## 第八章 集成学习

8.1 什么是集成学习？为什么集成学习被应用？

参考答案：集成学习是为了解决特定的计算程序，如分类器或专家知识等多种模式，进行战略性生产和组合。集成学习能提高模型的分类，预测，函数逼近等方面的精度。

8.2 什么是集成方法的一般原则，在集成方法中套袋（bagging）和爆发（boosting）指的是什么？

参考答案：集成方法的一般原则是要结合给定的学习算法多种预测模型，相对于单一模型，其有更强的健壮性。套袋是一种能提高易变的预测或分类方案集成方法。爆发方法被依次用来减少组合模型的偏差。爆发和装袋都可以通过降低方差减少误差。

8.3 简述Boosting、Bagging和“随机森林”的学习过程。

参考答案：Boosting是一种个体学习器间存在强依赖关系、必须串行生成的序列化方法。其基本思想是：增加前一个基学习器在训练训练过程中预测错误样本的权重，使得先前基学习器做错的训练样本在后续受到更多关注，然后基于调整后的分布训练下一个基学习器，一直向下串行直至产生需要的T个基学习器，Boosting最终将这T个学习器进行加权结合。

Bagging是一种个体学习器间不存在强依赖关系、可同时生成的并行化方法。Bagging使用“有放回”采样方式选取训练集，对于包含m个样本的训练集，进行m次有放回的随机采样操作，从而得到m个样本的采样集。照这样，我们就可采样出T个含m个样本的采样集，然后基于每个采样集训练出一个基学习器，最终将这T个基学习器进行结合。

随机森林（Random Forest）是Bagging的一个拓展变体。RF在以决策树为基学习器构建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。即在基决策树的训练过程中，在选择划分属性时，RF先从候选属性集中随机挑选出一个包含K个属性的子集，再从这个子集中选择一个最优划分属性。

8.4 如何增强集成学习中个体学习器的多样性？

参考答案：（1）数据样本扰动，即利用具有差异的数据集来训练不同的基学习器。例如：有放回自助采样法，但此类做法只对那些不稳定学习算法十分有效，例如：决策树和神经网络等，训练集的稍微改变能导致学习器的显著变动。

（2）输入属性扰动，即随机选取原空间的一个子空间来训练基学习器。例如：随机森林，从初始属性集中抽取子集，再基于每个子集来训练基学习器。但若训练集只包含少量属性，则不宜使用属性扰动。

（3）输出表示扰动，此类做法可对训练样本的类标稍作变动，或对基学习器的输出进行转化。

（4）算法参数扰动，通过随机设置不同的参数，例如：神经网络中，随机初始化权重与随机设置隐含层节点数。

8.5 分析Bagging通常为何难以提升朴素贝叶斯分类器的性能

参考答案：Bagging主要是降低分类器的方差，而朴素贝叶斯分类器没有方差可以减小。对全训练样本生成的朴素贝叶斯分类器是最优的分类器，不能用随机抽样来提高泛化性能。

8.6 随机森林为何比决策树Bagging集成的训练速度更快

参考答案：随机森林不仅会随机样本，还会在所有样本属性中随机几种出来计算。这样每次生成分类器时都是对部分属性计算最优，速度会比Bagging计算全属性要快。

8.7 GradientBoosting是一种常用的Boosting算法，试分析其与AdaBoost的异同。

参考答案：GradientBoosting与AdaBoost相同的地方在于要生成多个分类器以及每个分类器都有一个权值，最后将所有分类器加权结合起来。

不同在于： AdaBoost通过每个分类器的分类结果改变每个样本的权值用于新的分类器和生成权值，但每个样本不会改变。 GradientBoosting将每个分类器对样本的预测值与真实值的差值传入下一个分类器来生成新的分类器和权值(这个差值就是下降方向)，而每个样本的权值不变。

8.8 MultiBoosting算法与Iterative Bagging的优缺点。

参考答案：MultiBoosting由于集合了Bagging，Wagging，AdaBoost，可以有效的降低误差和方差，特别是误差。但是训练成本和预测成本都会显著增加。 Iterative Bagging相比Bagging会降低误差，但是方差上升。由于Bagging本身就是一种降低方差的算法，所以Iterative Bagging相当于Bagging与单分类器的折中。

8.9 试设计一种能提升k近邻分类器性能的集成学习算法。

参考答案：可以使用Bagging来提升k近邻分类器的性能，每次随机抽样出一个子样本，并训练一个k近邻分类器，对测试样本进行分类。最终取最多的一种分类。

8.10 为什么xgboost要用泰勒展开，优点是什么？

参考答案：xgboost使用了一阶和二阶偏导，二阶导数有利于梯度下降的更快更准，使用泰勒展开取得函数做自变量的二阶导数形式，可以在不选定损失函数具体形式的情况下，仅仅依靠输入数据的值就可以进行叶子分裂优化计算，本质上也就把损失函数的选取和模型算法优化/参数选择分开了。这种去耦合增加了xgboost的适用性，使得它按需选取损失函数，可以用于分类，也可以用于回归.

8.11 xgboost如何寻找最优特征？是有放回还是无放回？

参考答案：xgboost在训练的过程中给出各个特征的增益评分，最大增益的特征会被选出来作为分裂依据，从而记忆了每个特征对在模型训练时的重要性——从根到叶子中间节点涉及某特征的次数作为该特征重要性排序。

xgboost属于Boosting集成学习方法，样本是不放回的，因而每轮计算样本不重复。另一方面，xgboost支持子采样，也就是每轮计算可以不使用全部样本，以减少过拟合。进一步的，xgboost还有列采样，每轮计算按百分比随机采样一部分特征，既提高计算速度又减少过拟合.

8.12 bootstrap数据的抽样过程？

参考答案：bootstrap的过程是有放回地从总共N个样本中抽样n个样本。即对样本进行有放回的抽样，抽样次数等同于样本总数。

8.13 什么是OOB？随机森林中OOB是如何计算的，它有什么优缺点？

参考答案：bagging方法中Bootstrap每次约有1/3的样本不会出现在Bootstrap所采集的样本集合中，也就没有参加决策树的建立，把这1/3的数据称为袋外数据oob（out of bag）,它可以用于取代测试集误差估计方法。

袋外数据(oob)误差的计算方法如下：

对于已经生成的随机森林,用袋外数据测试其性能,假设袋外数据总数为O,用这O个袋外数据作为输入,带进之前已经生成的随机森林分类器,分类器会给出O个数据相应的分类,因为这O条数据的类型是已知的,则用正确的分类与随机森林分类器的结果进行比较,统计随机森林分类器分类错误的数目,设为X,则袋外数据误差大小=X/O;这已经经过证明是无偏估计的,所以在随机森林算法中不需要再进行交叉验证或者单独的测试集来获取测试集误差的无偏估计。

8.14 给定如表8.1所示训练数据.假设弱分类器由或产生，其阈值v使该分类器在训练数据集上分类误差率最低。试用AdaBoost算法学习一个强分类器.

表8.1 训练数据表

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | | 1 | | 2 | | 3 | | 4 | | 5 | | 6 | | 7 | | 8 | | 9 | | 10 | |
| x | | 0 | | 1 | | 2 | | 3 | | 4 | | 5 | | 6 | | 7 | | 8 | | 9 | |
| y | | 1 | | 1 | | 1 | | -1 | | -1 | | -1 | | 1 | | 1 | | 1 | | -1 | |

参考答案：

初始化数据权值分布

对m=1，

(a)在权值分布为的训练数据上，阈值v取2.5时分类误差率最低，故基本分类器为

(b)在训练数据集上的误差率

(c)计算

(d)更新训练数据的权值分布：

分类器sign[]在训练数据集上有3个误分类点.

对m=2，

(a)在权值分布为的训练数据上，阈值v是8.5时分类误差率最低，基本分类器为

(b)在训练数据集上的误差率

(c)计算

(d)更新训练数据权值分布：

分类器sign[]在训练数据集上有3个误分类点.

对m=3，

(a)在权值分布为的训练数据上，阈值v是5.5时分类误差率最低，基本分类器为

(b)在训练样本集上的误差率

(c)计算

(d)更新训练数据权值分布：

于是得到：

分类器sign[]在训练数据集上误分类点个数为0.

于是最终分类器为

8.15 已知如表8.2所示的训练数据，x的取值范围为区间[0.5,10.5],y的取值范围为区间[5.0,10.0]，学习这个回归问题的提升树模型，考虑只用树桩作为基函数.

表8.2 训练数据表

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | 1 | | 2 | | 3 | | 4 | | 5 | | 6 | | 7 | | 8 | | 9 | | 10 | |
|  | | 5.56 | | 5.70 | | 5.91 | | 6.40 | | 6.80 | | 7.05 | | 8.90 | | 8.70 | | 9.00 | | 9.05 | |

参考答案： 第1步求即回归树

首先通过以下优化问题：



求解训练数据的切分点s：



容易求得在内部使平方损失误差达到最小值的为



这里是的样本点数

求训练数据的切分点。根据所给数据，考虑如下切分点：

1.5，2.5，3.5，4.5，5.5，6.5，7.5，8.5，9.5

对各切分点，不难求出相应的及



例如，当s=1.5时，,



现将s及m(s)的计算结果列表如下（见表8.3）

表8.3 计算数据表

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| s | | 1.5 | | 2.5 | | 3.5 | | 4.5 | | 5.5 | | 6.5 | | 7.5 | | 8.5 | | 9.5 | |
| m(s) | | 15.72 | | 12.07 | | 8.36 | | 5.78 | | 3.91 | | 1.93 | | 8.01 | | 11.73 | | 15.74 | |

由上表可知，当s=6.5时m(s)达到最小值，此时

,所以会归树为

用拟合训练数据的残差见表8.4，表中

表8.4 残差表

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | 1 | | 2 | | 3 | | 4 | | 5 | | 6 | | 7 | | 8 | | 9 | | 10 | |
|  | | -0.68 | | -0.54 | | -0.33 | | 0.16 | | 0.56 | | 0.81 | | -0.01 | | -0.21 | | 0.09 | | 0.14 | |

用拟合训练数据的平方损失误差：



第2步求.方法与求一样，只是拟合的数据是表8.4的残差。可以得到：

用拟合训练数据的平方损失误差是



继续求得

用拟合训练数据的平方损失误差是



假设此时已满足误差要求，那么即为所求提升树.