

GBDT源码分析之一：总览



cathyxlyl (/u/103933f0bbf0) [+ 关注](#)

2017.04.24 23:20* 字数 2013 阅读 2268 评论 0 喜欢 15

(/u/103933f0bbf0)

0x00 前言

这个系列将会对python的scikit-learn算法包中GBDT算法的源码实现做一个深入梳理和解读。本文会首先对GBDT算法做一个简单的介绍，并对其源码的结构做一个整体上的梳理。因为这里偏重的是源码分析，所以如果想对GBDT算法本身的原理进行深入了解，可以阅读参考文献中推荐的几位大牛的文章。

文章结构

本文将分为下面几个部分：

1. 简要介绍一下GBDT算法的基本概念。
2. scikit-learn中GBDT算法的运行例子。
3. 对GBDT源码结构的一个整体梳理。这里我们会通过思维导图的方式展现GBDT算法实现涉及的主要源码构成。

0x01 GBDT简介

GBDT(Gradient Boosting Decision Tree) 又称 MART(Multiple Additive Regression Tree)或GBRT(Gradient Boosting Regression Tree)，是一种基于回归决策树的Boosting集成算法。

GBDT的核心从算法命名来看一目了然，即决策树（DT）和梯度提升（GB）。

决策树

决策树是一种十分常用和基础的监督学习算法，可适用于分类和回归问题；它将决策过程表述为树状结构，树中的不同路径代表不同的决策分支。决策树的构建过程由根节点出发，根据样本的属性（特征）不断将样本集分裂生成子节点，直至满足停止条件；树



结构的每个叶子节点都代表一个最终的预测结果，一般取落入该叶子节点的样本的众数/概率分布/平均值等。由于通过决策树算法生成的模型可以由一系列if-then规则表述，因此非常易于理解和实现，也是最简单的非线性算法之一。

决策树的关键技术包括分裂点的选择、分裂停止的条件以及避免过拟合的方法（如剪枝；合适的分裂停止条件也可以防止过拟合）。经典的决策树算法包括ID3、C4.5、CART等。

回归树

回归树即用来解决回归问题的决策树。在分类树中，样本标签是离散的或非有序的，我们取叶子节点样本标签的众数或概率分布作为预测结果；而在回归树中，样本标签一般是连续性的有序数据，我们取叶子节点中所有样本标签的平均值作为预测结果。

集成方法（Ensemble Method）

集成学习方法是将多个弱模型通过一定的组合方式组成一个新的强模型的方法，一般情况下集成的模型具有更强的预测和泛化能力。在机器学习问题中，这是一种非常强大的思路，也是“集体智慧”的典型例子。集成算法中的弱模型又称元算法；在GBDT中，回归树是GBDT的元算法。

我们在理解集成方法时，可以更多将其看作一些学习框架，重点在于理解这些框架的思路。各种集成算法（如GBDT、随机森林）的核心也可理解为将基本算法（如决策树）带入集成框架（如Boosting、Bagging）的产物。

Boosting与Gradient Boosting

Boosting的意思是“提升”，它关注被预测错误的样本，基于预测错误的部分构建新的弱模型并集成，是一种常用的迭代集成方法。原始的Boosting方法可以说是基于“样本”的，它会在一开始给所有样本附上相等的权重值，在每轮迭代（生成一个弱模型）后增加预测错误的样本的权重，减少预测正确的样本的权重，并在此基础上训练新的弱模型；最终通过加权或投票的形式对所有弱模型进行组合，生成强模型。

而Gradient Boosting和原始Boosting方法不同的地方在于，它在残差减少的梯度方向建立新的弱模型。直观上看，它用来训练第K轮弱模型的数据，来自于之前所有弱模型集成后的预测值和样本真实值的“差”（准确来说损失函数梯度减少的方向）。

基于上面描述的一系列概念，我们可以较为容易的理解：一个GBDT模型由多颗回归决策树组成；理论上在训练过程中的一轮迭代中，算法基于残差减少的梯度方向生成一颗决策树（scikit-learn在用GBDT解决多标签问题时，实际上在每一轮迭代中用了多棵回归树，本文中我们不对这种情况做深入说明）。在预测阶段，累加模型中所有决策树的预



测值(乘上步长/学习率)，即可计算整个模型的预测结果。

GBDT算法在实际生产中运用非常广泛，表达能力也很强，通常不需要复杂的特征工程就能得到较好的预测效果，还能输出特征重要性得分；同时通过设定合理的样本和特征抽样比例，可以在训练过程中实现交叉检验（cross validation），有效地减少模型过拟合的出现。缺点则是基于Boosting集成方法的算法较难实现并行化，且基于GBDT的模型会较为复杂，深入分析和调优会有一定困难性。

0x02 运行示例

scikit-learn中ensemble包下关于GBDT的算法有两个，分别用来解决回归问题 `GradientBoostingRegressor` 和分类问题 `GradientBoostingClassifier`，调用起来十分简单。

回归示例（波士顿房价数据集）



```
from sklearn.ensemble import GradientBoostingRegressor
from sklearn.datasets import load_boston
from sklearn.metrics import mean_squared_error
from sklearn.model_selection import train_test_split

# 导入数据
X_train, X_test, y_train, y_test = train_test_split(load_boston().data, load_boston().target,
                                                    test_size=0.3, random_state=0)

"""初始化算法, 设置参数

一些主要参数
loss: 损失函数, GBDT回归器可选'ls', 'lad', 'huber', 'quantile'。
learning_rate: 学习率/步长。
n_estimators: 迭代次数, 和learning_rate存在trade-off关系。
criterion: 衡量分裂质量的公式, 一般默认即可。
subsample: 样本采样比例。
max_features: 最大特征数或比例。

决策树相关参数包括max_depth, min_samples_split, min_samples_leaf, min_weight_fraction_leaf。

verbose: 日志level。
具体说明和其它参数请参考官网API。
"""
reg_model = GradientBoostingRegressor(
    loss='ls',
    learning_rate=0.02,
    n_estimators=200,
    subsample=0.8,
    max_features=0.8,
    max_depth=3,
    verbose=2
)

# 训练模型
reg_model.fit(X_train, y_train)

# 评估模型
prediction_train = reg_model.predict(X_train)
rmse_train = mean_squared_error(y_train, prediction_train)
prediction_test = reg_model.predict(X_test)
rmse_test = mean_squared_error(y_test, prediction_test)
print "RMSE for training dataset is %f, for testing dataset is %f." % (rmse_train, rmse_test)
"""Output:
RMSE for training dataset is 4.239157, for testing dataset is 10.749044.
"""
```

分类示例（鸢尾花分类数据集）



```
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.datasets import load_iris
from sklearn.metrics import confusion_matrix
from sklearn.model_selection import train_test_split

# 导入数据
X_train, X_test, y_train, y_test = train_test_split(load_iris().data, load_iris().target)

"""初始化算法, 设置参数

一些主要参数
loss: 损失函数, GBDT分类器可选'deviance', 'exponential'.
learning_rate: 学习率/步长。
n_estimators: 迭代次数, 和learning_rate存在trade-off关系。
criterion: 衡量分裂质量的公式, 一般默认即可。
subsample: 样本采样比例。
max_features: 最大特征数或比例。

决策树相关参数包括max_depth, min_samples_split, min_samples_leaf, min_weight_fraction_leaf。

verbose: 日志level。
具体说明和其它参数请参考官网API。
"""
clf_model = GradientBoostingClassifier(
    loss='deviance',
    learning_rate=0.01,
    n_estimators=50,
    subsample=0.8,
    max_features=1,
    max_depth=3,
    verbose=2
)

# 训练模型
clf_model.fit(X_train, y_train)

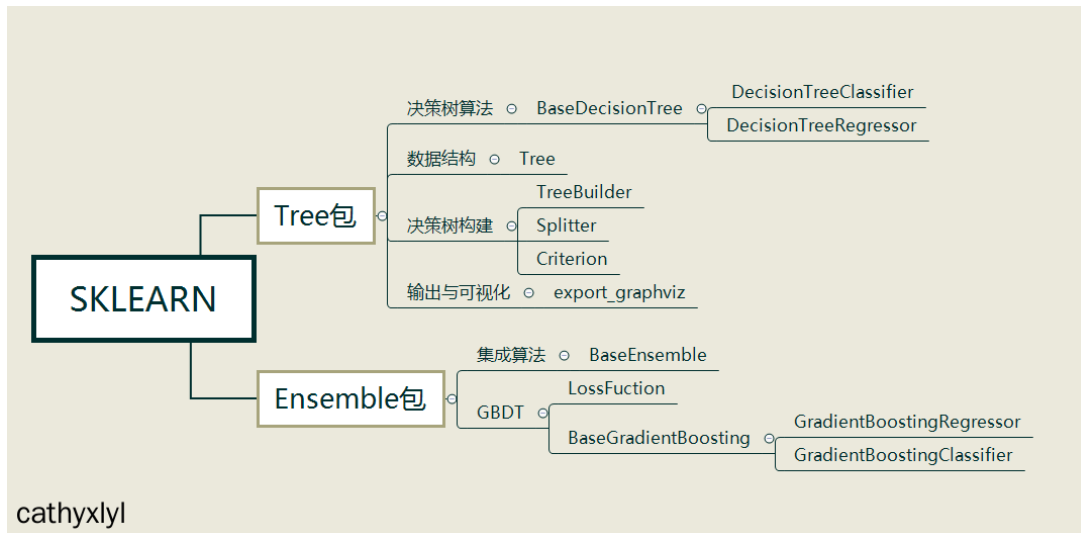
# 评估模型
prediction_train = clf_model.predict(X_train)
cm_train = confusion_matrix(y_train, prediction_train)
prediction_test = clf_model.predict(X_test)
cm_test = confusion_matrix(y_test, prediction_test)
print "Confusion matrix for training dataset is \n%s\n for testing dataset is \n%s." % (cm_train, cm_test)
"""Output:
Confusion matrix for training dataset is
[[40  0  0]
 [ 0 40  1]
 [ 0  1 38]]
 for testing dataset is
[[10  0  0]
 [ 0  8  1]
 [ 0  0 11]].
"""
```

0x03 源码总览



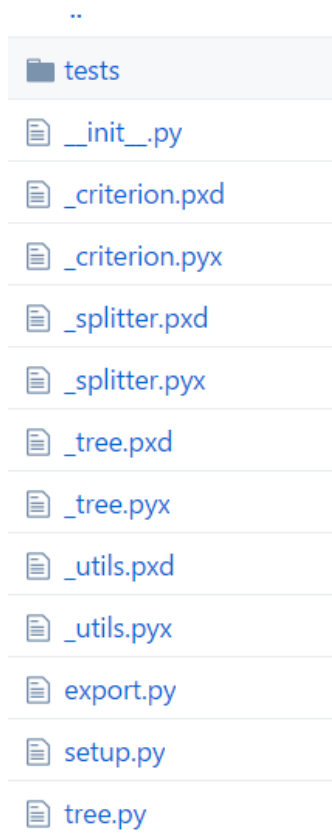
整体介绍

Python的scikit-learn包包含了我们常用的大部分的机器学习算法和数据处理方法，我们主要分析其中实现GBDT的源码。GBDT的实现源码依然可以被分为GB和DT两部分。其中DT为决策树部分，其源码在一个名为Tree的package下；GB为gradient boosting方法，其相关源码在一个名为Ensemble的package下。总体结构见下面的思维导图。



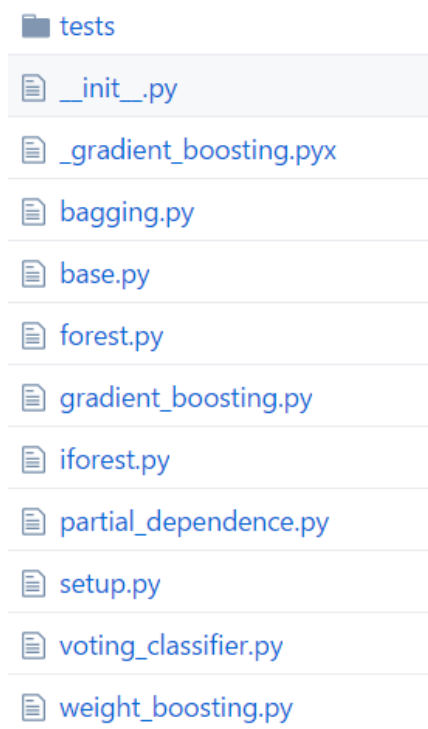
Tree包的源码结构截图如下。里面实现了决策树算法、决策树的基本数据结构Tree、决策树构建策略以及树的可视化等内容。





Ensemble包的源码结构截图如下。Ensemble包里还包含了如bagging、随机森林等其它主题，但我们主要关注其中的base.py和gradient_boosting.py文件。





在本系列后续的两篇文章里，我们将分别介绍Tree包和Ensemble包中和GBDT相关的内容。

0xFF 参考：

- GBDT源码：<https://github.com/scikit-learn/scikit-learn/tree/master/sklearn> (<https://github.com/scikit-learn/scikit-learn/tree/master/sklearn>)
- scikit-learn官方文档：<http://scikit-learn.org/stable/modules/ensemble.html#gradient-tree-boosting> (<http://scikit-learn.org/stable/modules/ensemble.html#gradient-tree-boosting>)
- XGBoost 与 Boosted Tree：<http://www.52cs.org/?p=429> (<http://www.52cs.org/?p=429>)（虽然是陈天奇介绍XGBoost的文章，但对Boosted Tree的概念讲的很清晰）
- 决策树模型组合之随机森林与GBDT：<http://www.cnblogs.com/LeftNotEasy/archive/2011/03/07/random-forest-and-gbdt.html> (<http://www.cnblogs.com/LeftNotEasy/archive/2011/03/07/random-forest-and-gbdt.html>)
- scikit-learn 梯度提升树(GBDT)调参小结：<http://www.cnblogs.com/pinard/p/6143927.html> (<http://www.cnblogs.com/pinard/p/6143927.html>)



作者：**cathyxlyl** (<http://cathyxlyl.github.io/>) | 简书 (<http://www.jianshu.com/users/103933f0bbf0/>) | GITHUB (<https://github.com/cathyxlyl>)

个人主页：<http://cathyxlyl.github.io/> (<http://cathyxlyl.github.io/>)
文章可以转载, 但必须以超链接形式标明文章原始出处和作者信息

 Machine Learning (/nb/12036111) 举报文章 © 著作权归作者所有



cathyxlyl (/u/103933f0bbf0)

写了 9119 字，被 40 人关注，获得了 44 个喜欢
(/u/103933f0bbf0)

+ 关注

 喜欢 (/sign_in?utm_source=desktop&utm_medium=not-signed-in-like-button)

1









更多分享

(<http://cwb.assets.jianshu.io>)



下载简书 App ▶

随时随地发现和创作内容



53.jpg)

(/apps/download?utm_source=nbc)

被以下专题收入，发现更多相似内容





数据科学家 (/c/30f808cd178d?utm_source=desktop&utm_medium=notes-included-collection)



机器学习与数据挖掘 (/c/9ca077f0fae8?utm_source=desktop&utm_medium=notes-included-collection)



Machine... (/c/4b3f92364497?utm_source=desktop&utm_medium=notes-included-collection)



程序员 (/c/NEt52a?utm_source=desktop&utm_medium=notes-included-collection)



机器学习 (/c/9744fc0319ad?utm_source=desktop&utm_medium=notes-included-collection)



玩转大数据 (/c/872c6d687a5f?utm_source=desktop&utm_medium=notes-included-collection)



数据乐园 (/c/a3017f6e996e?utm_source=desktop&utm_medium=notes-included-collection)

展开更多 ▾

机器学习算法小结与收割offer遇到的问题 (/p/ace5051d0023?utm_campaign=...

机器学习是做NLP和计算机视觉这类应用算法的基础，虽然现在深度学习模型大行其道，但是懂一些传统算法的原理和它们之间的区别还是很有必要的。可以帮助我们做一些模型选择。本篇博文就总结一下各种机器学习算法的特点和应用场景。本文是笔者结合自身面试中遇到的问题和总结网络上的资源得到的...

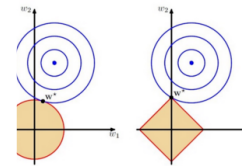


在河之简 (/u/5ff1acaa6334?utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation)

(/p/0d70bf2510b7?utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommendation)

面试 (/p/0d70bf2510b7?utm_campaign=maleskine&...

ML & DM 集成学习 模型融合 ensemble <http://wakemeup.space/?p=109> EM
EM算法的目标是找出有隐性变量的概率模型的最大可能性解，它分为两个过程
E-step和M-step，E-step通过最初假设或上一步得出的模型参数得到后验概率...



章鱼哥呀 (/u/19777d5480c1?utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation)



GBDT源码分析之三：GBDT (/p/1fa837221360?utm_campaign=maleskin...

0x00 前言 本文是《GBDT源码分析》系列的第三篇，主要关注和GBDT本身以及Ensemble算法在scikit-learn中的实现。0x01 整体说明 scikit-learn的ensemble模块里包含许多各式各样的集成模型，所有源码均在sklearn/ense...



cathyxlyl (/u/103933f0bbf0?utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation)

机器学习(Machine Learning)&深度学习(Deep Learning)资料(Chapter 1) (...)

机器学习(Machine Learning)&深度学习(Deep Learning)资料(Chapter 1) 注:机器学习资料篇目一共500条,篇目二开始更新 希望转载的朋友,你可以不用联系我,但是一定要保留原文链接,因为这个项目还在继续也在不定期更新. 希望看到文章的朋友...

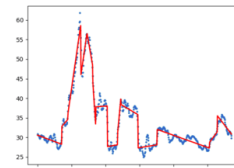


Albert陈凯 (/u/185a3c553fc6?utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation)

(/p/3c8cca3e1ca2?utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommendation)

机器学习_集成算法 (/p/3c8cca3e1ca2?utm_campaign=...

为什么使用集成算法 简单算法一般复杂度低,速度快,易展示结果,但预测效果往往不是特别好。每种算法好像一种专家,集成就是把简单的算法(后文称基算法/基模型)组织起来,即多个专家共同决定结果。如何组织算法和数据 这里我们的着眼点不是某个算法,某个函数,而是对数据和算法整体...



xieyan0811 (/u/a1e43e17d34d?utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation)

歌词|《爱太浓》(词作者:半岛雪)(/p/544b9fa59853?utm_campaign=...

《爱太浓》 爱太浓心太重 飞蛾飞不过眼瞳的牢笼 爱亦死心还在 月光跃然纸上一抹灰白 爱太浓雾太重 爱消失太快看不清雾霭 谁的存在一句活该 爱情转了个弯离开 爱太浓泪太重 心上了枷锁模糊的禁界 扼住了咽喉想嘶喊 暖冬转寒冰封了灰色地带 想说爱琴弦绷紧爱的惩戒 想要逃带着伤灯火...



半岛雪 (/u/aac15940bf44?utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation)

(/p/056e5c77f338?utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommendation)

听了一个月都舍不得删掉的一首歌!听一次哭一次!(/...

不要轻易放弃感情,谁都会心疼;不要冲动下做决定,会后悔一生。也许只一句分手,就再也不见;也许只一次主动,就能挽回遗憾。世界上没有不争吵的感情,只有不肯包容的心灵;生活中没有不会生气的人,只有不知原谅的心。痛经 少经 闭经 子宫肌瘤 卵巢囊肿 等妇科问题?!加微信:jkc55...





生活那点事 (/u/698f044efac9?utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation)

How To Read A Book 翻译之四 (/p/36e171d8c4ea?utm_campaign=male...

The levels of reading 这是练习翻译How to read a book 这本书的系列文章之四。因为逐句翻译过于繁杂，改为通篇概括意译，然后突出重点句子。在之前的章节里，我们区分了几个概念。我们把阅读的目标分为，娱乐性的、获取知识性的和提升认知型的。...



西北的星空 (/u/047d0d7d13ee?utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation)

(/p/09625b6e5662?utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommendation)

整容？美丽的背后可能是惊悚！ (/p/09625b6e5662?ut...

文/逸轩风水 随着社会大众对颜值的一再不良追求，整容逐渐从不被接受的行为，慢慢的让人开始习以为常。大街上，随处可见，微整形，整容。然而，天道循环，美丽的背后，潜藏着不为人知的惊悚！在面相上来讲，在脸上动刀子那是极为不明智的，每个人的脸不仅仅是为了好看，也有一些福气藏在...



逸轩风水 (/u/d90c9a60d7ef?utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation)

