 没有被选入 S do
6:            利用 UWA 评价 newH
7:          end for
8:            找出对应 UWA 值最大的 newH，将它加入 S 中
9:      end while
10:     i +=1
11:end while
12:for 每一个选择出来的基分类器集合 do
13:       for 该基分类器集合中的基分类器 do
14:                为它投上一票
15:       end for
16:end for
17:对于投票值大于 ThresholdOfBase 的基分类器将它选入最终的集成分类器
18:return
```

3.6 算法总结

至此，本文的基于基因表达数据的分类问题的随机化贪心特征选择算法已经介绍完毕。在本小节，本文将在此对算法进行回顾与总结，并且将讨论本文所提出的新的基于基因表达数据的分类问题的随机化贪心特征选择算法的时间复杂度进行讨论。

对于算法的时间复杂度来讲，假设传统的贪心特征选择算法的时间复杂度 O(f()),支持向量机模型算法的时间复杂度是 O(h()),传统的基于贪心算法的集成剪枝技术的时间复杂度是 O(g())，这样一来，传统的针对于基因表达数据的集分类问题的解决所需要的时间复杂度即为 O(f()+h()+g())。对于本文所提出的针对于基因表达数据的分类问题的随机化贪心选择算法的设计与应用问题而言，所需要的时间复杂度为 O(*TimeOfRanFeaSelec**f()+*numOfBaseClf**h()+*TimToChos**g())，其中(*TimeOfRanFeaSelec* = *numOfBaseClf*)，即时间复杂度为 O(max{ *TimeOfRanFeaSelec*, *TimToChos* }*max{f()，h()，g()})。可以看出改进后的算法的时间复杂度仅仅为改进之前的常数倍，就可以一定程度上解决传统的贪心算法搜索空间过于狭隘，容易陷入局部最优而非全局最优的缺点，这是在时间和算法的效率上的一个良好的折衷。

下面将给出整个针对于基因表达数据的分类算法的流程图如图 3.1 所示。

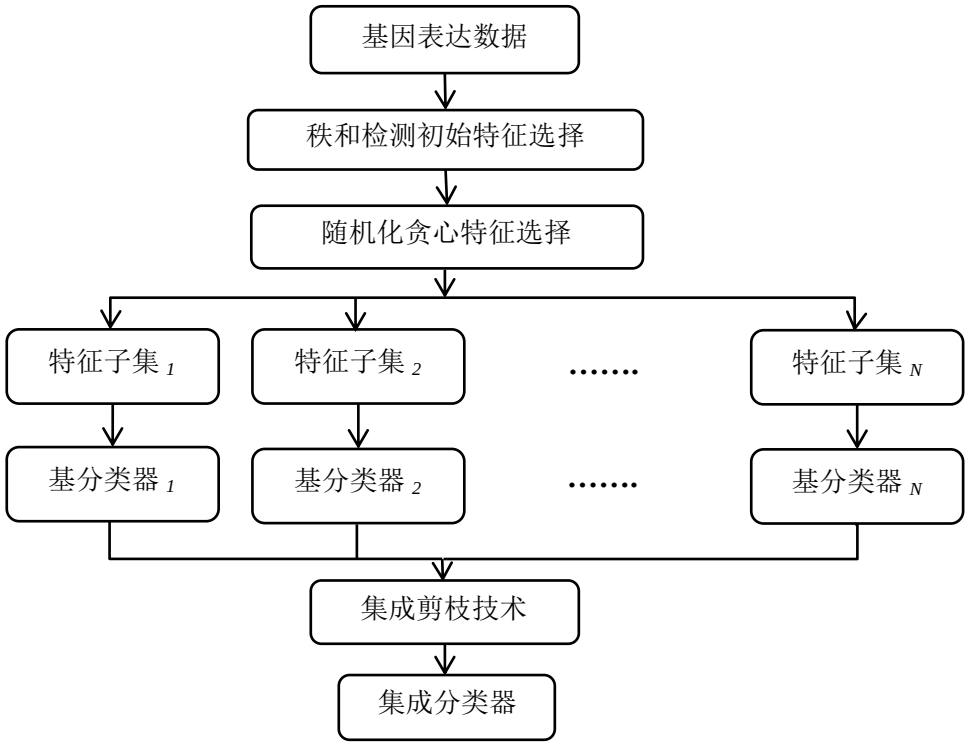

图 3.1　针对于基因表达数据的分类算法的流程图

4 实验结果与分析

4.1 实验数据

为了验证本文所提出的针对于基因表达数据的分类问题的随机化贪心特征选择算法的优越之处,本文拟采用来自于 TAIR(The Arabidopsis Information Resource, http://www.arabidopsis.org)的基因表达数据进行验证。在本文所选取的三个实验数据集合中正负样本的比例都是均匀的。

其中，ArabidopsisDrought 的数据集合是关于拟南芥水稻是否收到干旱影响的；ArabidopsisNitrogen的数据集合是关于拟南芥水稻是否收到氮钾影响的；ArabidopsisTEV的数据集合是关于拟南芥水稻是否收到 TEV 影响的。

表 4.1 列出了本文实验验证的数据集合的基本信息(数据集合类型名称、数据集合所含有的样本数目、数据集合属性的数目)。

表 4.1 实验数据集合的基本信息

数据集合类别	数据集合大小	样本特征数目
ArabidopsisDrought	59	20728
ArabidopsisNitrogen	42	22810
ArabidopsisTEV	53	32917

4.2 模型评估指标与方法

4.2.1 模型评估指标

对于评价指标，仅仅使用单一的评价指标对模型属性的度量不全面，本文采用查准率(precision)、查全率(recall)、*F1* 值、精度，下面将介绍各个指标的定义以及计算方法。为了介绍查准率,查全率以及 *F1* 值,先介绍真正例(true positive)、假正例(false positive)、真反例(true negative)、假反例(false negative)四种概念。真正例(*TP*)指的是样本本身为正例并且也被分类为正例；假正例(*FP*)指的是样本本身为反例，却被分类为正例的情况；真反例(*TN*)指的是样本本身为反例也被分类为反例的情况；假反例(*FN*)指的是样本本身为正例却被分类为反例的情况。本文中所利用到的评价指标的定义和计算如下：

查准率(precision)：分类器中所分类出的正例有多少是真正的正例，计算方法为 $P = \frac{TP}{TP + FP}$。

查全率(recall)：在所有的样本中，有多少的正例被准确地找到了，计算方法为 $R = \frac{TP}{TP + FN}$。

F1 值：*F1* 值是基于查准率和查全率的调和平均数，计算方法为 $F1 = \frac{2 * TP}{样本总数 + TP - 1}$。

精度(*ACC*):指的是样本中有多大比例的样本被正确地分类了，计算方法为 $ACC = \frac{TN + TP}{样本总数}$。

4.2.2 模型评估方法

对于模型评估方法，有留出法(hold out)、交叉验证法(cross validation)等。对于留出法而言，为了有充分的样本对于模型进行训练和测试，一般是使用 2/3 至 4/5 的样本进行测试，并且测试集的大小应该控制到 30 以上，对于本文所针对的基因表达数据而言，它具有“小样本”的特点，根本就无法满足留出法的要求，因此本文采用交叉验证(Cross Validation)的方法中常见的十折交叉验证与留二法相结合(leave two out)。

首先将数据集合 D 划分成为 10 个大小相等的互斥的数据子集，即 $D_1 \cup D_2 \cup ..... \cup D_{10} = D$ 且 $D_i \cap D_j = \varnothing (i \neq j)$。对于基于基因表达数据的随机化贪心算法进行十次，每一次选取六个数据子集作为训练集合，两个数据子集作为剪枝集合，两个数据子集作为测试集合。每一次都算出查准率、查全率、*F1* 值和精度。然后对于这十次得出的评估指标求平均值，即为最终的结果。

同时，为了体现出本文的针对于基因表达数据的随机化算法的优势，本文将选用 SVM 算法、Bagging 集成算法、Stacking 集成算法、Adaboost 集成算法和 RandomSubspace 集成算法与本文所提出的方法进行比较。使用 SVM 算法进行对比是为了体现出集成算法的优点；而 Bagging 集成算法、Stacking 集成算法、Adaboost 集成算法和 RandomSubspace 集成算法都是不同类别的集成算法的代表，与他们进行比较可以体现出本文的引入了随机化算法的优缺点。

4.3 实验结果

基于 ArabidopsisDrought、ArabidopsisNitrogen 和 ArabidopsisTEV 基因表达数据的实验结果如表 4.2、表 4.3、表 4.4 所示(在下表中，随机化算法就是本文的针对于基因表达数据的随机化贪心算法的缩写)。

表 4.2　基于 ArabidopsisDrought 的详细实验结果

classifier	*ACC*	*P*	*R*	*F1*
SVM	0.9661	0.967	0.967	0.967
Bagging	0.9152	0.903	0.933	0.918
Stacking	0.5084	0.508	1	0.674
Adaboost	0.983	0.968	1	0.984
RandomSubspace	0.9661	0.967	0.967	0.967
随机化算法	0.991	1	0.976	0.99

表 4.3　基于 ArabidopsisNitrogen 的详细实验结果

classifier	*ACC*	*P*	*R*	*F1*
SVM	0.4762	0.476	0.476	0.476
Bagging	0.619	0.593	0.762	0.667
Stacking	0.802	0.850	0.857	0.853
Adaboost	0.7143	0.737	0.667	0.7
RandomSubspace	0.5	0.5	0.381	0.432
随机化算法	0.825	0.825	1	0.8825

表 4.4　基于 ArabidopsisTEV 的详细实验结果

classifier	*ACC*	*P*	*R*	*F1*
SVM	0.6226	0.633	0.679	0.655
Bagging	0.7169	0.724	0.75	0.737
Stacking	0.5283	0.528	1	0.691
Adaboost	0.73	0.742	0.663	0.670
RandomSubspace	0.6415	0.696	0.571	0.627
随机化算法	0.71	0.756	0.683	0.676

从表 4.2 可以看出，基于 ArabidopsisDrought 数据集合的随机化贪心特征选择方法在所有评价指标上都比 SVM 算法、Bagging 集成算法、Stacking 集成算法和 RandomSubspace 集成算法好，相比于 AdaBoost 算法，除了在 *F1* 值上略逊一筹之外，本文所提出的算都是较优的。

从表 4.3 可以看出，基于 ArabidopsisNitrogen 数据集合的随机化贪心特征选择方法在所有评价指标上都比其他的作为对比的算法有优势

从表 4.4 可以看出，基于 ArabidopsisTEV 数据集合的随机化贪心特征选择方法除了召回率(recall)和 *F1* 上的值没有优于 Bagging 集成算法之外，在其他的评价指标上相比于其他的算法都是有优势的。

综上，本文所提出的基于基因表达数据的随机化贪心特征选择方法除了在个别的情况下都是优于传统的没有引入随机性的算法的。

结　论

由于本文的基于基因表达数据的分类问题，传统的做法是特征选择，在集成剪枝得到集成分类器地贪心特惠特征选择方法，尽量地考虑到了整个搜索空间，解决了搜索空间过于狭方法。在特选择和集成剪枝的那一个步骤都是使用到了贪心算法，因此，可以将集成剪枝算法中的每一个基分类器看成是最终的集成分类器的一个特征。众所周知，贪心算法的搜索空间过于狭并且容易陷入局部最优。因此本文设计了随机化贪心特征选择方法，通过若干次的第一步随机的并且容易陷入局部最优的问题。而且这样做的时间复杂度仅仅是传统方法的数倍，但是呢即使这样处理对于基因表达数据问题来讲，所需要的时间还是过长，这是因为基因的数目实在是太大多了，因此在贪心特征选择之间，设计加入了一个基因初选的步骤。

最终，本文基于 ArabidopsisDrought、ArabidopsisNitrogen 和 ArabidopsisTEV 基因表达数据采用十折交叉验证的方法将本文提出的方法与 SVM 算法、Bagging 集成算法、Stacking 集成算法、Adaboost 集成算法和 RandomSubspace 集成算法相对比，证明了本文提出的算法的优越性。

参 考 文 献

[1] 黄德双.基因表达谱数据挖掘方法研究[M].北京:科学出版社,2009.

[2] OSAKABE Y, OSAKABE K, SHINOZAKI K, et al. Response of Plants to Water Stress[J]. Frontiers in Plant Science, 2014, 5, Article 86:1-8.

[3] BREIMAN L. Bagging predictors Mach[J]. Learn, 1996, 24:123－140.

[4] SWINDELL W, HUEBNER M, WEBER A. Plastic and adaptive gene expression patterns associated with temperature stress in Arabidopsis thaliana [J]. Heredity, 2007, 99(2):143-150.

[5] LIANG Y C, ZHANG F, WANG J X. Prediction of drought-resistant genes in Arabidopsis thaliana using SVM-RFE [J]. PloS one, 2011, 6(7):e21750.

[6] 林海建,张志明,沈亚欧,等.基因芯片研究植物逆境基因表达新进展[J].遗传,2009,31(12):1192-1204.

[7] SCHAPIRE R. The boosting approach to machine learning: an overview[J]. MSRI Workshop on Nonlinear Estimation and Classification,2013,s 28－30 (3):135-142.

[8] WOLPERT D.H. Stacked generalization[J]. Neural Networks, 1992, 5: 241－259.

[9] 吴佳楠.基因表达数据分析方法及其应用研究[D].长春：吉林大学，2013.

[10] KHASHEI M, HAMADANI A Z, BIJARI M. A fuzzy intelligent approach to the classification problem in gene expression data analysis[J]. Knowledge-Based Systems, 2012, 27:465-474.

[11] FERNÁNDEZ-NAVARRO F, HERVÁS-MARTÍNEZ C, RUIZ R, et al. Evolutionary Generalized Radial Basis Function neural networks for improving prediction accuracy in gene classification using feature selection[J]. Applied Soft Computing, 2012, 12(6):1787-1800.

[12] PRODROMIDIS A L , STOLFO S J . Cost complexity-based pruning of ensemble classifiers[J]. Knowledge & Information Systems, 2001, 3:449－469.

[13] MARTINEZ-MUNOZ ,GSUAREZ A. Aggregation ordering in bagging[C].Interna- tional Conference on Artificial Intelligence and Applications,Acta,2004，258-263.

[14] WANG S L, LI X L, ZHANG S W, et al. Tumor classification by combining PNN classifier ensemble with neighborhood rough set based gene reduction [J]. Computers in Biology and Medicine, 2010, 40(2):179-189.

[15] 孙鑫.机器学习中特征选择问题研究[D].长春:吉林大学,2013.

[16] 张靖,胡学钢,李培培.基于迭代Lasso的肿瘤分类信息基因选择方法研究[J].模式识别与人工智能,2014,27(1):49-59.

[17] HANSEN L K, SALAMON P. Neural network ensembles[J]. IEEE Trans. Pattern Anal. Mach. Intell，1990, 12: 993－1001.

[18] KROGH A, Vedelsby J. Neural network ensembles, cross validation, and active learning[D].Cambridge: MIT, 1995.

[19] MART′INEZ-MU˜NOZ G, HERN′ANDEZ-LOBATO D.SU′AREZ A, An analysis of ensemble pruning techniques based on ordered aggregation[J]. IEEE Trans. Pattern Anal. Mach. Intell, 2009, 31: 245－259.

[20] DAI Q, A competitive ensemble pruning approach based on cross-validation technique[J]. Knowledge-Based Systems,2013, 37: 394－414.

[21] MARGINEANTU D, DIETTERICH T. Pruning adaptive boosting[C] .Proceedings of the 14th International Conference on Machine Learning，California, 1997, 9(4):324-325.

[22] PARTALAS I ,TSOUMAKAS G ,VLAHAVAS I. An ensemble uncertainty aware- measure for directed hill climbing ensemble pruning[J] Mach. Learn,2010, 81:257－282.

[23] BANfiELD R E , HALL L O ,BOWYER K W, et al.Ensemble diversity measures and their application to thinning[J]. Inf. Fusion,2005, 6: 49－62.

[24] CARUANA R, NICULESCU-MIZIL A, CREW G, et al. Ensemble selection from libraries of models[C]. Proceedings of the 21st International Conference on Machine Learning, Banff ,2004, 18.

[25] CHEN L, CHEN W Q, QIU C, WU Y, et al. LibD3C ensemble classifiers with a clustering and dynamic selection strategy[J]. Neurocomputing ,2014 ,123:424－435.

[26] YANG X B, SONG X N, CHEN Z H, et al. On multigranulation rough sets in incomplete information system [J]. International Journal of Machine Learning and Cybernetics, 2012, 3(3):223-232.

[27] ZHAO G D,WU Y. Feature Subset Selection for Cancer Classification Using Weight Local Modularity [J].Scientific Reports.2016, 10:100-118.

致　　谢

这次毕业设计，应用到了我的本科学到的基础知识，使得我对四年学到的东西有了一个统筹的认识。在秩和检测进行基因初选的阶段，我利用到了概率论里面的假设检验的知识；在这次毕业设计所使用到的基分类器——支持向量机中，我使用到了工科数学分析里的拉格朗日乘数法的知识与线性代数的知识；在对于贪心算法的优劣性的讨论中，我再一次的地回顾了算法的时间复杂度，空间复杂度，等等。

在这次毕业设计当中，我非常感谢我的导师孟军老师和张晶师姐，感谢她们每一周的耐心的指导。同时我也很每一周参加组会的同学，他们对于自己的项目的介绍也让我了解到了更多的有关于机器学习的知识，比如神经网络等。