Part D: 梯度提升树

1. 用于回归的GBDT

设数据集为$D=\{(X_1,y_1),\dots,(X_N,y_N)\}$，模型的损失函数为$L(y,\hat{y})$，现希望利用多棵回归决策树来进行模型集成：设第$m$轮时，已知前$m-1$轮中对第$i$个样本的集成输出为$F_{m-1}(X_i)$，则本轮的集成输出$\hat{y}_i$为

$$F_m(X_i)=F_{m-1}(X_i)+h_m(X_i)$$

其中，h_m是使得当前轮损失$\sum_{i=1}^{N}L(y_i,\hat{y}_i)$达到最小的决策树模型。

特别地，当$m=0$时，$F_0(X_i)=\arg\min_{\hat{y}}\sum_{i=1}^{N}L(y_i,\hat{y})$。

记第m轮的损失函数为

$$G(h_m)=\sum_{i=1}^{N}L(y_i,F_{m-1}(X_i)+h_m(X_i))$$

令上述损失最小化不同于一般的参数优化问题，我们需要优化的并不是某一组参数，而是要在所有决策树模型组成的函数空间中，找到一个h^*使得$G(h^*)$最小。因此我们不妨这样思考：学习一个决策树模型等价于对数据集$\tilde{D}=\{(X_1,h^*(X_1)),\dots,(X_N,h^*(X_N))\}$进行拟合，设$w_i=h^*(X_I)$，$\mathbf{w}=[w_1,\dots,w_N]$，此时的损失函数可改记为

$$G(\mathbf{w})=\sum_{i=1}^{N}L(y_i,F_{m-1}(X_i)+w_i)$$

由于只要我们获得最优的$\mathbf{w}$，就能拟合出第m轮相应的回归树，此时一个函数空间的优化问题已经被转换为了参数空间的优化问题，即对于样本i而言，最优参数为

$$w_i=\arg\min_{w}L(y_i,F_{m-1}(X_i)+w)$$

对于可微的损失函数L，由于当$\mathbf{w}=\mathbf{0}$时的损失就是第$m-1$轮预测产生的损失，因此我们只需要在$w_i=0$处进行一步梯度下降（若能保证合适的学习率大小）就能够获得使损失更小的w_i^*，而这个值正是我们决策树需要拟合的$h^*(X_i)$。 以损失函数$L(y,\hat{y})=\sqrt{|y-\hat{y}|}$为例，记残差为

$$r_i=y_i-F_{m-1}(X_i)$$

则实际损失为

$$L(w_i)=\sqrt{|r_i-w_i|}$$

根据在零点处的梯度下降可知：

$$\begin{aligned}w_i^*&=0-\frac{\partial L}{\partial w}\Big|_{w=0}\\&=-\frac{1}{2\sqrt{r_i}}sign(r_i)\end{aligned}$$

为了缓解模型的过拟合现象，我们需要引入学习率参数η来控制每轮的学习速度，即获得了由$\mathbf{w}^*$拟合的第m棵树h^*后，当前轮的输出结果为

$$\hat{y}_i=F_{m-1}(X_i)+\eta h_m^*(X_i)$$

对于上述的梯度下降过程，还可以从另一个等价的角度来观察：若设当前轮模型预测的输出值为$\tilde{w}_i=F_{m-1}(X_i)+w_i$，求解的问题即为

$$\tilde{w}_i=\arg\min_{\tilde{w}}L(y_i,\tilde{w})$$

由于当$\tilde{w}=F_{m-1}(X_i)$时，损失函数的值就是上一轮预测结果的损失值，因此只需将L在$\tilde{w}$在$\tilde{w}=F_{m-1}(X_i)$的位置进行梯度下降，此时当前轮的预测值应为

$$\tilde{w}_i^*=F_{m-1}(X_i)-\frac{\partial L}{\partial \tilde{w}}\Big|_{\tilde{w}=F_{m-1}(X_i)}$$

从而当前轮学习器h需要拟合的目标值w_i^*为

Contents

Print to PDF

【练习】对于均方损失函数和绝对值损失函数，请分别求出模型的初始预测F_0。

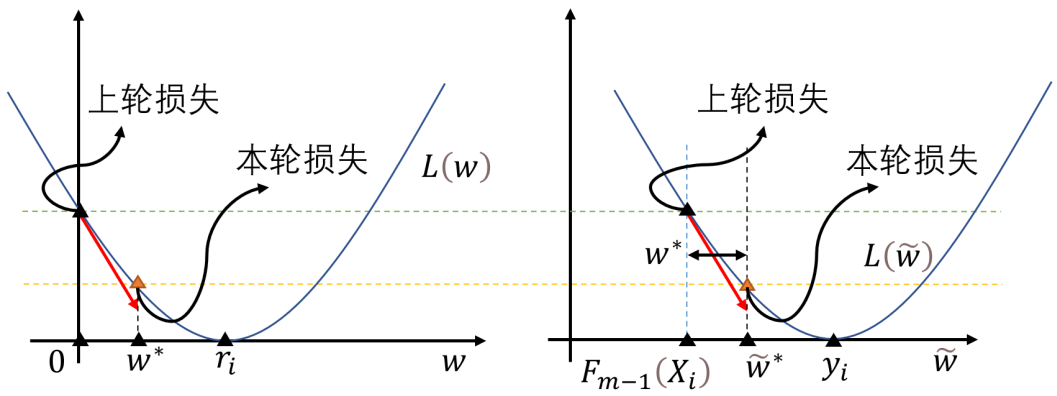
【练习】给定了上一轮的预测结果$F_{m-1}(X_i)$和样本标签y_i，请计算使用平方损失时需要拟合的w_i^*。

【练习】当样本i计算得到的残差$r_i=0$时，本例中的函数在$w=0$处不可导，请问当前轮应当如何处理模型输出?

【练习】除了梯度下降法之外，还可以使用牛顿法来逼近最值点。请叙述基于牛顿法的GBDT回归算法。

$$
\begin{aligned}
w_i^* &= \tilde{w}_i - F_{m-1}(X_i) \\
&= 0 - \frac{\partial L}{\partial w}\frac{\partial w}{\partial \tilde{w}}\Big|_{\tilde{w}=F_{m-1}(X_i)} \\
&= 0 - \frac{\partial L}{\partial w}\Big|_{\tilde{w}=F_{m-1}(X_i)} \\
&= 0 - \frac{\partial L}{\partial w}\Big|_{w=0}
\end{aligned}
$$

上述的结果与先前的梯度下降结果完全一致，事实上这两种观点在本质上没有任何区别，只是损失函数本身进行了平移，下图展示了它们之间的联系。



GBDT的特征重要性

在sklearn实现的GBDT中，特征重要性的计算方式与随机森林相同，即利用相对信息增益来度量单棵树上的各特征特征重要性，再通过对所有树产出的重要性得分进行简单平均来作为最终的特征重要性。

2. 用于分类的GBDT

CART树能够同时处理分类问题和回归问题，但是对于多棵CART进行分类任务的集成时，我们并不能将树的预测结果直接进行类别加和。在GBDT中，我们仍然使用回归树来处理分类问题，那此时拟合的对象和流程又是什么呢？

对于K分类问题，我们假设得到了K个得分$F_{1i},\dots,F_{Ki}$来代表样本i属于对应类别的相对可能性，那么在进行Softmax归一化后，就能够得到该样本属于这些类别的概率大小。其中，属于类别k的概率即为$\frac{e^{F_{ki}}}{\sum_{c=1}^{K}e^{F_{ci}}}$。此时，我们就能够使用多分类的交叉熵函数来计算模型损失，设$\mathbf{y}_i=[y_{1i},\dots,y_{Ki}]$为第$i$个样本的类别独热编码，记$\mathbf{F}_i=[F_{1i},\dots,F_{Ki}]$，则该样本的损失为

$$
L(\mathbf{y}_i,\mathbf{F}_i)=-\sum_{c=1}^{K}y_{ci}\log\frac{e^{F_{ci}}}{\sum_{\tilde{c}=1}^{K}e^{F_{\tilde{c}i}}}
$$

上述的K个得分可以由K棵回归树通过集成学习得到，树的生长目标正是使得上述的损失最小化。记第m轮中K棵树对第i个样本输出的得分为$\mathbf{h}_i^{(m)}=[h_{1i}^{(m)},\dots,h_{Ki}^{(m)}]$，则此时$\mathbf{F}_i^{(m)}=\mathbf{F}_i^{(m-1)}+\mathbf{h}_i^{(m)}$。与GBDT处理回归问题的思路同理，只需要令损失函数$L(\mathbf{y}_i,\mathbf{F}_i)$在$\mathbf{F}_i=\mathbf{F}_i^{(m-1)}$处梯度下降即可：

$$
\mathbf{F}_i^{*(m)}=\mathbf{F}_i^{(m-1)}-\frac{\partial L}{\partial \mathbf{F}_i}\Big|_{\mathbf{F}_i=\mathbf{F}_i^{(m-1)}}
$$

我们需要计算第二项中每一个梯度元素，即

$$
-\frac{\partial L}{\partial \mathbf{F}_i}\Big|_{\mathbf{F}_i=\mathbf{F}_i^{(m-1)}}=[-\frac{\partial L}{\partial F_{1i}}\Big|_{\mathbf{F}_i=\mathbf{F}_i^{(m-1)}},\dots,-\frac{\partial L}{\partial F_{Ki}}\Big|_{\mathbf{F}_i=\mathbf{F}_i^{(m-1)}}]
$$

对于第k个元素有

$$
\begin{aligned}
-\frac{\partial L}{\partial F_{ki}}\Big|_{\mathbf{F}_i=\mathbf{F}_i^{(m-1)}} &= \frac{\partial}{\partial F_{ki}}\sum_{c=1}^{K}y_{ci}\log\frac{e^{F_{ci}}}{\sum_{\tilde{c}=1}^{K}e^{F_{\tilde{c}i}}}\Big|_{\mathbf{F}_i=\mathbf{F}_i^{(m-1)}} \\
&= \frac{\partial}{\partial F_{ki}}\sum_{c=1}^{K}y_{ci}F_{ki}\Big|_{\mathbf{F}_i=\mathbf{F}_i^{(m-1)}} - \frac{\partial}{\partial F_{ki}}\sum_{c=1}^{K}y_{ci}\log[\sum_{\tilde{c}=1}^{K}e^{F_{\tilde{c}i}}]\Big|_{\mathbf{F}_i=\mathbf{F}_i^{(m-1)}} \\
&= y_{ki} - \frac{\partial}{\partial F_{ki}}\sum_{c=1}^{K}y_{ci}\log[\sum_{\tilde{c}=1}^{K}e^{F_{\tilde{c}i}}]\Big|_{\mathbf{F}_i=\mathbf{F}_i^{(m-1)}}
\end{aligned}
$$

由于在上式的第二项里，K个y_{ci}中只有一个为1，且其余为0，从而得到

$$
\begin{aligned}
-\frac{\partial L}{\partial F_{ki}}\Big|_{\mathbf{F}_i=\mathbf{F}_i^{(m-1)}} &= y_{ki} - \frac{\partial}{\partial F_{ki}}\log[\sum_{\tilde{c}=1}^{K}e^{F_{\tilde{c}i}}]\Big|_{\mathbf{F}_i=\mathbf{F}_i^{(m-1)}} \\
&= y_{ki} - \frac{e^{F_{ki}^{(m-1)}}}{\sum_{c=1}^{K}e^{F_{ci}^{(m-1)}}}
\end{aligned}
$$

此时，K棵回归树的学习目标为：

$$
\begin{aligned}
\mathbf{h}_i^{*(m)} &= \mathbf{F}_i^{*(m)} - \mathbf{F}_i^{(m-1)} \\
&= -\frac{\partial L}{\partial \mathbf{F}_i}\Big|_{\mathbf{F}_i=\mathbf{F}_i^{(m-1)}} \\
&= [y_{1i} - \frac{e^{F_{1i}^{(m-1)}}}{\sum_{c=1}^{K} e^{F_{ci}^{(m-1)}}}, \ldots, y_{Ki} - \frac{e^{F_{Ki}^{(m-1)}}}{\sum_{c=1}^{K} e^{F_{ci}^{(m-1)}}}]
\end{aligned}
$$

同时，为了减缓模型的过拟合现象，模型在第m轮实际的$\mathbf{F}_i^{*(m)}$为$\mathbf{F}_i^{(m-1)} + \eta \mathbf{h}_i^{*(m)}$。

由于每一轮都需要进行K棵树的拟合，因此GBDT在处理多分类时的速度较慢。事实上，我们可以利用概率和为1的性质，将K次拟合减少至$K-1$次拟合，这在处理类别数较少的分类问题时，特别是在处理二分类问题时，是非常有用的。

具体来说，此时我们需要$K-1$个得分，记为$F_{1i}, \ldots, F_{(K-1)i}$，则样本相应属于$K$个类别的概率值可表示为

$$
[\frac{e^{F1i}}{1+\sum_{c=1}^{K-1} e^{Fci}}, \ldots, \frac{e^{F(K-1)i}}{1+\sum_{c=1}^{K-1} e^{Fci}}, \frac{1}{1+\sum_{c=1}^{K-1} e^{Fci}}]
$$

当$K \geq 3$时，仍然使用独热编码来写出损失函数：

$$
L(F_{1i}, \ldots, F_{(K-1)i}) = y_{Ki} \log[1 + \sum_{c=1}^{K-1} e^{Fci}] - \sum_{c=1}^{K-1} y_{ci} \log \frac{e^{Fci}}{\sum_{c=1}^{K} e^{Fci}}
$$

类似地记$\mathbf{F}_i = [F_{1i}, \ldots, F_{(K-1)i}]$，我们可以求出负梯度：

【练习】请验证多分类负梯度的结果。

$$
-\frac{\partial L}{\partial F_{ki}}\Big|_{\mathbf{F}_i=\mathbf{F}_i^{(m-1)}} = \begin{cases} -\dfrac{e^{F_{ki}^{(m-1)}}}{\sum_{c=1}^{K-1} e^{F_{ci}^{(m-1)}}} & y_{Ki}=1 \\ y_{ki} - \dfrac{e^{F_{ki}^{(m-1)}}}{\sum_{c=1}^{K-1} e^{F_{ci}^{(m-1)}}} & y_{Ki}=0 \end{cases}
$$

当$K=2$时，不妨规定$y_i \in \{0,1\}$，此时损失函数可简化为

$$
L(F_i) = -y_i \log \frac{e^{F_i}}{1+e^{F_i}} - (1-y_i) \log \frac{1}{1+e^{F_i}}
$$

负梯度为

【练习】请验证二分类负梯度的结果。

$$
-\frac{\partial L}{\partial F_i}\Big|_{F_i=F_i^{(m-1)}} = y_i - \frac{e^{F_i}}{1+e^{F_i}}
$$

最后，我们可以使用各个类别在数据中的占比情况来初始化$\mathbf{F}^{(0)}$。具体地说，设各类别比例为$p_1, \ldots, p_K$（$K \geq 3$），我们希望初始模型的参数$F_1^{(0)}, \ldots, F_{K-1}^{(0)}$满足

$$
[\frac{e^{F_{1i}^{(0)}}}{1+\sum_{c=1}^{K-1} e^{F_{ci}^{(0)}}}, \ldots, \frac{e^{F_{(K-1)i}^{(0)}}}{1+\sum_{c=1}^{K-1} e^{F_{ci}^{(0)}}}, \frac{1}{1+\sum_{c=1}^{K-1} e^{F_{ci}^{(0)}}}] = [p_1, \ldots, p_{K-1}, p_K]
$$

对二分类（0-1分类）而言，设正负样本占比分别为p_1和p_0，则初始模型参数$F^{(0)}$应当满足

【练习】设二分类数据集中正样本比例为10%，请计算模型的初始参数$F^{(0)}$。

$$
[\frac{1}{1+e^{F_i^{(0)}}}, \frac{e^{F_i^{(0)}}}{1+e^{F_i^{(0)}}}] = [p_0, p_1]
$$

> **单调约束（Monotonic Constraints）**
>
> 有时我们会对某个特征或某些特征如何影响模型的输出有先验的知识，例如每天投入在学习的有效时间上越长就越有可能在考试中取得好的成绩，即有效学习时间长度和考试分数是一种单调增的约束关系。许多GBDT的实现（sklearn中的Histogram-Based GBDT、XGBoost和LightGBM）都提供了单调约束的参数选项，有关其在内部的实现原理可以参考本文。

3. XGBoost算法

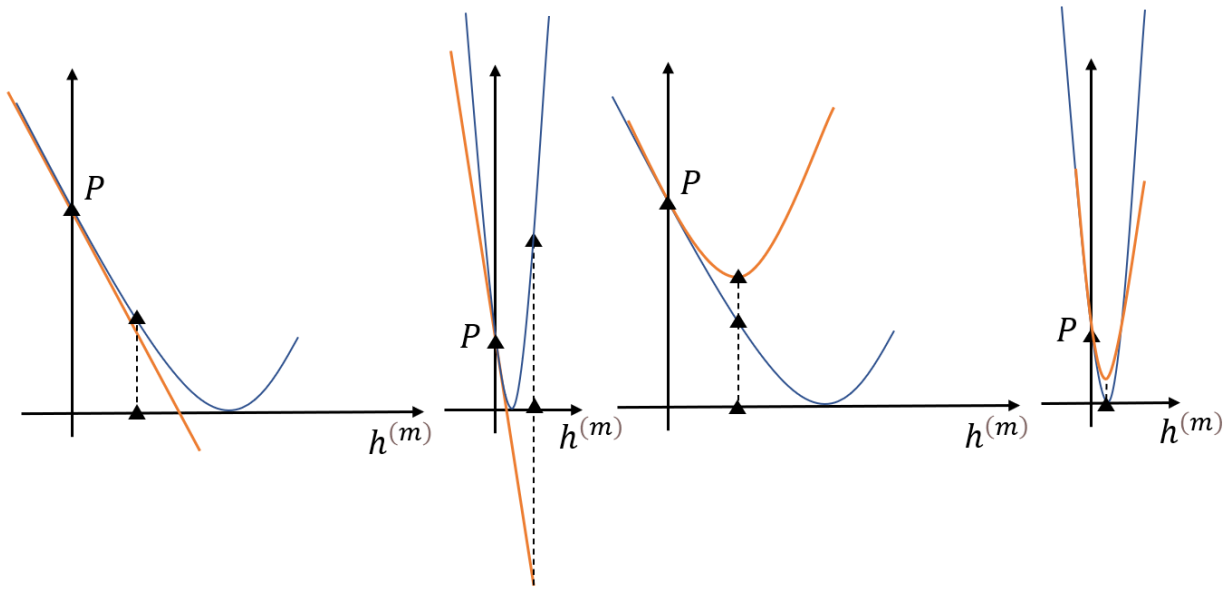
由于树模型较强的拟合能力，我们需要对模型进行正则约束来控制每轮模型学习的进度，除了学习率参数之外，XGBoost还引入了两项作用于损失函数的正则项：首先我们希望树的生长受到抑制而引入γT，其中的T为树的叶子节点个数，γ越大，树就越不容易生长；接着我们希望模型每次的拟合值较小而引入$\frac{1}{2}\lambda \sum_{i=1}^{T} w_i^2$，其中的$w_i$是回归树上第$i$个叶子结点的预测目标值。记第$m$轮中第$i$个样本在上一轮的预测值为$F_i^{(m-1)}$，本轮需要学习的树模型为$h^{(m)}$，此时的损失函数即为

$$
L^{(m)}(h^{(m)}) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j + \sum_{i=1}^{N} L(y_i, F_i^{(m-1)} + h^{(m)}(X_i))
$$

从参数空间的角度而言，损失即为

$$L^{(m)}(F_i^{(m)}) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j + \sum_{i=1}^{N} L(y_i, F_i^{(m)})$$

不同于上一节中GBDT的梯度下降方法，XGBoost直接在$h^{(m)} = 0$处（或$F_i^{(m)} = F_i^{(m-1)}$处）将损失函数近似为一个二次函数，从而直接将该二次函数的顶点坐标作为$h^{*(m)}(X_i)$的值，即具有更小的损失。梯度下降法只依赖损失的一阶导数，当损失的一阶导数变化较大时，使用一步梯度获得的$h^{*(m)}$估计很容易越过最优点，甚至使得损失变大（如子图2所示）；二次函数近似的方法需要同时利用一阶导数和二阶导数的信息，因此对于$h^{*(m)}$的估计在某些情况下会比梯度下降法的估计值更加准确，或说对各类损失函数更有自适应性（如子图3和子图4所示）。



为了得到$h^{*(m)}(X_i)$，记$h_i = h^{(m)}(X_i)$，$\mathbf{h} = [h_1, \ldots, h_N]$，我们需要先将损失函数显式地展开为一个关于$h^{(m)}(X_i)$的二次函数，：

$$\begin{aligned}
L^{(m)}(\mathbf{h}) &= \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j + \sum_{i=1}^{N} L(y_i, F_i^{(m-1)} + h_i) \\
&\approx \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j + \sum_{i=1}^{N} [L(y_i, F_i^{(m-1)}) + \frac{\partial L}{\partial h_i}\Big|_{h_i=0} h_i + \frac{1}{2}\frac{\partial^2 L}{\partial h_i^2}\Big|_{h_i=0} h_i^2] \\
&= \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j + \sum_{i=1}^{N} [\frac{\partial L}{\partial h_i}\Big|_{h_i=0} h_i + \frac{1}{2}\frac{\partial^2 L}{\partial h_i^2}\Big|_{h_i=0} h_i^2] + constant
\end{aligned}$$

由于近似后损失的第二项是按照叶子结点的编号来加和的，而第三项是按照样本编号来加和的，我们为了方便处理，不妨统一将第三项按照叶子结点的编号重排以统一形式。设叶子节点j上的样本编号集合为I_j，记$p_i = \frac{\partial L}{\partial h_i}\Big|_{h_i=0}$且$q_i = \frac{\partial^2 L}{\partial h_i^2}\Big|_{h_i=0}$，忽略常数项后有

【练习】请写出$L^{(m)}(F_i^{(m)})$在$F_i^{(m)} = F_i^{(m-1)}$处的二阶展开。

【练习】试说明不将损失函数展开至更高阶的原因。

【练习】请写出平方损失下的近似损失。

$$\begin{aligned}
\tilde{L}^{(m)}(\mathbf{h}) &= \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j + \sum_{i=1}^{N} [p_i h_i + \frac{1}{2} q_i h_i^2] \\
&= \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j + \sum_{j=1}^{T} [(\sum_{i \in I_j} p_i) w_j + \frac{1}{2}(\sum_{i \in I_j} q_i) w_i^2] \\
&= \gamma T + \sum_{j=1}^{T} [(\sum_{i \in I_j} p_i) w_j + \frac{1}{2}(\sum_{i \in I_j} q_i + \lambda) w_i^2] \\
&= \tilde{L}^{(m)}(\mathbf{w})
\end{aligned}$$

上式的第二个等号是由于同一个叶子节点上的模型输出一定相同，即I_j中样本对应的h_i一定都是w_j。此时，我们将损失统一为了关于叶子节点值$\mathbf{w} = [w_1, \ldots, w_T]$的二次函数，从而可以求得最优的输出值为

$$w_j^* = -\frac{\sum_{i \in I_j} p_i}{\sum_{i \in I_j} q_i + \lambda}$$

当前模型的近似损失（忽略常数项）即为

$$\begin{aligned}
\tilde{L}^{(m)}(\mathbf{w}^*) &= \gamma T + \sum_{j=1}^{T} [-\frac{(\sum_{i \in I_j} p_i)^2}{\sum_{i \in I_j} q_i + \lambda} + \frac{1}{2}\frac{(\sum_{i \in I_j} p_i)^2}{\sum_{i \in I_j} q_i + \lambda}] \\
&= \gamma T - \frac{1}{2}\sum_{j=1}^{T} \frac{(\sum_{i \in I_j} p_i)^2}{\sum_{i \in I_j} q_i + \lambda}
\end{aligned}$$

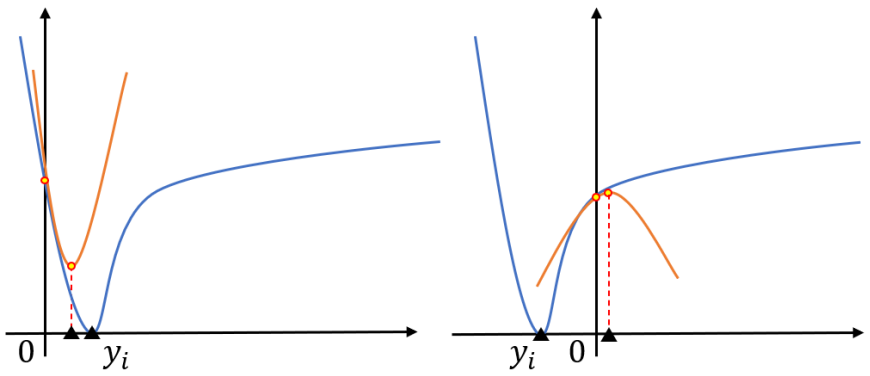
在决策树的一节中，我们曾以信息增益作为节点分裂行为操作的依据，信息增益本质上就是一种损失，增益越大即子节点的平均纯度越高，从而损失就越小。因此我们可以直接将上述的近似损失来作为分裂的依据，即选择使得损失减少得最多的特征及其分割点来进行节点分裂。由于对于某一个节点而言，分裂前后整棵树的损失变化只和该节点I及其左右子节点I_L与L_R的w^*值有关，此时分裂带来的近似损失减少量为

$$
\begin{aligned}
G &= [\gamma T - \frac{1}{2}\frac{(\sum_{i\in I} p_i)^2}{\sum_{i\in I} q_i + \lambda}] - [\gamma(T+1) - \frac{1}{2}\frac{(\sum_{i\in I_L} p_i)^2}{\sum_{i\in I_L} q_i + \lambda} - \frac{1}{2}\frac{(\sum_{i\in I_R} p_i)^2}{\sum_{i\in I_R} q_i + \lambda}] \\
&= \frac{1}{2}[\frac{(\sum_{i\in I_L} p_i)^2}{\sum_{i\in I_L} q_i + \lambda} + \frac{(\sum_{i\in I_R} p_i)^2}{\sum_{i\in I_R} q_i + \lambda} - \frac{(\sum_{i\in I} p_i)^2}{\sum_{i\in I} q_i + \lambda}] - \gamma
\end{aligned}
$$

模型应当选择使得G达到最大的特征和分割点进行分裂。

> **XGBoost的特值处理**
>
> XGBoost不支持分类变量处理，此处的特值是指稀疏值和缺失值，它们的处理方式类似：把0值或缺失值固定，先统一划分至左侧子节点，遍历非0值或非缺失值分割点进行不纯度计算，再统一划分至右侧子节点，又进行非0值或非缺失值分割点的遍历计算，从而得到当前节点当前特征的稀疏值或缺失值默认分配方向以及最佳分割点。特别的是，当训练时特征没有遇到缺失值但预测值出现时，它将会被分配给子节点样本数较多的一侧。

最后我们来重新回到单个样本的损失函数上：由于XGBoost使用的是二阶展开，为了保证函数在拐点处取到的是近似损失的最小值，需要满足二阶导数$q_i > 0$。当损失函数不满足此条件时，h_i^*反而会使得损失上升，即如下图中右侧的情况所示，而使用梯度下降法时并不会产生此问题。因此，我们应当选择在整个定义域上或在y_i临域上二阶导数恒正的损失函数，例如平方损失。



【练习】在下列的三个损失函数$L(y, \hat{y})$中，请选出一个不应作为XGBoost损失的函数并说明理由。

- Root Absolute Error: $\sqrt{|y-\hat{y}|}$
- Squared Log Error: $\frac{1}{2}[\log(\frac{y+1}{\hat{y}+1})]^2$
- Pseudo Huber Error: $\delta^2(\sqrt{1+(\frac{y-\hat{y}}{\delta})^2}-1)$

4. LightGBM算法

LightGBM的GBDT原理与XGBoost的二阶近似方法完全一致，并且在此基础上提出了两个新算法，它们分别是单边梯度采样（GOSS）以及互斥特征绑定（EFB）。

单边梯度采样

在GBDT中，计算出的梯度值绝对值越小则说明样本预测地越是准确，而梯度绝对值越大则说明样本预测的偏离程度越大，因此我们可以考虑对梯度绝对值小的样本进行抽样。具体说，对样本梯度绝对值排序后，先选出Top $a\%$梯度绝对值对应的样本，再从剩下$(1-a)$的样本中抽取$b\%$的样本（此处$b\%$是对于总样本的百分比）。此时，考虑基于均方损失的GBDT回归，记当前节点、左子节点、右子节点的梯度均值为$\bar{g}, \bar{g}_L, \bar{g}_R$，设特征及其分割点为$F, d$，原先的信息增益为

$$
\begin{aligned}
Gain(F,d) &= \frac{1}{N}[\sum_{i=1}^{N}(g_i - \bar{g})^2 - \sum_{i=1}^{N_L}(g_i^{(L)} - \bar{g}_L)^2 - \sum_{i=1}^{N_R}(g_i^{(R)} - \bar{g}_R)^2] \\
&= \frac{1}{N}[(\sum_{i=1}^{N} g_i^2 - N\bar{g}^2) - (\sum_{i=1}^{N_L} {g_i^{(L)}}^2 - N{\bar{g}_L}^2) - (\sum_{i=1}^{N_R} {g_i^{(R)}}^2 - N{\bar{g}_R}^2)] \\
&\propto \frac{1}{N}[\frac{(\sum_{i=1}^{N_L} g_i^{(L)})^2}{N_L} + \frac{(\sum_{i=1}^{N_R} g_i^{(R)})^2}{N_R}]
\end{aligned}
$$

记划分到左子节点对应的a部分样本为A_L、划分到左子节点对应的b部分抽样样本为B_L、划分到右子节点对应的a部分样本为A_R、划分到右子节点对应的b部分抽样样本为B_R。对于抽样部分的梯度和，我们使用$\frac{1-a}{b}$来进行补偿，例如原来从10个样本中划分6个为a部分，从剩下的4个中抽出两个为b部分，那么b部分的样本梯度和估计就是抽出两个样本的梯度和乘以$\frac{1-0.6}{0.2}$。因此，可以写出对应的$\tilde{Gain}(F,d)$为

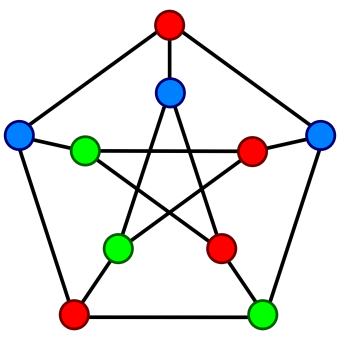
$$
\tilde{Gain}(F,d) = \frac{1}{N}[\frac{(\sum_{i\in A_L} g_i + \frac{1-a}{b}\sum_{i\in B_L} g_i)^2}{N_L} + \frac{(\sum_{i\in A_R} g_i + \frac{1-a}{b}\sum_{i\in B_R} g_i)^2}{N_R}]
$$

互斥特征绑定

实际的数据特征中可能有许多稀疏特征，即其非零值的数量远小于零值的数量，因此希望能够将这些特征进行合并来减少稀疏特征的数量，从而减少直方图构建的时间复杂度。我们将任意两个特征都不同时取非零值的特征集合称为一族互斥特征，数据集中的所有特征可被划分为这样的若干族互斥特征，例如下面就是一族互斥特征。

	特征1	特征2	特征3
样本1	0	1	0
样本2	-1	0	0
样本3	0	0	0

LightGBM提出了将互斥特征合并为单个特征的策略，从而让构建直方图的时间复杂度得以降低，因此需要找到最少的互斥绑定数量，即最少可以划分为几族。遗憾的是这个问题等价于图的着色问题，故它是NP-Hard的，目前并不存在多项式复杂度的解决方案，但我们可以通过近似方法来求解。为什么互斥特征绑定问题与图着色问题等价？如果我们把图的每一个顶点看做特征，将顶点之间是否存在边取决于两个特征是否存在同时为非零值的情况，若是则连接，那么此时没有边的顶点则代表他们之间满足互斥条件，将其涂上同种颜色作为同一族互斥特征，而寻找最少的绑定数量即是要寻找图的最少着色数。下图展示了Petersen图最少需要三种着色数。



【练习】请求出顶点最大度（即最多邻居数量）为d的无向图在最差和最好情况下需要多少种着色数，同时请构造对应的例子。

【练习】在最差情况下LightGBM会生成几族互斥特征？这种情况的发生需要满足什么条件？

在实际操作中，由于严格互斥的特征数量可能还并不算多，但是几乎互斥的特征数量却很多，若存在一个样本使得两个特征同时为非零值则称它们存在一次冲突，所谓几乎互斥即一族特征之间的冲突总数不超过给定的最大冲突数K，此时即使两个顶点之间存在边的连接，只要新加入的顶点能够使得这族特征满足几乎互斥的条件，那么就仍然可进行合并（或着相同颜色），如果此时新顶点与任意一族特征都不满足几乎互斥，那么将自身作为新的一族互斥特征集合的第一个元素（或着新的颜色）。

上述的讨论解决了特征绑定的问题，但我们只是将互斥特征放在了同一个集合里，还没有解决特征合并的问题。直观上说，我们需要用一个特征来表示多个特征时，要求新特征关于原特征族是可辨识的，即存在一一对应的关系。设需要合并的特征为$F_1,\dots,F_m$，它们对应的箱子分割点编号为$B_{i1},\dots,B_{ik_i}(i=1,\dots,m)$。由稀疏性，这里假设$B_{i1}$是0对应的箱子。对于样本$s$而言，如果其对应的特征都为0时，则投放至$\tilde{B}_1$号，若第$i$个特征非0，且其原特征对应的所在箱子为$B_{ij}$，则投放至$\tilde{B}_k$号，其中

$$k=j+\sum_{p=1}^{i-1}k_p$$

对于上述的互斥特征绑定算法而言，我们确实能够对原数据集的特征进行互斥划分，也提取得到了新的直方图分割点，但考虑如下的情况：特征一和特征二是一族互斥特征，当遍历分割点位于特征一对应的非零区域时，此时右侧的点位对应所有的样本被划入右子节点，可此时划入右子节点的特征二非零值，由于互斥特性，本质上其特征一的值还是零，那么这种划分方法与不进行特征绑定单独考虑特征一相同位置的分割点，它们所计算出的信息增益值由于样本划分不同而会产生差异，这与论文中所描述的互斥特征绑定算法能够无损地提高性能不一致。如果有读者清楚其中缘由，欢迎对教程本段内容作出改进或补充说明，在此感谢。

知识回顾

1. GBDT和梯度下降方法有什么联系？
2. 请叙述GBDT用于分类问题的算法流程。
3. XGBoost和GBDT树有何异同？（可从目标损失、近似方法、分裂依据等方面考虑）
4. 请叙述LightGBM中GOSS和EFB的作用及算法流程。
