Boosting and Application to LTR

zjgtan

October 28, 2014

1 What is Boosting?

1.1 Additive Model and Boost

1.2 Classification and Regression Tree

Definition CART是一种典型的Tree-Based模型。Tree-Based Models将input space划分为cuboid regions，并为每一个region分配a simple model(a constance)。Tree-Based Models可以看做是一种model combination method，在input space的每一个点上只有一个model负责预测。

Prediction 对于一个x，我们通过遍历CART树，找到x所在的cuboid region。在每一个region中，独立的模型进行预测。

Growing a Tree 采用greedy optimization的策略构建CART模型：从根节点开始，在每次迭代中选择某一维变量及其threshold，最优的选择是使得裂变得到的cuboid regions可以为原region中的样本提供最优的预测。

1.2.1 Pruning

Method Pruning主要目的是控制CART的复杂度，防止Overfitting。一种简单的策略是当residual error的减少下降到某一阈值时，CART树的叶节点停止分裂。但是实践表明，刚开始的split残差可能没有得到很好的减少，但是可能多分类几次却出现了较大的优化。因为贪婪的建树策略只是取局部的最优解。因此，通常的：首先建立一个large tree，基于训练样本的数据量决定叶节点的数量；接着，基于平衡预测精度与模型复杂度的策略进行剪枝。

Criterion T_0表示原始树；叶节点有$\tau = 1, \ldots, |T|$，其中，$|T|$表示所有叶节点的数量；R_τ表示叶节点τ中的样本数量。
对于Regression，R_τ中的最优预测为区域中样本target的均值。

$$y_\tau = \frac{1}{N_\tau} \sum_{x_n \in R_\tau} t_n \tag{1}$$

因此，每个区域的residual sum-of-squares为

$$Q_\tau(T) = \sum_{x_n \in R_\tau} t_n - y_\tau{}^2 \tag{2}$$

对于Classification，$p_{\tau k}$表示RegionR_τ中k类样本的比例，通常有两种指标：
cross-entropy

$$Q_\tau(T)=\sum_{k=1}^{K}p_{\tau k}\ln p_{\tau k} \tag{3}$$

Gini index

$$Q_\tau(T)=\sum_{k=1}^{K}p_{\tau k}(1-p_{\tau k}) \tag{4}$$

当区域中样本类别越集中于某一类时，这两个指标越小，说明划分越好。

1.3 AdaBoost

1.3.1 AdaBoost算法框架

AdaBoost的算法框架如下表所示

Algorithm 1 AdaBoost

Initialization: 将数据权重w_n初始化为$w_n^{(1)}=\frac{1}{N}$，其中$n=1,\ldots,M$
Output: 最终得到预测模型

$$Y_M(x)=sign(\sum_{m=1}^{M}\alpha_m y_m(x)) \tag{5}$$

1: for each $m\in[1,M]$ do
2: Minimizing the weighted error function

$$J_m=\sum_{n=1}^{N}w_n^{(m)}I(y_m(x_n)\neq t_n) \tag{6}$$

3: 计算加权平均误差

$$\epsilon_m=\frac{\sum_{n=1}^{N}w_n^{(m)}I(y_m(x_n)\neq t_n)}{\sum_{n=1}^{N}w_n^{(m)}} \tag{7}$$

4: 有弱分类器的组合系数

$$\alpha_m=\ln\frac{1-\epsilon_m}{\epsilon_m} \tag{8}$$

5: 更新训练数据权重

$$w_n^{(m+1)}=w_n^{(m)}\exp\{\alpha_m I(y_m(x_n)\neq t_n)\} \tag{9}$$

6: end for

1.3.2 Interpretation: Minimization of an exponential error function

exponential error function:

$$E=\sum_{n=1}^{N}\exp\{-t_n f_m(x_n)\} \tag{10}$$

其中，$f_m(x_n)$定义为一组弱分类器的线性加权

$$f_m(x)=\frac{1}{m}\alpha_l y_l(x) \tag{11}$$

$t_n\in\{-1,1\}$，上述的exponential error function中，当t_n与$f_m(x)$)同号时，error function得到最小值。我们的目标是最小化exponential error function, with respect to α_l and $y_l(x)$

继续推导，我们将得到AdaBoost算法框架。我们不进行全局的误差最小化，而是在固定$y_1(x),\ldots,y_{m-1}(x)$和$\alpha_1,\ldots,\alpha_m-1$的情况下，寻找$\alpha_m$和$y_m(x)$。首先对error function进行分解

$$\begin{aligned}E&=\sum_{n=1}^{N}\exp\{-t_n f_{m-1}(x_n)-\frac{1}{2}t_n\alpha_m y_m(x_n)\}\\&=\sum_{n=1}^{N}w_n^{(m)}\exp\{-\frac{1}{2}t_n\alpha_m y_m(x_n)\}\end{aligned} \tag{12}$$

这里是关键的部分。将训练集划分为第m个弱分类器的正确分类集合m和误分类集合m

$$\begin{aligned}E=e^{-\frac{\alpha_m}{2}}\sum_{n\in T_m}+e^{\frac{\alpha_m}{2}}\sum_{n\in M_m}w_n^{(m)}\\=(e^{\frac{\alpha_m}{2}}-e^{-\frac{\alpha_m}{2}})\sum_{n=1}^{N}w_n^{(m)}I(y_m(x_n)\neq t_n)+e^{-\frac{\alpha_m}{2}}\sum_{n=1}^{N}w_n^{(m)}\end{aligned} \tag{13}$$

继续将上式分解，目标是得到2-5式的结果。首先对$y_m(x_n)$进行最小化，9中第二项为常量，仅有第一项与$y_m(x_n)$有关，并且与AdaBoost算法框架的2式相同，因此优化2式就是对$y_m(x_n)$进行优化。

继而对α_m进行优化，简单偏导，十分易得。

最后是对权重进行更新。有公式$t_n y_m(x_n)=1-2I(y_m(x_n)\neq t_n)$容易得到结果。

2 Gradient Boosting

2.1 Numerical Optimization

2.2 Gradient Boosting

2.3 Application

3 Application: Learning to rank

3.1 RankBoost

3.2 GBRank