GBDT多分类

Softmax回归的对数损失函数

当使用逻辑回归处理多标签的分类问题时，如果一个样本只对应于一个标签，我们可以假设每个样本属于不同标签的概率服从于几何分布，使用多项逻辑回归（Softmax Regression）来进行分类：

$$P(Y=y_i|x)=h_\theta(x)\begin{bmatrix}P(Y=1|x;\theta)\\P(Y=2|x;\theta)\\.\\.\\.\\P(Y=k|x;\theta)\end{bmatrix}$$

$$=\frac{1}{\sum_{j=1}^{k}e^{\theta_j^T x}}\begin{bmatrix}e^{\theta_1^T x}\\e^{\theta_2^T x}\\.\\.\\.\\e^{\theta_k^T x}\end{bmatrix}$$

当存在样本可能属于多个标签的情况时，我们可以训练 k 个二分类的逻辑回归分类器。第 i 个分类器用以区分每个样本是否可以归为第 i 类，训练该分类器时，需要把标签重新整理为"第 i 类标签"与"非第 i 类标签"两类。通过这样的办法，我们就解决了每个样本可能拥有多个标签的情况。

在二分类的逻辑回归中，对输入样本 x 分类结果为类别1和0的概率可以写成下列形式：

$$P(Y=y|x;\theta)=(h_\theta(x))^y(1-h_\theta(x))^{1-y}$$

其中，$h_\theta(x)=\frac{1}{1+e^{-\theta^T x}}$ 是模型预测的概率值，y 是样本对应的类标签。

将问题泛化为更一般的多分类情况：

$$P(Y=y_i|x;\theta)=\prod_{i=1}^{K}P(y_i|x)^{y_i}=\prod_{i=1}^{K}h_\theta(x)^{y_i}$$

由于连乘可能导致最终结果接近0的问题，一般对似然函数取对数的负数，变成最小化对数似然函数。

$$-logP(Y=y_i|x;\theta)=-log\prod_{i=1}^{K}P(y_i|x)^{y_i}=-\sum_{i=1}^{K}y_i log(h_\theta(x))$$

GBDT多分类原理

将GBDT应用于二分类问题需要考虑逻辑回归模型，同理，对于GBDT多分类问题则需要考虑以下Softmax模型：

$$P(y=1|x)=\frac{e^{F_1(x)}}{\sum_{i=1}^{k}e^{F_i(x)}}$$

$$P(y=2|x)=\frac{e^{F_2(x)}}{\sum_{i=1}^{k}e^{F_i(x)}}$$

$$\cdots\cdots$$

$$P(y=k|x)=\frac{e^{F_k(x)}}{\sum_{i=1}^{k}e^{F_i(x)}}$$

其中 $F_1\ldots F_k$ 是 k 个不同的CART回归树集成。每一轮的训练实际上是训练了 k 棵树去拟合softmax的每一个分支模型的负梯度。softmax模型的单样本损失函数为：

$$loss=-\sum_{i=1}^{k}y_i\log P(y_i|x)=-\sum_{i=1}^{k}y_i\log\frac{e^{F_i(x)}}{\sum_{j=1}^{k}e^{F_j(x)}}$$

这里的 $y_i\ (i=1\ldots k)$ 是样本label在k个类别上作one-hot编码之后的取值，只有一维为1，其余都是0。由以上表达式不难推导：

$$-\frac{\partial loss}{\partial F_i}=y_i-\frac{e^{F_i(x)}}{\sum_{j=1}^{k}e^{F_j(x)}}=y_i-p(y_i|x)$$

可见，这 k 棵树同样是拟合了样本的真实标签与预测概率之差，与GBDT二分类的过程非常类似。下图是Friedman在论文中对GBDT多分类给出的伪代码：

Algorithm 6: $\mathbf{L}_K$_TreeBoost

$F_{k0}(\mathbf{x})=0,\quad k=1,K$

For $m=1$ to M do:

  $p_k(\mathbf{x})=\exp(F_k(\mathbf{x}))/\sum_{l=1}^{K}\exp(F_l(\mathbf{x})),\quad k=1,K$

  For $k=1$ to K do:

    $\tilde{y}_{ik}=y_{ik}-p_k(\mathbf{x}_i),\quad i=1,N$

    $\{R_{jkm}\}_{j=1}^{J}=J\text{–terminal node }tree(\{\tilde{y}_{ik},\mathbf{x}_i\}_1^N)$

    $\gamma_{jkm}=\frac{K-1}{K}\frac{\sum_{\mathbf{x}_i\in R_{jkm}}\tilde{y}_{ik}}{\sum_{\mathbf{x}_i\in R_{jkm}}|\tilde{y}_{ik}|(1-|\tilde{y}_{ik}|)},\quad j=1,J$

    $F_{km}(\mathbf{x})=F_{k,m-1}(\mathbf{x})+\sum_{j=1}^{J}\gamma_{jkm}1(\mathbf{x}\in R_{jkm})$

  endFor

endFor

end Algorithm

http://blog.csdn.net/qq_22238533

根据上面的伪代码具体到多分类这个任务上面来，我们假设总体样本共有 K 类。来了一个样本 x ，我们需要使用GBDT来判断 x 属于样本的哪一类。

第一步我们在训练的时候，是针对样本 x 每个可能的类都训练一个分类回归树。举例说明，目前样本有三类，也就是 $K=3$ ，样本 x 属于第二类。那么针对该样本的分类标签，其实可以用一个三维向量 $[0,1,0]$ 来表示。0 表示样本不属于该类，1 表示样本属于该类。由于样本已经属于第二类了，所以第二类对应的向量维度为 1 ，其它位置为 0 。

针对样本有三类的情况，我们实质上在每轮训练的时候是同时训练三颗树。第一颗树针对样本 x 的第一类，输入为 $(x,0)$ 。第二颗树输入针对样本 x 的第二类，输入为 $(x,1)$ 。第三颗树针对样本 x 的第三类，输入为 $(x,0)$ 。这里每颗树的训练过程其实就CART树的生成过程。在此我们参照CART生成树的步骤即可解出三颗树，以及三颗树对 x 类别的预测值 $F_1(x),F_2(x),F_3(x)$, 那么在此类训练中，我们仿照多分类的逻辑回归 ，使用Softmax 来产生概率，则属于类别 1 的概率为：

$$p_1(x)=\frac{exp(F_1(x))}{\sum_{k=1}^{3}exp(F_k(x))}$$

并且我们可以针对类别 1 求出残差 $\tilde{y}_1=0-p_1(x)$ ；类别 2 求出残差 $\tilde{y}_2=0-p_2(x)$ ；类别 3 求出残差 $\tilde{y}_3=0-p_3(x)$ 。

然后开始第二轮训练，针对第一类输入为 $(x,\tilde{y}_1)$, 针对第二类输入为 $(x,\tilde{y}_2)$ ，针对第三类输入为 $(x,\tilde{y}_3)$ 。继续训练出三颗树。一直迭代M轮。每轮构建3颗树。

当 $K=3$ 时，我们其实应该有三个式子：

$$F_{1M}(x)=\sum_{m=1}^{M}c_{1m}I(x\epsilon R_{1m})$$

$$F_{2M}(x)=\sum_{m=1}^{M}c_{2m}I(x\epsilon R_{2m})$$

$$F_{3M}(x)=\sum_{m=1}^{M}c_{3m}I(x\epsilon R_{3m})$$

当训练完以后，新来一个样本 x_1 ，我们要预测该样本类别的时候，便可以有这三个式子产生三个值 F_{1M},F_{2M},F_{3M} 。样本属于某个类别的概率为：

$$p_i(x)=\frac{exp(F_{iM}(x))}{\sum_{k=1}^{3}exp(F_{kM}(x))}$$

个人理解:

训练过程:

1. 初始化F_0=0
2. 将一个样本x_i的标签y进行one-hot encoding(K维), 然后将one-hot向量分别作为标签值输入到K个分类器中
3. 根据F_i计算softmax概率值p, 同一轮的每棵树求得一个概率值
4. 根据损失函数对F_i的偏导公式计算负梯度度
5. 计算每个叶子节点的输出值, 并加到强分类器中,