

CSDN新首页上线啦，邀请你来立即体验！(http://blog.csdn.net/)

立即体验

CSDN

博客 (http://blog.csdn.net/?ref=toolbar) 学院 (http://edu.csdn.net/?ref=toolbar)

下载 (http://download.csdn.net/?ref=toolbar) 更多 ▼

🔍

🔗

📄

登录 (https://passport.csdn.net/account/login?ref=toolbar)

注册 (http://passport.csdn.net/account/mobileregister?ref=toolbar&action=mobileRegister)

activity?utm_source=csdnblog1

posted here

XGBoost源码阅读笔记(1)--代码逻辑结构

原创 2017年07月22日 20:46:51

标签：机器学习 (http://so.csdn.net/so/search/s.do?q=机器学习&t=blog) /

xgboost (http://so.csdn.net/so/search/s.do?q=xgboost&t=blog) /

源码 (http://so.csdn.net/so/search/s.do?q=源码&t=blog)

📖 783

👍

🔖

💬

🔗

flydreamforever (http://bl...)

+ 关注

(http://blog.csdn.net/flydreamforever)

原创 粉丝 喜欢 未开通

16 1 1 (https://gite

他的最新文章

更多文章 (http://blog.csdn.net/flydreamforever)

XGBoost源码阅读笔记(2)--树构造之Exact Greedy Algorithm (http://blog.csdn.net/flydreamforever/article/details/76219727)

XGBoost源码阅读笔记(1)--代码逻辑结构 (http://blog.csdn.net/flydreamforever/article/details/75805924)

windows下python安装xgboost (http://blog.csdn.net/flydreamforever/article/details/70767818)

一. XGBoost简介

XGBoost(eXtreme Gradient Boosting)是基于GB (Gradient Boosting) 模型框架实现的一个高效、便捷、可扩展的一个机器学习库。该库先由陈天奇在2014年完成v0.1版本之后开源到github[1]上，当前最新版本是v0.6。目前在各类相关竞赛中都可以看到其出现的身影，如kaggle[2]，在2015年29个竞赛中，top3队伍发表的解决方案中有17个方案使用了XGBoost，而只有11个解决方案使用了深度学习；同时在2015KDDCup中top10队伍都使用了XGBoost[3]。由于其与GBDT (Gradient Boosting decision Tree) 存在一定相似之处，网上也经常会有人将GBDT和XGBoost做个对比[4]。最近正好读了陈天奇的论文《XGBoost: A Scalable Tree Boosting System》[3]，从论文中可以看出XGBoost新颖之处在于：

1. 使用了正则化的目标函数，其加入的惩罚项会控制模型复杂度(叶子个数)和叶子结点的得分权重

$$\mathcal{L}(\phi) = \sum_i l(\hat{y}_i, y_i) + \sum_k \Omega(f_k)$$

where $\Omega(f) = \gamma J + \frac{1}{2} \lambda \|w\|^2$

叶子个数 叶子权重

图1-1 目标函数

2. 使用Shrinkage，通过一个因子η缩减每次最新生成树的权重，其目的是为了降低已生成的树对后续树的影响。
3. 支持列(特征)采样，该方式曾被用于随机森林。可以预防过拟合且加快模型训练速度。
4. 并行计算。Boost方式树是串行生成的，所以其在寻找树分裂点时候进行并行计算，加快模型训练速度。
- 在寻找分裂点时候论文中也提到多种方式：
1. 基本枚举贪婪搜索算法。该方式将特征按其值排序之后，枚举每个特征值作为其分裂点并计算该分裂点的增益，然后选择最大增益的分裂点
2. 近似贪婪搜索算法。该方式在寻找分裂点前会将所有的特征按其对应值进行排序后选择其百分位的点作为候选集合，在执行基本枚举贪婪搜索法。
3. 加权分位数法(weighted quantile sketch)。该方法可以用于对加权数据的处理。
4. 稀疏分裂点查找。可以加快模型对稀疏数据处理。

其与GBDT不同之一在于其对目标函数进行二阶泰勒展开，使用了二阶导数加快模型收敛速度。总的来说XGBoost受到欢迎最重要的一个因素在于其快速的训练过程。

二. 源码下载及编译

Linux上的源码下载和编译过程如下[5]:

```
[cpp]
1. git clone --recursive https://github.com/dmlc/xgboost
2. cd xgboost
3. make

使用--recursive命令是因为XGBoost使用了作者自己编写的分布式计算库，通过这个命令可以下载对应的库，编译好之后就可以开始阅读源码了，XGBoost主要代码目录结构如下：
```

```
[cpp]
1. |--xgboost
2. |--include
3. |--xgboost      //定义了xgboost相关的头文件
4. |--src
5. |--c_api
6. |--common      //一些通用文件，如对配置文件的处理
7. |--data        //使用的数据结构，如DMatrix
8. |--gbm         //定义了若分类器，如gbtree和gblinear
9. |--metric      //定义评价函数
```

```
10. |--objective    //定义目标函数
11. |--tree        //对树的一些列操作
```

36302

👤 1449

windows下python安装xgboost (<http://blog.csdn.net/flydreamforever/article/details/70767818>)

👤 1091

stm32简介 (<http://blog.csdn.net/flydreamforever/article/details/9622577>)

👤 1069

三. 源码逻辑结构

程序的执行入口在cli_main.cc文件中

```
[cpp]
1. //cli_main.cc
2. |--main()
3. |--CLIRunTask()
4. |--CLIParam::configure()
5. |--switch(param.task)
6. {
7.     case kTrain: CLITrain(param);break;
8.     case KDumpModel: CLIDumpModel(param);break;
9.     case KPredict: CLIPredict(param);break;
10. }
```

在main函数中只调用了CLIRunTask()函数，在该函数中可以看出，程序通过函数configure()解析配置文件后，根据参数task选择对应的执行函数。我们这里主要看下训练函数CLITrain();

```
[cpp]
1. //cli_main.cc
2. |--CLITrain()
3. |--DMatrix::Load()
4. |--Learner::Create()
5. |--Learner::Configure()
6. |--Learner::InitModel()
7. |--for (int iter = 0; iter < max_iter; ++iter)
8. {
9.     Learner::UpdateOneIter();
10.    Learner::EvalOneIter();
11. }
```

在CLI函数中，先是将训练数据加载到内存中，然后开始创建Learner类实例，接着调用Learner的configure函数配置参数，调用InitModel()初始化模型。然后就开始XGboost的Boosting训练，主要调用的是Learner的UpdateOneliter()函数。

```
[cpp]
1. //learner.cc
2. |--UpdateOneIter()
3. |--learner::LazyInitDMatrix()
4. |--learner::PredictRaw()
5. |--ObjFunction::GetGradient()
6. |--GradientBooster::DoBoost()
```

在每次迭代过程中，LazyInitDMatrix()先初始化需要用到的数据结构。GetGradient()获取目标函数的一阶导和二阶导，最后DoBoost()执行Boost操作生成一棵回归树。Class GradientBoost是一个抽象类，他定义了Gradient Boost的抽象接口。其派生出的两个类Class GBTree和 Class GBLinear 分别对应着配置文件里面的参数“gbtree”和“gblinear”，Class GBTree主要使用的回归树作为其弱分类器，而Class GBLinear使用的是线性回归或逻辑回归作为其弱分类器。Class GBTree用的比较多，其DoBoost()函数执行的操作如下：

```
[cpp]
1. //gbtree.cc
2. |--GBTree::DoBoost()
3. |--GBTree::BoostNewTrees()
4. |--GBTree::InitUpdater()
5. |--TreeUpdater::Update()
```

DoBoost()调用了BoostNewTrees()函数。在BoostNewTrees()中先初始化了TreeUpdater实例，在调用其Update函数生成一棵回归树。TreeUpdater是一个抽象类，根据使用算法不同其派生出许多不同的Updater，这些Updater都在src/tree目录下。

```
[cpp]
1. |--src
2. |--tree
3. |--updater_basemaker-inl.h
4. |--updater_colmaker.cc
5. |--updater_skmaker.cc
6. |--updater_refresh.cc
7. |--updater_prune.cc
8. |--updater_hismaker.cc
9. |--updater_fast_hist.cc
```

文件updater_basemaker-inl.h中定义了一个派生自TreeUpdater的类BaseMaker。Class ColMaker使用的是基本枚举贪婪搜索算法，通过枚举所有的特征来寻找最佳分裂点；Class SkMaker派生自BaseMaker，使用近似的sketch方法寻找最佳分裂点；Class TreeRefresher用于刷新数据集上树的统计信息和叶子值；Class TreePruner是树的剪枝操作；Class HistMaker使用的是直方图法，该方法在论文中并没有提到，所以也不是很清楚。

至此便可以大致了解XGBoost源码的逻辑结构，目前源码只看到这里。等看了各算法的具体实现之后在后续文章中写其具体实现细节。

四. 参考文献

[1]. <https://github.com/dmlc/xgboost> (<https://github.com/dmlc/xgboost>)

[2]. <https://www.kaggle.com> (<https://www.kaggle.com>)

[3]. Tianqi Chen and Carlos Guestrin. XGBoost: A Scalable Tree Boosting System. In 22nd SIGKDD Conference on Knowledge Discovery and Data Mining, 2016 网址: <https://arxiv.org/abs/1603.02754> (<https://arxiv.org/abs/1603.02754>)

[4]. <https://www.zhihu.com/question/41354392> (<https://www.zhihu.com/question/41354392>)

[5]. <http://xgboost.readthedocs.io/en/latest/build.html> (<http://xgboost.readthedocs.io/en/latest/build.html>)



相关文章推荐

baseline日常使用小结 (http://blog.csdn.net/launch_225/article/details/13289987)

一. 涉及到的函数(4点)Summary of DBMS_SPM Subprograms

launch_225 (http://blog.csdn.net/launch_225) 2013年10月28日 11:07 1294

机器学习：机器学习中的损失函数 (http://blog.csdn.net/li_dongxuan/article/details/556...)

机器学习中的损失函数在机器学习中，损失函数是用来衡量预测结果与实际值之间差别大小的指标。一般的损失函数有5五种：一. Gold Standard（标准式，0-1式）主要用于理想sample，这种一般很...

li_dongxuan (http://blog.csdn.net/li_dongxuan) 2017年02月18日 15:28 406

XGBoost源码阅读笔记(2)--树构造之Exact Greedy Algorithm (<http://blog.csdn.net/flydre...>)

在上一篇《XGBoost源码阅读笔记(1)--代码逻辑结构》中向大家介绍了XGBoost源码的逻辑结构，同时也简单介绍了XGBoost的基本情况。本篇将继续向大家介绍XGBoost源码是如何构造一颗回...

flydreamforever (<http://blog.csdn.net/flydreamforever>) 2017年07月27日 20:47 674

MySQL源码阅读笔记之代码结构 (<http://blog.csdn.net/jwxxyk/article/details/8524050>)

代码版本: MySQL5.5.28 源代码目录布局: 下面就一些重要的源代码文件夹作一个简单的说明: BUILD: 编译配置并为所有被支持的平台制作文件, 内含在各个平台、各种编译器下进行编译和链...

jwxxyk (<http://blog.csdn.net/jwxxyk>) 2013年01月21日 15:13 2329

mysql源码阅读笔记 (1) 底层物理页面的数据结构 (<http://blog.csdn.net/huyangyamin/art...>)

Innodb数据页结构

huyangyamin (<http://blog.csdn.net/huyangyamin>) 2015年02月08日 17:00 1279

Java源码集合类TreeMap学习1——数据结构4平衡二叉树创建代码 (<http://blog.csdn.net...>)

平衡二叉排序树上插入一个新的元素递归算法，还是比较复杂的，特别是代码的实现上想要理解还是要动手去一步步去手动执行代码。个人理解这个算法和看示例代码也是费了很大一番功夫，理解程度上还是初级阶段。总之还是...

muyufenghua (<http://blog.csdn.net/muyufenghua>) 2017年05月14日 16:59 268

OpenCv学习笔记(1)---CvTermCriteria---迭代算法终止条件结构体的---OpenCV源码分析 ...

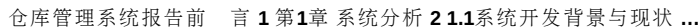
 maweifei (<http://blog.csdn.net/maweifei>) 2016年04月22日 21:28 1887

1.AlbumBrowserActivity此类继承自ListActivity实现接口 View.OnCreateContextMenuListener, MusicUtils.Defs, ...

WDYShowTime (<http://blog.csdn.net/WDYShowTime>) 2017年04月05日 14:53 313

转载自: <http://www.linuxidc.com/Linux/2016-07/133215.htm> TensorFlow0.8发布以来受到了大量机器学习领域爱好者的关注, 目前其...

s_sunnyy (http://blog.csdn.net/s_sunnyy) 2017年03月16日 10:01 295



[/http://download...](#) 2011年02月23日 11:26 577KB [下载 \(](#)

TensorFlow0.8发布以来受到了大量机器学习领域爱好者的关注，目前其项目在github上的follow人数在同类项目中排名第一。作为google的第一个开源项目，TensorFlow的源码结构...

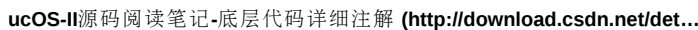
sydpz1987 (<http://blog.csdn.net/sydpz1987>) 2016年05月27日 22:16 4829

Entity种类生物大致划分为四种：攻击型，被动型，水生型（也就是鱿鱼）和环境型（也就是蝙蝠）。攻击型生物有一个每游戏刻（1/20秒）一次的生成周期。被动型和水生型生物只有每400刻（20秒）一次...

qp120291570 (<http://blog.csdn.net/qp120291570>) 2016年05月02日 01:51 8473

HashMap的默认大小是多少呢？HashMap的底层存储结构是什么？HashMap的hash值如何计算？为什么HashMap的容量只能是2的n次幂？本文一一解答...

vicklin (<http://blog.csdn.net/vicklin>) 2015年03月30日 12:33 954



/http://download , 2015年05月09日 10:51 692KB 下载 (

Chrome启动代码流程: (v2.0版, Windows平台) 应用程序启动过程: 1. WinMain函数为入口点, 定义在文件\chrome\app\chrome_exe_main...

zero_lee (http://blog.csdn.net/zero_lee) 2012年08月15日 15:20 3880

beans包的层级结构 bean工程的源码结构如图所示: beans包中的各个源码包的功能如下: src/main/java 用于展示Spring的主要逻辑 src/main/resourc...



nangongyanya (<http://blog.csdn.net/nangongyanya>) 2016年12月23日 15:33 224

jQuery (版本2.0.3) 整体结构如下: 下载地址: <https://code.jquery.com/jquery/> 版权声明: 以下为本人在妙味课堂听课的笔记 (function(window...

Emily_lhj (http://blog.csdn.net/Emily_lhj) 2016年12月05日 15:55 157



nginx源码阅读笔记.array和list数据结构 (<http://blog.csdn.net/axiaojikuaipao/article/deta...>)

1.概述 nginx中,array和list 的实现 和queue的实现不同,queue的实现是不依赖于具体的结构的,可以申明任何结构体只要该结构体中间含有queue的实例就可以将它们组织起来,那么...

 axiaojikuaipao (<http://blog.csdn.net/axiaojikuaipao>) 2017年10月18日 22:13  36



docker源码阅读笔记—github配置及代码提交操作 (http://blog.csdn.net/qq_21898173/ar...)

这篇博文属于Github操作经验及技巧的记录, 通过向github提交docker源码的研究纪录一点github操作的技巧和经验, 这对以后对个人github上代码的管理和提交也很有帮助, 适用于一些git...

 qq_21898173 (http://blog.csdn.net/qq_21898173) 2017年04月01日 15:15  818

IPMsg源码阅读笔记 (1) (<http://blog.csdn.net/yang6696100/article/details/45999987>)

打开下载的源码包, 成功的导入VS2013后是一个名字为IPMsg的解决方案, 这个解决方案里面有6个项目: --install --IPMsg --libpng --TLib --uninst --zl...

 yang6696100 (<http://blog.csdn.net/yang6696100>) 2015年05月26日 00:55  457