研究生算法课课堂笔记

上课日期：2016/11/10　　第(1)节课

组长学号及姓名：1601111267 柯伟辰

组员学号及姓名：1601214537 杜仑

1601214564 王韵

内容概要

第一节课主要讲授的内容总结如下：

（1） 总体介绍 Ensemble Learning

a) 主要挑战：如何保证个体学习器的多样性和准确度

b) Ensemble Learning 的主要方法

（2） 详细介绍 Random Forest

a) 数据扰动方法：Bagging

b) 特征扰动方法：Random Subspace

c) 基于 OOB 交叉验证的注意事项

d) 连续属性的切分点查找的优化方法

（3） 为 AdaBoost 的介绍开头

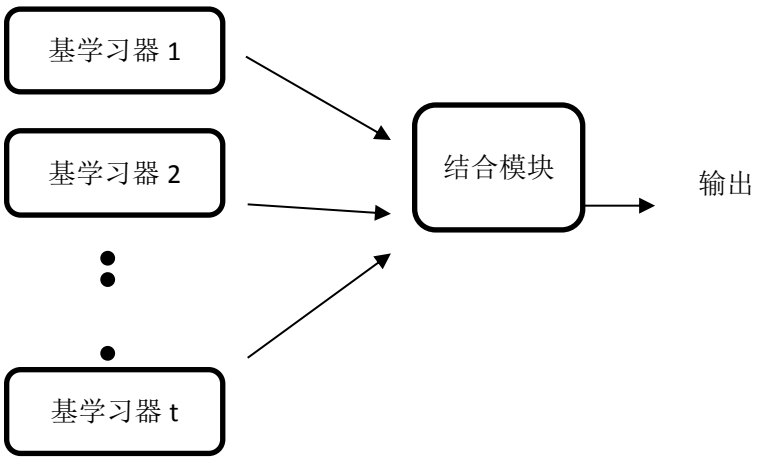
a) Boosting 简介

b) Voting 方法小结

详细内容

● 集成学习（Ensemble Learning）

1. Ensemble Learning 定义：通过构建并结合多个学习器来完成学习任务[①]。（如图 1 所示）集成学习通过将多个学习器进行结合，常可获得比单一学习器显著优越的泛化性能。在 Ensemble Learning 中，最常用的基学习器是树。



[①] 引自参考文献[1] 172 页

图 1 集成学习示意图

2. Ensemble Learning 的主要挑战是：基学习器的选择，包括：
   a) 基学习器的多样性
   b) 基学习器的准确性

事实上，基学习器的“准确性”和“多样性”本身就存在冲突，一般的，准确性很高以后，要增加多样性就需要牺牲准确性。如何产生并结合“好而不同”的基学习器是 Ensemble Learning 研究的核心。

例如，如果我们使用原始训练数据的采样来训练基学习器的话，随着采样率的下降，每个基学习器的多样性增强了，但是准确性减弱了。值得一提的是，在 Big Data 时代这个方法变得非常实用。Big Data 的原始训练数据是足够大的，在分布式环境里面，只要用训练数据的一份采样填满每台机器的内存，每台机器训练出一个模型，最后所有机器一起投票，就是天然的一种 Ensemble Learning 的做法。

3. 根据基学习器的生成方式，集成学习方法大致可以分为两类：
   a) 基学习器之间存在强依赖关系，并且串行生成。代表方法 Boosting
   b) 基学习器之间不存在强依赖关系，可并行生成。代表方法 Random Forest

Question：Ensemble Learning 有可能比准确率最高的子学习器好吗？为什么？
Answer：可能。如下例所示，O 表示正确分类，X 表示错误分类。每个子学习器的分类正确率为 66.7%，但 Uniform Voting 后的正确率为 100%。

	样本 1	样本 2	样本 3
基学习器 1	O	O	X
基学习器 2	O	X	O
基学习器 3	X	O	O

但是，这样的结果想要达成也是有条件的。如果采用这种一人一票的 Uniform Voting 的方法的话，要满足下面两个条件才会获得比较好的效果：
1）每个基学习器的分类正确率都大于 50%，投票结果集成后才是有效的。
2）各个基学习器之间具有足够好的多样性，同时分类正确率差异不能太大。

● 随机森林（Random Forest）

Random Forest 通过 Bagging(Bootstrap Aggregation)和 Random Subspace 两种策略分别从数据维度和特征维度对数据集进行扰动，解决基学习器的多样性挑战。

1. 数据扰动方法 Bagging

在集成学习中，为了使基学习器具有多样性，一种可能的做法就是对训练样本进行采样。

（1） 自助采样法（Bootstrap Sampling）

给定包含 m 个样本的原始数据集 D，我们对该样本数据集进行 m 次有放回抽样得到 D‘，D’为自助采样结果。显然，D 中有一部分样本会在 D‘中多次出现，而另一部分样本不出现。做一个简单估计，样本在 m 次采样中始终不被采到的概率是$\left(1-\frac{1}{m}\right)^m$,取极限得到

$$\lim_{m\to\infty}\left(1-\frac{1}{m}\right)^m \to \frac{1}{e} \approx 0.368$$

即通过自助采样，初始数据集 D 中约有 36.8%的样本未出现在采样数据集 D‘中。

（2） Bagging 算法框架

我们可以采用自助采样法采样出 T 个含有 m 个训练样本的样本集，然后基于每个采样集训练出一个基学习器，再将这些基学习器进行结合，其中对分类任务常使用简单投票法，对回归任务常使用简单平均法。Bagging 算法描述如算法 1。

输入：训练集 D={(x1, y1), (x2,y2),…, (xn, yn)};
基学习算法 F
训练轮数 T

过程：

1: for t=1,2,…,T do

2: ht = F(D,D’)

3: end for

输出：$\mathbf{H}(\mathbf{x}) = \operatorname{argmax}_{y\in Y}\sum_{t=1}^{T} \mathrm{II}\,(\mathrm{ht}(\mathbf{x}) = \mathrm{y})$ ②

算法 1: Bagging 算法③

2. 特征扰动方法 Random Subspace

随机森林（Random Forest）是 Bagging 的一个扩展变体，在以决策树为基学习器构建的 Bagging 集成基础上，进一步在决策树的训练过程中引入了特征采样。特征采样方法总结如下：

（1） 直接采样：

采样方式 1: 建树之前先采样特征集，即同一棵决策树中每个结点对应的可

② Ⅱ(•) 为指示函数，在•为真和假时分别取值为 0,1

③ 参考文献[1] 178 页

选子特征集相同。这部分特征集是在建树之前就事先选出来的。例如，建立树 A 时，采样了 20 个特征出来，树 A 只能在这 20 个特征里面选择切割点；建立树 B 时，再选 20 个特征出来，树 B 只能在这 20 个特征里面选择切割点，以此类推。

采样方式 2：建树时，实时为当前结点随机生成子特征集，即同一棵决策树中每个结点对应的可选子特征集不同。具体来讲，就是每一个结点只能在一部分特征上进行分割，另一部分特征不能用来分割。可以用来分割的特征是每个结点建立的时候采样得出的。例如树 A 的根节点可以从 20 个特征中选取切割点，切割之后生成的子节点要再重新采样 20 个特征，从这些新的采样特征中选择切割点。通常认为这种方法生成的模型随机性更好，从而多样性也就更好。

（2） 特征映射

将 n 维特征空间映射到 d 维特征空间（n>d），通常对应于一种特征的组合方式，如特征的线性组合方式可描述如下：

$$Z_{d\times 1}^{(i)} = w_{d\times n} X_{n\times 1}^{(i)}$$

其中$X^{(i)}$为原始特征空间下的第i个样本，$Z^{(i)}$为映射后特征空间下的第i个样本，w是投影矩阵。

举例：斜决策树（Oblique Decision tree）提出了一种特征线性组合的策略：

$$\mathrm{x_{new}} = \alpha x_i + (1-\alpha)x_j, 0 < \alpha < 1$$

其中x_i, x_j为原始特征空间下的任意两个特征，x_{new}为特征组合后的新特征，以x_{new}为特征的切分将不会平行于特征轴。

注意：在进行特征映射时，通常需要进行特征值的预处理，即 Feature Scaling 和 Mean Normalization，以消除量纲的影响。通常选用如下公式：

$$x_{norm} = \frac{x_{origin} - \mu}{\mathrm{std}(x)}$$

其中μ 为样本均值，std 为样本标准差。这么做的原因是在线性组合时如果没有进行这些预处理的话，数值比较小的特征会被数值比较大的特征“淹没”。例如一个特征 A 的取值是 0-4，另一个特征 B 的取值是 0-1000，那么 A 和 B 进行线性组合的时候，很容易就完全看不到 A 的影响了。

值得注意的是，特别是在需要比较特征之间关系的时候，这样的预处理是必要的，比如在 Linear Regression、Neural Network 等模型中。而在进行传统决策树构建的时候，由于每个特征是独立使用的，因此没有必要进行这样的预处理。

3. 基于 OOB 交叉验证的注意事项

OOB（Out Of Bag）：自助采样过程使得每个子学习器只使用了初始训练集中约 63.2%的样本，因此剩下的约 36.8%的样本是天然的验证集。

同理，对 Random Forest 而言，每棵树在训练时有 36.8%的样本没有被用到。需要注意的是，在测试时，一棵树只能为它没有“见过”的样本进行投票。因此在 Random Forest 中每棵树对应的验证集都不一样。

4. 连续属性切分点查找的优化方法

在训练具有连续属性的随机森林的时候，为了增强每个基学习器的多样性，常常使用随机化的方法。具体来讲，当按连续属性 Xi 对特征空间进行切分时，首先按 Xi 对原始样本集进行排序，之后有三种较为常用的随机化切分方法：

方法一：随机选一个切分点。

方法二：随机选多个切分点（比如 5 个）并从中选择一个信息增益（或 Gini 指数等其它指标）最大的作为最终切分点。

方法三：随机选择两个 y 值不相等的样本点 a, b, 然后在 a 和 b 之间随机选取一个值作为切分点。

Question: Random Forest 中的决策树是高一点比较好还是矮一点比较好？

Answer: Random Forest 中的决策树高一点比较好。高的树 Variance 相对较高，虽然可能泛化能力较弱，但是由于 Random Forest 这个方法本身是强 Bias 的，所以能在很大程度上消除每个模型高 Variance，低 Bias 的影响。相反，如果每个决策树都很低，表达能力都很弱，Bias 本身就高，那么和 Random Forest 结合起来只会雪上加霜，准确率会更加降低。

Question: 为什么 Random Forest 随机选一个切分点是合理的？

Answer: 随机选取切分点虽然降低了单棵树的准确度，但是增加了树的多样性。这是 Ensemble Learning 中多样性和准确性之间的 trade-off，牺牲单棵树准确度，以提升多样性。

● AdaBoost

1. AdaBoost 简介

AdaBoost 的全称是 Adaptive Boosting，它是一个可将弱学习器提升为强学习器的算法，其工作机制为：先从初始训练集训练出一个基学习器，再根据基学习器的表现对训练样本权重分布进行调整，使得之前基学习器做错的训练样本在后续获得更多的关注，然后基于调整后的样本分布来训练下一个基学习器。

2. Voting 方法小结：

Random Forest：uniform voting，即一个模型一票，每个模型票的效力是一样的；

AdaBoost：weighted voting，即模型之间有权重的不同，权重高的模型投票的效力较高。

一般来讲，几种模型的准确率有如下关系：
Boosting > Random Forest > 纯 Bagging > 纯 Decision tree。

● 参考文献

[1] 周志华. 机器学习[M]. 清华大学出版社, 2016.
[2] 李航. 统计学习方法[M]. 清华大学出版社, 2012.