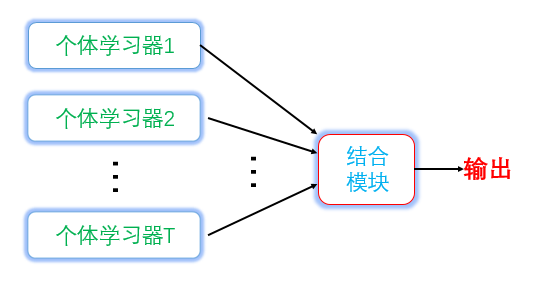
**基于WEKA的数据挖掘算法学习（五）**

# 集成学习的理论知识

## 1.1个体与集成

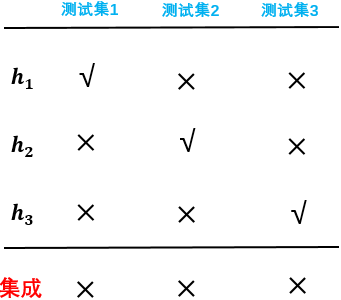
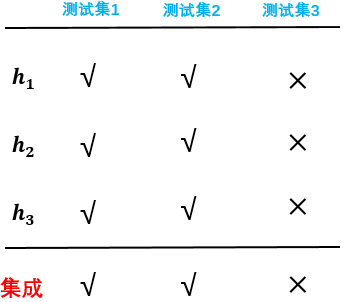
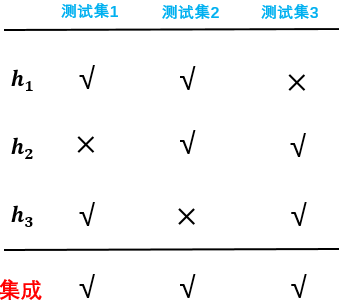
集成学习通过构建并结合多个学习器来完成学习任务，也成为多分类器系统。一般结构：产生一组“个体学习器”，再用策略将其结合起来。个体集成器通常由现有的学习算法从训练集产生，BP神经网络、C4.5等等。



**图1 集成学习示意图**

由集成学习示意图可以清楚展示集成学习之间的关系，将多个学习器结合，通常可获得比单一学习器显著优越的泛化性能。

要获得比单一学习器更好的性能，获得好的集成，应该满足个体学习器“好而不同”。



**图2 集成个体应好而不同示意图**

很明显，由图2知道，集成效果是第一幅图实例，即个体好而不同，集成效果更好。

目前两类集成学习方法：1、个体学习器存在强依赖关系，必须串行生成的序列化方法；2、个体学习器间不存在强依赖关系，可同时化生成的并行化方法。

## 1.2 Boosting

Boosting是一族可将弱学习器提升为强学习器的算法。工作机制：先从初始训练集训练出一个基学习器，由基学习器的表现对训练样本分布进行调整，然后根据调整后的样本分布来训练下一个基学习器；重复此过程，直到学习器数目达到指定的值T，将T个基学习器进行加权结合。

代表是AdaBoost算法,其中，f是真实函数。AdaBoost算法推导过程主要在于求三个东西：

错误率：

分类器权重更新：

样本分布更新公式：

最终获得输出：

## 1.3 Bagging

并行式集成学习方法，采用自助采样法经过m次随机采样操作，得到m个样本的采样集，初始训练集中约有63.2%的样本出现在样本集中。

采样出T个含m个训练样本的采样集，基于每个样本集训练出一个基学习器，再将基学习器结合。

Bagging算法过程：

对1...T个样本集，

输出：

此外，自助采样过程给Bagging带来一个优点：基学习器只用到训练集中63.2%的样本，剩下约36.8%的样本可用作验证集来对泛化性能进行“包外估计”。

包外样本其他作用：基学习器是决策树，可使用包外样本来辅助剪枝；当基学习器是神经网络时，可用包外样本来辅助早期停止以减小过拟合风险。

## 1.4 Random Forest

随机森林，简称：RF，是Bagging的扩展变体。RF在以决策树为基学习器的Bagging集成的基础上，进一步在决策树的训练过程中引入了随机属性选择。

在RF中，对基决策树的每个结点，从该结点的属性集合中随机选择一个包含k个属性的子集，再从子集中选择一个最优属性用于划分，推荐值。

随机森林中的基学习器多样性来自于样本扰动和属性扰动，加入属性扰动会使个体学习器的性能有所下降，随着学习器的数目增加，随机森林可收敛到更低的泛化误差。

## 1.5多样性

### 1.5.1误差-分歧

个体学习器泛化误差的加权均值：

个体学习器的加权分歧值：

于是，得到集成的泛化误差为：

个体学习器的准确性越高，多样性越好，则集成越好，但实际任务中很难直接优化。

### 1.5.2多样性增强

数据样本扰动：给定初始数据集，可产生出不同的数据子集，利用不同的数据子集训练出不同的个体学习器。通常基于采样法，如Bagging中使用自助采样，而在AdaBoost中使用序列采样。

输入属性扰动：训练子集通常由一组属性描述，不同的属性子集提供了观察数据的不同视角。从不同的子空间训练出的个体学习器必然会不同。在数据只包含少量属性，或者冗余属性很少时，不宜使用属性扰动法。

输出属性扰动：对输出表示进行操纵以增强多样性，可对训练样本的类标记稍作标记，也可对输出表示进行转化，还可将原任务拆解为多个同时求解的子任务。

算法参数扰动：基学习算法一般都有参数进行设置，通过随机设置不同的参数，往往可产生差别较大的个体学习器。

不同的多样性增强机制可同时使用，如随机森林中同时使用了数据样本扰动和输入属性扰动。

# 2.weka算法实现

## 2.1数据集的准备

1、UCI完备数据集：magic04.csv

2、UCI不完备数据集：Annealing.csv

3、UCI不平衡数据集： abalone.csv

## 2.2 AdboostM1

选择五折交叉数据验证，选定classifier=DecisionStump分类器不断调整numlterations参数（其在weka中默认值为10，即会训练迭代提升DecisionStump分类器十次），进行数据测试，结果如下：

当numlterations= 5时：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 数据集 | Precision | Recall值 | F值 | AUC值 | Correctly | Time |
| magic04 | 0.766 | 0.771 | 0.763 | 0.816 | 77.11% | 1.18s |
| annealing | 0.703 | 0.837 | 0.764 | 0.895 | 83.71% | 0.05s |
| abalone | 0.085 | 0.210 | 0.120 | 0.638 | 21.02% | 0.05s |

当numlterations= 10时：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 数据集 | Precision | Recall值 | F值 | AUC值 | Correctly | Time |
| magic04 | 0.790 | 0.790 | 0.790 | 0.849 | 78.96% | 2.45s |
| annealing | 0.703 | 0.837 | 0.764 | 0.895 | 83.71% | 0.03s |
| abalone | 0.085 | 0.210 | 0.120 | 0.638 | 21.02% | 0.14s |

当numlterations= 15时：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 数据集 | Precision | Recall值 | F值 | AUC值 | Correctly | Time |
| magic04 | 0.829 | 0.812 | 0.809 | 0.861 | 81.25% | 3.36s |
| annealing | 0.703 | 0.837 | 0.764 | 0.895 | 83.71% | 0.05s |
| abalone | 0.85 | 0.210 | 0.120 | 0.638 | 21.02% | 0.05s |

上周使用了UCI不平衡数据集： abalone.csv，跑出来效果不好，这次又使用了集成学习算法来跑这个数据集。从跑出来的数据结果来看，abalone.csv的效果还是最差的，效果最好的是annealing数据集。

通过更改参数numlterations值，即训练迭代提升DecisionStump分类器次数，在参数numlterations值变大时，迭代次数增加，建模时间会增加。此外，调整参数对magic04训练集效果有很大改善，而基本不影响abalone.csv和annealing.csv。

此外，AdboostM1还可以尝试更改基学习器，即可更改classifier分类器的种类，weka中提供了J48、M5P等等。

## 2.3 Bagging

选择五折交叉数据验证，选定numlterations参数=10（其在weka中默认值为10，即会训练迭代提升DecisionStump分类器十次），不断调整classifier分类器进行数据测试，结果如下：

当classifier=DecisionStump时：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 数据集 | Precision | Recall值 | F值 | AUC值 | Correctly | Time |
| magic04 | 0.790 | 0.790 | 0.790 | 0.849 | 78.96% | 2.45s |
| annealing | 0.703 | 0.837 | 0.764 | 0.895 | 83.71% | 0.03s |
| abalone | 0.085 | 0.210 | 0.120 | 0.638 | 21.02% | 0.14s |

当classifier=RandomTree时：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 数据集 | Precision | Recall值 | F值 | AUC值 | Correctly | Time |
| magic04 | 0.871 | 0.870 | 0.867 | 0.917 | 86.99% | 4.5s |
| annealing | 0.939 | 0.940 | 0.939 | 0.983 | 93.99% | 0.16s |
| abalone | 0.216 | 0.235 | 0.223 | 0.672 | 23.49% | 0.78s |

当classifier=J48时：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 数据集 | Precision | Recall值 | F值 | AUC值 | Correctly | Time |
| magic04 | 0.869 | 0.870 | 0.868 | 0.922 | 87.00% | 31.16s |
| annealing | 0.921 | 0.922 | 0.921 | 0.980 | 92.23% | 0.98s |
| abalone | 0.215 | 0.228 | 0.220 | 0.702 | 22.82% | 4.1s |

从实验结果可以知道，在调整分类器后，annealing和abalone在RandomTree分类器下结果最佳，而magic04则在J48分类器下结果最佳。数据集在J48下最花费时间。

此外，还可以尝试更改numlterations参数，改变训练迭代分类器次数。

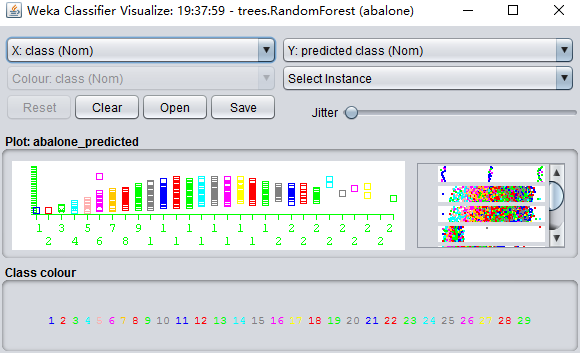
## 2.4 RandomForest

选择五折交叉数据验证，进行数据测试，结果如下：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 数据集 | Precision | Recall值 | F值 | AUC值 | Correctly | Time |
| magic04 | 0.880 | 0.880 | 0.815 | 0.878 | 80.99% | 30.27s |
| annealing | 0.935 | 0.935 | 0.934 | 0.985 | 93.48% | 1.39s |
| abalone | 0.225 | 0.242 | 0.232 | 0.735 | 20.47% | 6.71s |

从实验结果来看，数据集annealing.csv效果最佳，abalone.csv仍然不好。RandomForest构建决策树过程花费时间较长，因此建模时间总体很长，加上五折交叉验证，时间耗费比较大。

Classifier Visualize：



# 3.本周学习心得体会

本周学习了《机器学习 周志华版》中的集成学习算法的理论知识。分别就集成学习中

Adboost算法、Bagging算法、Random Forest算法进行了学习。Adboost算法利用基学习器的线性组合，Bagging算法利用自助采样，Random Forest算法利用在决策树训练过程中加入随机属性选择。总的来说，公式推导过程很难。