[校招-基础算法]GBDT/XGBoost常见问题


Jack Stark
机器学习，深度学习，智能风控，计算机视觉
关注他

在非深度学习的机器学习模型中，基于GBDT算法的XGBoost、lightgbm等有着非常优秀的性能，校招算法岗面试中“出镜率”非常高。这方面的资料非常多，因此本文不是原创，参考了很多面经、解读文章等，对GBDT相关的问题做了总结。

XGBoost的原理

这个我建议直接看陈天奇的PPT和原论文。由于理论还是比较复杂的，所以需要时不时的回头读一读。

介绍一下XGBoost的原理

XGBoost是基于GBDT的一种算法或者说工程实现。

GBDT是一种基于boosting集成思想的加法模型，训练时采用前向分布算法进行贪婪的学习，每次迭代都学习一棵CART树来拟合之前 t-1 棵树的预测结果与训练样本真实值的残差。

XGBoost的基本思想和GBDT相同，但是做了一些优化，如默认的缺失值处理，加入了二阶导数信息、正则项、列抽样，并且可以并行计算等。

XGBoost和GBDT的不同点：

- GBDT是机器学习算法，XGBoost是该算法的**工程实现**。
- 在使用CART作为基分类器时，XGBoost显式地加入了**正则项**来控制模型的复杂度，有利于防止过拟合，从而提高模型的泛化能力。
- GBDT在模型训练时只是用了代价函数的一阶导数信息，XGBoost对代价函数进行**二阶泰勒展开**，可以同时使用一阶和二阶导数。（好处：相对于GBDT的一阶泰勒展开，XGBoost采用二阶泰勒展开，可以更为精准的逼近真实的损失函数。需要注意的是，损失函数需要二阶可导。）
- 传统的GBDT采用CART作为基分类器，XGBoost支持**多种类型的基分类器**，比如线性分类器。
- 传统的GBDT在每轮迭代时使用全部的数据，XGBoost则采用了与随机森林相似的策略，支持对数据进行**列采样**。
- 传统的GBDT没有涉及对缺失值进行处理，XGBoost能够自动学习出**缺失值的处理策略**。
- **特征维度上的并行化**。XGBoost预先将每个特征按特征值排好序，存储为块结构，分裂结点时可以采用多线程并行查找每个特征的最佳分割点，极大提升训练速度。

GBDT的梯度相关问题

GBDT中的梯度是什么对什么的梯度?

当前损失函数L(yi, F(x))对树F(xi)的梯度。

给一个有m个样本，n维特征的数据集，如果用LR算法，那么梯度是几维?

对权重w有n维，对bias有1维，因此是n+1维。

m*n的数据集，如果用GBDT，那么梯度是几维？m维？n维？m*n维？或者是与树的深度有关？或者与树的叶子节点的个数有关?

目前还不是很确定😭，后面补充。

XGBoost的并行是怎么做的?

由于XGBoost是一种boosting算法，树的训练是串行的，不能并行。这里的并行指的是特征维度的并行。在训练之前，每个特征按特征值对样本进行预排序，并存储为Block结构，在后面查找特征分割点时可以重复使用，而且特征已经被存储为一个个block结构，那么在寻找每个特征的最佳分割点时，可以利用多线程对每个block并行计算。

XGBoost算法防止过拟合的方法有哪些?

- 在目标函数中添加了正则化。叶子节点个数+叶子节点权重的L2正则化。
- 列抽样。训练时只使用一部分的特征。
- 子采样。每轮计算可以不使用全部样本，类似bagging。
- early stopping。如果经过固定的迭代次数后，并没有在验证集上改善性能，停止训练过程。
- shrinkage。调小学习率增加树的数量，为了给后面的训练留出更多的空间。

使用XGBoost训练模型时，如果过拟合了怎么调参?

- 控制模型的复杂度。包括max_depth,min_child_weight,gamma 等参数。
- 增加随机性，从而使得模型在训练时对于噪音不敏感。包括subsample,colsample_bytree。
- 减小learning rate，但需要同时增加estimator 参数。

XGBoost的正则项是什么?

叶子节点个数和**叶子节点权重的L2正则。**

XGBoost为什么对缺失值不敏感?

一些**涉及到对样本距离的度量的模型**，如SVM和KNN，如果缺失值处理不当，最终会导致模型预测效果很差。

而**树模型对缺失值的敏感度低**，大部分时候可以在数据缺失时时使用。原因就是，一棵树中每个结点在分裂时，寻找的是某个特征的最佳分裂点（特征值），完全可以不考虑存在特征值缺失的样本，也就是说，如果某些样本缺失的特征值缺失，对寻找最佳分割点的影响不是很大。

另外，XGBoost还有一些处理缺失值的方法。

XGBoost怎么处理缺失值?

这是XGBoost的一个优点。具体处理方法为：

- 在某列特征上寻找分裂节点时，不会对缺失的样本进行遍历，只会对非缺失样本上的特征值进行遍历，这样减少了为稀疏离散特征寻找分裂节点的时间开销。
- 另外，为了保证完备性，对于含有缺失值的样本，会分别把它分配到左叶子节点和右叶子节点，然后再选择分裂后增益最大的那个方向，作为预测时特征值缺失样本的默认分支方向。
- 如果训练集中没有缺失值，但是测试集中有，那么默认将缺失值划分到右叶子节点方向。

XGBoost中的一棵树的停止生长条件

- 当树达到最大深度时，停止建树，因为树的深度太深容易出现过拟合，这里需要设置一个超参数max_depth。
- 当新引入的一次分裂所带来的增益Gain<0时，放弃当前的分裂。这是训练损失和模型结构复杂度的博弈过程。
- 当引入一次分裂后，重新计算新生成的左、右两个叶子结点的样本权重和。如果任一个叶子结点的样本权重低于某一个阈值，也会放弃此次分裂。这涉及到一个超参数:最小样本权重和，是指如果一个叶子节点包含的样本数量太少也会放弃分裂，防止树分的太细。

XGBoost可以做特征选择，它是如何评价特征重要性的?

XGBoost中有三个参数可以用于评估特征重要性：

- **weight**：该特征在所有树中被用作分割样本的总次数。
- **gain**：该特征在其出现过的所有树中产生的平均增益。
- **cover**：该特征在其出现过的所有树中的平均覆盖范围。覆盖范围这里指的是一个特征用作分割点后，其影响的样本数量，即有多少样本经过该特征分割到两个子节点。

随机森林和GBDT的异同点

相同点：

- 都是由多棵树组成，最终的结果都是由多棵树一起决定。

不同点：

- **集成学习**：RF属于bagging思想，而GBDT是boosting思想。
- **偏差-方差权衡**：RF不断的降低模型的方差，而GBDT不断的降低模型的偏差。
- **训练样本**：RF每次迭代的样本是从全部训练集中有放回抽样形成的，而GBDT每次使用全部样本。
- **并行性**：RF的树可以并行生成，而GBDT只能顺序生成(需要等上一棵树完全生成)。
- **最终结果**：RF最终是多棵树进行多数表决（回归问题是取平均），而GBDT是加权融合。
- **数据敏感性**：RF对异常值不敏感，而GBDT对异常值比较敏感。
- **泛化能力**：RF不易过拟合，而GBDT容易过拟合。

逻辑回归(LR)和GBDT的区别

- LR是线性模型，可解释性强，很容易并行化，但学习能力有限，需要大量的人工特征工程
- GBDT是非线性模型，具有天然的特征组合优势，特征表达能力强，但是树与树之间无法并行训练，而且树模型很容易过拟合；
- 对于高维稀疏数据，GBDT效果不如LR。

XGBoost中如何对树进行剪枝

- 在目标函数中增加了正则项：使用叶子结点的数目和叶子

杂度。
- 在结点分裂时，定义了一个阈值，如果分裂后目标函数的增益小于该阈值，则不分裂。
- 当引入一次分裂后，重新计算新生成的左、右两个叶子结点的样本权重和。如果任一个叶子结点的样本权重低于某一个阈值（最小样本权重和），也会放弃此次分裂。
- XGBoost 先从顶到底建立树直到最大深度，再从底到顶反向检查是否有不满足分裂条件的结点，进行剪枝。

XGBoost如何选择最佳分裂点?

XGBoost在训练前预先将特征按照特征值进行了排序，并存储为block结构，以后在结点分裂时可以重复使用该结构。

因此，可以采用特征并行的方法利用多个线程分别计算每个特征的最佳分割点，**根据每次分裂后产生的增益，最终选择增益最大的那个特征的特征值作为最佳分裂点。**如果在计算每个特征的最佳分割点时，对每个样本都进行遍历，计算复杂度会很大，这种全局扫描的方法并不适用大数据的场景。XGBoost还提供了一种直方图近似算法，对特征排序后仅选择常数个候选分裂位置作为候选分裂点，极大提升了结点分裂时的计算效率。

XGBoost的延展性很好，怎么理解?

可以有三个方面的延展性：

- **基分类器**：弱分类器可以支持CART决策树，也可以支持LR和Linear。
- **目标函数**：支持自定义loss function，只需要其二阶可导。因为需要用二阶泰勒展开，得到通用的目标函数形式。
- **学习方法**：Block结构支持并行化，支持 Out-of-core计算。

参考资料：

陈天奇的PPT和原论文

面试官如何判断面试者的机器学习水平?
www.zhihu.com

珍藏版 | 20道XGBoost面试题，你会几个？(上篇)

珍藏版 | 20道XGBoost面试题，你会几个？(下篇)

编辑于 2019-09-18

机器学习　集成学习　秋招

文章被以下专栏收录

机器学习小王子
机器学习、深度学习、数据结构和算法的总结笔记

关注专栏

推荐阅读

[校招-基础算法]softmax反向传播

深度学习中，softmax是个基础的函数或者层，很简单，因此适合考察候选人的基础理论知识。那么softmax反向传播应该怎么推导呢?

Jack ... 发表于机器学习小...

[校招-基础算法]机器学习问题

为什么树模型的特征不需要归一化？使用归一化的目的是因为各个特征之间数值大小不同使得它们对某些模型（通过梯度下降法求解的模型通常是需要归一化的），比如神经网络的输出和损失计算的影...

Jack ... 发表于机器学习小...

本文约1
本文作者
3种处理
包括数据
方法和算
方法。 分

清华大学

11 条评论

切换为时间排序

写下你的评论...

三除米甫 5 个月前

梯度是m维还是n维的那个问题, 答案应该是m维(sample size)

2

点儿点儿 回复 三除米甫 4 个月前

求大佬详解[思考]

赞

yyqlx 回复 点儿点儿 3 个月前

因为GBDT求导是关于所有生成树的和，而所有生成树的和是一维的数值，又总共有m个sample，所以梯度是m维的。

1

展开其他 1 条回复

Strickland 5 个月前

xgboost降低方差的那部分，增加estimator不是会增加过拟合吗？estimator就是建的树吧

赞

Jack Stark (作者) 回复 Strickland 5 个月前

那句话的关键是减少学习率，让每棵树学的少一点。

赞

Strickland 回复 Jack Stark (作者) 5 个月前

那学习率相等的情况，增加estimator会增加过拟合的风险吧

赞

展开其他 1 条回复

多姆杨 4 个月前

GBDT梯度的维数，难道不应该是f(xi)的个数吗？有几个xi就有几个维吧我的理解是，所以梯度的维度应该是m.....个人理解，不知道对不对哈

赞

yyqlx 回复 多姆杨 3 个月前

是这样的

赞

天地景逸 2 个月前

你好，想问下，从GBDT最中分类器的公式来看，是M个决策树相加的结果，但是为什么每一个决策树的叶子结点数都是J，能帮忙讲解一下吗

赞