Hessian矩阵在XGBoost算法的应用小结

原创 石头 机器学习算法那些事 2019-01-21

前言

Hessian矩阵最常见的应用是牛顿法最优化算法，其主要思想是搜寻一阶导数为0的函数极值点，本文深入浅出的总结了Hessian矩阵在XGboost算法中的两种应用，即权重分位点算法和样本权重和算法 。

目录

1.Hessian矩阵的定义

Hessian矩阵(Hessian matrix或Hessian)是一个自变量为向量的实值函数的二阶导偏导数组成的方块矩阵，此实值函数如下：

$$f(x_1, x_2, ...x_n)$$

自变量为向量：

$$\vec{x} = (x_1, x_2, ...x_n)$$

因此，Hessian矩阵H(f)，定义为：

$$\begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix}$$

2. 最小叶子节点样本权重和算法

官方文档对XGBoost算法参数最小叶子节点样本权重和 (min_child_weitht) 的定义：

> minimum sum of instance weight (**hessian**) needed in a child. If the tree partition step results in a leaf node with the sum of instance weight less than min_child_weight, then the building process will give up further partitioning. In linear regression mode, this simply corresponds to **minimum number of instances** needed to be in each node. The larger, the more conservative the algorithm will be.

翻译：min_child_weight定义为最小叶子节点样本权重(hessian)和，如果树分裂步骤产生的叶子节点上所有样本权重和小于min_child_weight，就停止分裂 。在线性回归的模型中，最小样本个数反映了最小叶子节点样本权重和 。min_child_weight越大，模型越保守，即降低了模型的复杂度 ，避免过拟合 。

下面讨论树模型和线性模型用hessian表示节点样本权重的含义 。

还记得hessian矩阵的定义吗？hessian是函数值对变量的二阶导，xgboost算法中用损失函数对预测值的二阶导表示hessian，如下：

$$\frac{\partial^2 L(y,\hat{y})}{\partial(\hat{y})^2}$$

其中，L 表示损失函数，y和$\hat{y}$分别表示真实值和预测值 。

2.1 线性回归模型

若某一节点的样本数为n，其损失函数：

$$L(y,\hat{y})=\frac{1}{2}\sum_{i=1}^{n}(y_i-\hat{y}_i)^2 \qquad (2.1)$$

对(2.1)式求二阶导：

$$\frac{\partial^2 L(y,\hat{y})}{\partial(\hat{y})^2}=\frac{1}{2}\frac{\partial^2(\sum_{i=1}^{n}(y_i-\hat{y}_i)^2)}{\partial(\hat{y})^2}$$

$$=\sum_{i=1}^{n}1=n$$

因此，线性回归模型中，节点的样本数表示了样本权重和 。

2.2 树模型

树模型中用Hessian表示样本权重，决定某一节点是否切分的条件是比较该节点的样本权重和与min_child_weight的大小，若大于，则切分；反之则不切分（假设其他条件满足切分）。

为了加深理解，本节用二分类逻辑回归举例：

二分类逻辑回归的损失函数：

$$L(y,\hat{y}) = -\frac{1}{n}\sum_{i=1}^{n}(y_i * \log(\sigma(\hat{y}_i)) + (1-y_i) * \log(1-\sigma(\hat{y}_i))) \qquad (2.2)$$

其中：

$$\sigma(\hat{y}_i) = \frac{1}{1+e^{-\hat{y}_i}} \qquad (2.3)$$

$$\frac{\partial(\sigma(\hat{y}_i))}{\partial(\hat{y}_i)} = \sigma(\hat{y}_i) * (1-\sigma(\hat{y}_i)) \qquad (2.4)$$

对（2.2）式求一阶导，得：

$$\frac{\partial L(y,\hat{y})}{\partial(\hat{y})} = \frac{\partial[-\frac{1}{n}\sum_{i=1}^{n}(y_i * \log(\sigma(\hat{y}_i)) + (1-y_i) * \log(1-\sigma(\hat{y}_i)))]}{\partial(\hat{y}_i)} \qquad (2.5)$$

$$\Rightarrow -\frac{1}{n}\sum_{i=1}^{n}(y_i * \frac{1}{\sigma(\hat{y}_i)} * \sigma(\hat{y}_i) * (1-\sigma(\hat{y}_i)) + (-1) * (1-y_i) * \frac{1}{1-\sigma(\hat{y}_i)} * \sigma(\hat{y}_i) * (1-\sigma(\hat{y}_i))$$

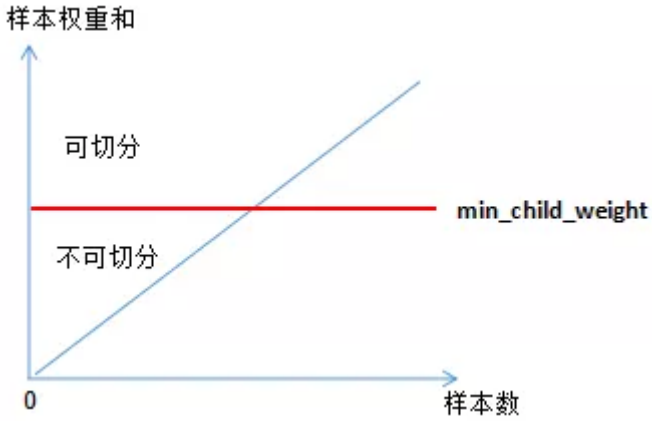
$$\Rightarrow -\frac{1}{n}\sum_{i=1}^{n}(y_i * (1-\sigma(\hat{y}_i)) - (1-y_i) * \sigma(\hat{y}_i)) \qquad (2.6)$$

对(2.6)式再次求一阶导，得：

$$\frac{\partial^2(L(y,\hat{y}))}{\partial((\hat{y}))^2} = -\frac{1}{n}\sum_{i=1}^{n}(-y_i * \sigma(\hat{y}_i)(1-\sigma(\hat{y}_i)) - (1-y_i) * \sigma(\hat{y}_i)(1-\sigma(\hat{y}_i)))$$

$$\Rightarrow \frac{1}{n}\sum_{i=1}^{n}\sigma(\hat{y}_i)(1-\sigma(\hat{y}_i)) \qquad (2.7)$$

(2.7)式表示节点所有样本的权重和，可以定性的画出(2.7)式与min_child_weight的关系：



由上图可知，样本数与样本权重成正相关的关系，因此树模型的样本权重和在一定程度上反映了样本数 。

由xgboost算法的性质可知：

(1)、当yi为1时，$\hat{y}_i$ 越大，损失函数越小，(2.7)式越接近于0 。

(2)、当yi为0时，$\hat{y}_i$ 越小，损失函数越小，(2.7)式越接近于0 。

因此，（2.7）式也可以表示某一节点的纯度(purity)，若纯度大于 min_child_weight，那么可继续对该节点切分；纯度小于 min_child_weight，则不切分 。

3. 权重分位点算法

对损失函数的二阶导值进行排序，然后根据分位点进行切分，选择最佳切分点 。这里不展开了，具体算法请参考XGBoost切分点算法 。

4. 总结

XGBoost算法用损失函数的二阶导（Hessian）表示样本权重，这一思想在线性回归模型中反映了节点的样本个数，在树模型中反映了该节点的纯度 。

参考:

https://stats.stackexchange.com/questions/317073/explanation-of-min-child-weight-in-xgboost-algorithm#

推荐阅读

XGBoost算法原理小结

XGBoost切分点算法

XGBoost参数调优小结