Chapter 8 Ensemble Learning

Jinghao.Zhao

Abstract

This is the notes for ensemble learning, section 8.1 to 8.5, including formula and key points.

1 Individual and Ensemble

The structure of ensemble learning is that first generating a group of individual learner(base learner), then combining them based on some strategy. Individual learner should not bad than random picking($\geq 50\%$). There are two types of ensemble learning

1. 个体学习器之间有强依赖关系 $\Rightarrow$ 串行 $\Rightarrow$ Boosting or Random Forest.
2. 个体学习器之间不存在强依赖关系 $\Rightarrow$ 并行 $\Rightarrow$ Bagging.

2 Boosting

Boosting will train a base learner first, then change the distribution of sample which makes wrong classified samples draw more attention in next base learner. Boost will repeat this step until the number of base learner reach a predetermined quantity T.
Adaboost is one of the Boost algorithm based on additive model(加性模型). which is the linear addition

$$H(\mathbf{x}) = \sum_{t=1}^{T} \alpha_t h_t(\mathbf{x}) \tag{1}$$

Then minimize the loss by it. The algorithm is as following, support we have m samples for dataset D

1. Initial the sample weight distribution($\mathcal{D}_1$), always be $\frac{1}{m}$;
2. Training a base learner h_t based on the weight distribution($\mathcal{D}$);
3. Calculate the error ϵ_t of h_t, if $\epsilon_t > 0.5$ this base learner will be ignore;
4. Calculate the weight α_t of classifier h_t;

5. Update sample weight distribution $\mathcal{D}$, return step 2 and train next learner until the number of base learner is T.

Step 4 will calculated by

$$\alpha_t = \frac{1}{2} ln(\frac{1-\epsilon_t}{\epsilon_t}). \tag{2}$$

Step 5 will be calculated by

$$\begin{aligned} \mathcal{D}_{t+1}(x) = \frac{\mathcal{D}_t}{Z_t} \times \begin{cases} exp(-\alpha_t), \ if \ h_t(x) = f(x) \\ exp(-\alpha_t), \ if \ h_t(x) \neq f(x) \end{cases} \\ = \frac{\mathcal{D}_t \cdot e^{(-\alpha_t f(x) h_t(x))}}{Z_t} \end{aligned} \tag{3}$$

The loss function of $h_t(x)$ will be (8.9) and α_t can be calculated by (8.10). Boosting的关键是求出每个基分类器的权重以及每次分类后样本的分布权重，其推导过程如下:

$$l_{exp}(\alpha_t h_t | \mathcal{D}_t) = E_{x \sim \mathcal{D}}[exp(-f(x) \alpha h_t(x))], \tag{4}$$

其中$\mathcal{D}$为样本分布, f(x) 为样本真实值, $h_t(x)$为预测值, 二者都具有二值性, 对其求偏导，

$$\frac{\partial l_{exp}}{\partial \alpha_t} = e^{-\alpha_t}(1-\epsilon_t) + e^{\alpha_t}\epsilon_t, \tag{5}$$

其中ϵ_t是当前基分类器的错误率。令偏导(8.6)=0, 得

$$e^{-\alpha_t}(1-\epsilon_t) = e^{\alpha_t}\epsilon_t \tag{6}$$

解,得

$$\alpha_t = \frac{1}{2} ln(\frac{1-\epsilon_t}{\epsilon_t}) \tag{7}$$

这就是分类器权重的公式。

接下来调整样本分布，由于此前t-1个分类器结果已经成型，即$H_{t-1}(x)$固定，因此需要计算的是加入当前分类器t后的损失。由(8.12)以及指数函数e^x的泰勒展开式[1]可得(8.13)及(8.14). 由于$H_{t-1}(x)$固定，所以$E_{x \sim \mathcal{D}}[e^(-f(x)H_{t-1}(x)]$为常数。由(8.18)可知，$h_t(x)$是基于样本分布$\mathcal{D}_t$时应具有最小化的误差，且该误差应小于0.5。下面推导获得分布的过程。

$$\begin{aligned} \mathcal{D}_{t+1}(\mathbf{x}) = \frac{\mathcal{D}(x) e^{-f(x)H_t(x)}}{E_{x \sim \mathcal{D}}[e^{-f(x)H_t(x)}]}, H_t(x) = H_{t-1}(x) + \alpha_t h_t(x) \\ \Rightarrow \mathcal{D}_{t+1}(\mathbf{x}) = \mathcal{D}(x) \frac{e^{-f(x)(H_{t-1}(x) + \alpha_t h_t(x))}}{E_{x \sim \mathcal{D}}[e^{-f(x)H_t(x)}]} \\ = \mathcal{D}_t(\mathbf{x}) e^{-f(x)\alpha_t h_t(x)} \frac{E_{x \sim \mathcal{D}}[e^{-f(x)H_{t-1}(x)}]}{E_{x \sim \mathcal{D}}[e^{-f(x)H_t(x)}]} = (8.19) \end{aligned} \tag{8}$$

即算法中分布的更新公式，分数部分即为规范化因子Z_t。

Boosting方法主要关注降低偏差。

[1] $e^x = e^0 + \frac{e^1}{1!}x + \frac{e^2}{2!}x^2 + o(x^3)$

3 Bagging与随机森林

3.1 Bagging

Bagging是基于对样本重复采样的集成分类方法。首先使用自出采样法(Bootstrap sampling)[2]抽取出T个大小为m的子数据集，然后用基分类器对每个子集进行分类，最后依据投票法或者平均法进行分类及回归。对于剩余约36.8%未被抽取的样本，将其用于验证分类器泛化性能的验证集进行包外估计。

Bagging主要关注与降低方差。

3.2 Random Forest

随机森林是Bagging的变体，在以决策树为基分类器的基础上，增加了属性的随机选择。在随机森林中，每次选择分类属性时，首先从属性集合(假设有d个属性备选)中随机选择一个包含k(一般令$k = \log_2 d$)个属性的子集，再按照选择策略计算出最优属性。

4 结合策略

对于基分类器的分类结果，可以用不同的方法将其结合，如平均法，投票法和学习法。

4.1 平均法

平均法主要用于回归或者数值型的输出，分为简单平均法和加权平均法，假设有T个基分类器，

$$H(x) = \sum_{i=1}^{T} \omega_i h_i(x) \tag{9}$$

ω_i表示每个基分类器的权重，当ω均为1时表示简单平均法。权重一般通过训练数据学习而来。

4.2 投票法

投票法主要是根据基分类器中分类结果的占比来决定最终分类结果

1. 绝对多数投票法，当最多数类别占比超过50%时认为分类合理，否则拒绝分类。
2. 相对多数投票法，取比例最高的分类结果作为最后结果。
3. 加权投票法，$H(x) = c_{\underset{j}{\operatorname{argmax}} \sum_{i=1}^{T} \omega_i h_i^j(x)}$

[2]对样本有放回的抽取m个数据

4.3 学习法

利用另外的学习器来结合基分类器的输出。该学习器的输入为基分类器的输出，而标记等同于原始样本的标记。

5 多样性

由于个体学习器应尽量好而不同，即个体学习器的准确性越高，多样性越大，则集成的效果越好。给定数据集与分类结果，基于分类器的预测结果列联表，

	$h_i = +1$	$h_i = -1$
$h_i = +1$	a	c
$h_i = -1$	b	d

衡量多样性的指标有

1. 不合度量 $dis_{ij} = \frac{b+c}{m}$
2. 相关系数 $\rho_{ij} = \frac{ad-bc}{\sqrt{(a+b)(a+c)(c+d)(b+d)}}$
3. Q-统计量 $Q_{ij} = \frac{ad-bc}{ad+bc}$
4. κ-统计量 $\kappa = \frac{p_1-p_2}{1-p_2}$，其中p1表示两个分类器一致的概率，$p_1 = \frac{a+d}{m}$，p2表示分类器偶然一致的概率，$p_2 = \frac{(a+b)(a+c)+(c+d)(b+d)}{m^2}$

通过以下方法可以增强多样性

1. 数据样本扰动。基于采样法，产生不同的样本子集，训练不同的个体学习器。对决策树、神经网络有效，而贝叶斯、线性学习器、SVM对样本数据扰动不敏感，因此无效。
2. 输入属性扰动。从初始属性集中抽取若干个子集，基于这些子集训练基学习器。
3. 输出表示扰动。对输出表示进行扰动以增强多样性，有时将分类转换为回归进行学习。
4. 算法参数扰动。对于参数少的学习器，可以将其中某些环节用近似方法替代。例如决策树中的属性选取机制。