

# DarkScope从这里开始

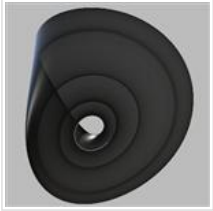
a learner,like Machine Learning.

目录视图

摘要视图

RSS 订阅

## 个人资料



Dark\_Scope



访问: 499094次

积分: 4811

等级: BLOG &gt; 5

排名: 第3440名

原创: 77篇 转载: 2篇

译文: 0篇 评论: 587条

## 个人介绍

DarkScope, 喜欢机器学习和一些ACM算法//学习ing//求交流, 求指教! =新浪微博 我是Darkscope

## 文章搜索

## 博客专栏



机器学习从原理到实践

文章: 5篇

阅读: 92274

## 文章分类

C++ (3)

ACM (7)

杂七杂八 (9)

ASM (5)

QT相关 (2)

机器学习 (39)

大学杂念集 (6)

C/C++ (1)

机器学习读书笔记 (4)

修身养性治情 (3)

web相关 (4)

nlp (6)

代码 (13)

为什么未来是全栈工程师的世界 前端精品课程免费看, 写课评赢心动大礼! JavaScript知识库发布

## GBDT(Gradient Boosting Decision Tree) 没有实现只有原理

标签: machine learning 统计学 算法 GBDT 趋势

2014-05-03 10:23

31113人阅读

评论(20)

收藏

举报

分类: 机器学习 (38)

版权声明: 本文为博主原创文章, 未经博主允许不得转载。

目录(?)

[+]

阿弥陀佛, 好久没写文章, 实在是受不了了, 特来填坑, 最近实习了(ting)解(shuo)到(le)很多工业界常用的算法, 诸如GBDT, CRF, topic model的一些算法等, 也看了不少东西, 有时间可以详细写一下, 而至于实现那真的是没时间没心情再做了, 等回学校了再说吧。今天我们要说的就是GBDT(Gradient Boosting Decision Tree)

## 〇.前序

GBDT是看一个大牛团队做推荐算法比赛的时候拿这个模型来处理最后得到的所有的feature并输出结果的模型, 想到自己以前天真地拿着SVD单模型调参加这类比赛的时候真是.....闻者伤心, 听者流泪啊, 别的不谈, 这次讲GBDT主要是因为了解GBDT的一些前置条件我都在博客里写过, 可以直接跳到关键部分开写.....进入正题吧

## 一.前置条件

### 1.决策树

参看我以前的一篇博客: [http://blog.csdn.net/dark\\_scope/article/details/13168827](http://blog.csdn.net/dark_scope/article/details/13168827)

虽然里面写的都是决策分类树, 而我们这次主讲的是决策回归树, 不过其实都差不多, 决策回归树呢就是把分到某个分支上的所有训练样例的目标值求平均或者取中位数返回而已。

### 2.boosting

一般来说哦讲boosting都以adaboost这个特例开始讲, 所以你可以先看一看我的这篇博客: [AdaBoost--从原理到实现](#)

然后我们来接着讲boosting.....新开一章吧, 这个其实是主要内容

## 二.boosting 提升方法

提升方法其实是一个比adaboost概念更大的算法, 因为adaboost可以表示为boosting的前向分布算法(Forward stagewise additive modeling)的一个特例, boosting最终可以表示为:

$$f(\mathbf{x}) = w_0 + \sum_{m=1}^M w_m \phi_m(\mathbf{x})$$

2015年12月 (1)  
2015年07月 (1)  
2015年03月 (1)  
2015年02月 (1)  
2014年05月 (2)

[展开](#)

AdaBoost--从原理到实现 (56962)  
【面向代码】学习 Deep (44138)  
【面向代码】学习 Deep (41470)  
【面向代码】学习 Deep (31848)  
GBDT(Gradient Boosting) (31112)  
趣写算法系列之--匈牙利 (25327)  
RNN以及LSTM的介绍和 (23420)  
新浪微博小爬虫 (18911)  
从item-base到svd再到rb (17585)  
【面向代码】学习 Deep (17269)

【面向代码】学习 Deep (100)  
【面向代码】学习 Deep (82)  
趣写算法系列之--匈牙利 (69)  
【面向代码】学习 Deep (48)  
新浪微博小爬虫 (44)  
AdaBoost--从原理到实现 (34)  
从item-base到svd再到rb (32)  
UFLDL练习(Sparse Auto (28)  
RNN以及LSTM的介绍和 (21)  
GBDT(Gradient Boosting) (20)

\*Xcode 调试方法总结  
\*Android框架设计模式 (三) ——Observer Method  
\*浅谈Storm流式处理框架  
\*有关深度学习领域的几点想法  
\* 管理Java垃圾回收的五个建议  
\* 大数据并发问题

趣写算法系列之--匈牙利算法 xiaolinzeyi: 牛牛牛  
从item-base到svd再到rbm, 多种suprader: 大赞楼主, 准备学习下  
RNN以及LSTM的介绍和公式梳理 刘伟世: @sinat\_19717219:靠门结构吧, 在T时刻, 可以用遗忘门的开闭程度 (0~1) 来决定保留多少...  
RNN以及LSTM的介绍和公式梳理 刘伟世: 博主在最后关于4.11的解

其中的 $w$ 是权重,  $\Phi$ 是弱分类器(回归器)的集合,其实就是一个加法模型(即基函数的线性组合)

**前向分布算法** 实际上是一个贪心的算法,也就是在每一步求解弱分类器 $\Phi(m)$ 和其参数 $w(m)$ 的时候不去修改之前已经求好的分类器和参数:

### 算法 8.2 (前向分步算法)

输入: 训练数据集  $T = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$ ; 损失函数  $L(y, f(x))$ ; 基函数集  $\{b(x; \gamma)\}$ ;

输出: 加法模型  $f(x)$ .

(1) 初始化  $f_0(x) = 0$

(2) 对  $m = 1, 2, \dots, M$

(a) 极小化损失函数

$$(\beta_m, \gamma_m) = \arg \min_{\beta, \gamma} \sum_{i=1}^N L(y_i, f_{m-1}(x_i) + \beta b(x_i; \gamma)) \quad (8.16)$$

得到参数  $\beta_m, \gamma_m$

(b) 更新

$$f_m(x) = f_{m-1}(x) + \beta_m b(x; \gamma_m) \quad (8.17)$$

(3) 得到加法模型

$$f(x) = f_M(x) = \sum_{m=1}^M \beta_m b(x; \gamma_m) \quad (8.18)$$

(图自《统计学习方法》)

为了表示方便, 我们以后用 $\beta$ 代替 $w$ 进行描述了, 图中的 $b$ 是之前说的 $\Phi$ 弱分类器

OK, 这也就是提升方法 (之前向分布算法) 的大致结构了, 可以看到其中存在变数的部分其实就是极小化损失函数 这关键的一步了, 如何选择损失函数决定了算法的最终效果(名字).....这一步你可以看出算法的“趋势”, 以后再单独把“趋势”拿出来吧, 因为我感觉理解算法的关键之一就是理解算法公式的“趋势”

## 三.各种提升方法

不同的损失函数和极小化损失函数方法决定了boosting的最终效果, 我们现在来说几个常见的boosting:

Name	Loss	Derivative	$f^*$	Algorithm
Squared error	$\frac{1}{2}(y_i - f(\mathbf{x}_i))^2$	$y_i - f(\mathbf{x}_i)$	$E[y \mathbf{x}_i]$	L2Boosting
Absolute error	$ y_i - f(\mathbf{x}_i) $	$\text{sgn}(y_i - f(\mathbf{x}_i))$	$\text{median}(y \mathbf{x}_i)$	Gradient boosting
Exponential loss	$\exp(-\tilde{y}_i f(\mathbf{x}_i))$	$-\tilde{y}_i \exp(-\tilde{y}_i f(\mathbf{x}_i))$	$\frac{1}{2} \log \frac{\pi_i}{1-\pi_i}$	AdaBoost
Logloss	$\log(1 + e^{-\tilde{y}_i f_i})$	$y_i - \pi_i$	$\frac{1}{2} \log \frac{\pi_i}{1-\pi_i}$	LogitBoost

(图自 Machine Learning A Probabilistic Perspective) 对于二分类问题来说: 其中 $\pi_i = \text{sigm}(2f(\mathbf{x}_i))$ ,  $y_i \in \{-1, +1\}$ ,  $y_i \in \{0, 1\}$

广义上来讲, 所谓的Gradient Boosting 其实就是在更新的时候选择梯度下降的方向来保证最后的结果最好, 一些书上讲的“残差”方法其实就是L2Boosting吧, 因为它所定义的残差其实就是L2Boosting的Derivative, 接下来我们着重讲一下弱回归器(不知道叫啥了, 自己编的是决策树的情况, 也就是GBDT. (不知道为何上表的Absolute被命名为了Gradient boosting, 关于Gradient boosting在后面会有更细致的介绍)

## 四.GBDT

对于决策树, 其实可以把它表示为下式, 即是把特征空间划分为多个区域, 每个区域返回某个值作为决策树的预测值

$$T(x; \Theta) = \sum_{j=1}^J \gamma_j I(x \in R_j),$$

其中 $R_j$ 是区域,  $\gamma$ 是返回值,  $I()$ 在其中的条件成立情况下为1, 否则为0.其中的参数 $J$ 可以大概看做树的深度的一个表示, 这是一个待调的参数

我们知道Gradient Boosting最重要的一步就是去拟合下式:

$$g_{im} = \left[ \frac{\partial L(y_i, f(x_i))}{\partial f(x_i)} \right]_{f(x_i)=f_{m-1}(x_i)}$$

对于不同的Loss function, 其梯度有不同的表达式:

释：“这两个东西的最后的Loss function是分开的，彼此之间没有关...

从item-base到svd再到rbm，多和奶茶楚楚：博主图画得很清晰很好看，请问用的什么软件

理解机器学习算法的一点心得  
ericck: 私以为只要你能对算法有一种说得通的解释，就是OK的对于同一种算法，就有许多种解释，感觉每种解释都能说...

OpenCV 的人脸detect及PCA匹配  
gmyPuPuTongTong: 你好，你的这个对我现在在学习的很有用。能麻烦你把整个工程发过来吗？598273892@qq.com,...

UFLDL练习(Sparse Autoencoder)  
markvq: @Dark\_Scope:.....我看了你的源码，原来不用标准化就行了233

UFLDL练习(Sparse Autoencoder)  
Dark\_Scope: @markvq:忘掉了，不过应该是有正确结果的

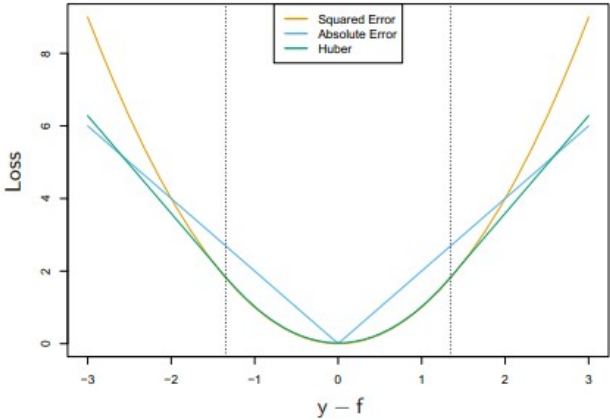
UFLDL练习(Sparse Autoencoder)  
markvq: 不知道你有没有跑后面的MNIST的，这个IMPLEMENTATION好像得不到正确的输出。

TABLE 10.2. Gradients for commonly used loss functions.

Setting	Loss Function	$-\partial L(y_i, f(x_i))/\partial f(x_i)$
Regression	$\frac{1}{2}[y_i - f(x_i)]^2$	$y_i - f(x_i)$
Regression	$ y_i - f(x_i) $	$\text{sign}[y_i - f(x_i)]$
Regression	Huber	$y_i - f(x_i)$ for $ y_i - f(x_i)  \leq \delta_m$ $\delta_m \text{sign}[y_i - f(x_i)]$ for $ y_i - f(x_i)  > \delta_m$ where $\delta_m = \alpha\text{th-quantile}\{ y_i - f(x_i) \}$
Classification	Deviance	$k\text{th component: } I(y_i = \mathcal{G}_k) - p_k(x_i)$

(图自The Elements of Statistic Learning)

前三种对应的loss function如下图：其中Huber是低于某个值表现为square error，高于某个值则表现为线性



下面是GBDT的大概框架：(Gradient Tree Boosting应该是GBDT另一种说法，有误请指正)

Algorithm 10.3 Gradient Tree Boosting Algorithm.

1. Initialize  $f_0(x) = \arg \min_{\gamma} \sum_{i=1}^N L(y_i, \gamma)$ .
2. For  $m = 1$  to  $M$ :
  - (a) For  $i = 1, 2, \dots, N$  compute
$$r_{im} = - \left[ \frac{\partial L(y_i, f(x_i))}{\partial f(x_i)} \right]_{f=f_{m-1}}.$$
  - (b) Fit a regression tree to the targets  $r_{im}$  giving terminal regions  $R_{jm}, j = 1, 2, \dots, J_m$ .
  - (c) For  $j = 1, 2, \dots, J_m$  compute
$$\gamma_{jm} = \arg \min_{\gamma} \sum_{x_i \in R_{jm}} L(y_i, f_{m-1}(x_i) + \gamma).$$
  - (d) Update  $f_m(x) = f_{m-1}(x) + \sum_{j=1}^{J_m} \gamma_{jm} I(x \in R_{jm})$ .
3. Output  $\hat{f}(x) = f_M(x)$ .

(算法自The Elements of Statistical Learning)

整个框架描述得其实已经很清晰了，就不在这里赘述了，总之所谓Gradient就是去拟合Loss function的梯度，将其作为新的弱回归树加入到总的算法中即可。

五.尾巴

本文大概写了一下GBDT的框架和原理，后续其实还有涉及到参数的选择(如树的深度)，正则化(regularization)

等内容，主要是在实现的时候要注意，有时间会写一份toy代码出来。

#### 【Reference】

- 【1】《The Elements of Statistical Learning》
- 【2】《统计学习方法》
- 【3】《Machine Learning A Probabilistic Perspective》

顶 4 踩 2

上一篇 [从item-base到svd再到rbm，多种Collaborative Filtering\(协同过滤算法\)从原理到实现](#)

下一篇 [理解机器学习算法的一点心得](#)

#### 我的同类文章

##### 机器学习（38）

- [RNN以及LSTM的介绍和公...](#) 2015-07-25 阅读 23236
- [从item-base到svd再到rbm...](#) 2013-12-14 阅读 17564
- [SVM --从"原理"到实现](#) 2013-11-23 阅读 8567
- [AdaBoost--从原理到实现](#) 2013-11-03 阅读 56751
- [【面向代码】学习 Deep Le...](#) 2013-08-13 阅读 17234
- [理解机器学习算法的一点心得](#) 2014-05-10 阅读 5011
- [CNN\(Convolutional Neural ...](#) 2013-12-03 阅读 9456
- [KNN\(k-nearest neighbor al...](#) 2013-11-13 阅读 5107
- [决策树--从原理到实现](#) 2013-10-27 阅读 12076

[更多文章](#)

#### 猜你在找

[解码皮肤美化算法](#)

[有趣的算法（数据结构）](#)

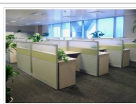
[《C语言/C++学习指南》加密解密篇（安全相关算法）](#)

[Java经典算法讲解](#)

[数据结构和算法](#)



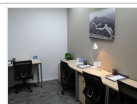
成都七中



创业办公室出



开户送彩金



服务式办公室



html网页



虚拟办公室



通信系统

[查看评论](#)

12楼 [fs302](#) 2015-11-09 19:40发表



博主好牛逼，特来膜拜~

11楼 [ouzhanming](#) 2015-07-27 18:45发表

你可以看看Tianqi Chen的General Functional Matrix Factorization Using Gradient Boosting这篇论文，是ICML 2013的best



paper。而且作者也有项目代码的开源。

Re: [lingerlanlan](#) 2015-07-27 19:36发表



回复ouzhanning: 好的, 谢谢

10楼 [zhouwenyuan1015](#) 2015-07-21 19:17发表



你好, 你用的统计学习方法, 作者是?

9楼 [lingerlanlan](#) 2015-07-15 18:16发表



博主, 请问怎么用gbdt来融合各种推荐算法吗? 网上好难找到这方面的资料。

Re: [ouzhanning](#) 2015-07-27 18:43发表



回复lingerlanlan: 你可以看看Tianqi Chen的General Functional Matrix Factorization Using Gradient Boosting这篇论文, 是ICML 2013的best paper。而且作者也有项目代码的开源。

Re: [Dark\\_Scope](#) 2015-07-15 22:44发表



回复lingerlanlan: 当时做百度电影推荐算法比赛的团队有分享的ppt, 你可以搜索一下, 我记得讲得挺好的

8楼 [zhouxiaojia235](#) 2015-04-09 16:12发表



博主您好, 请教一个问题, 在用gradient boosting做二分类的时候(不是回归), 由于仅仅是二分类, 那么每次迭代只需要一棵树进行拟合, 那么分裂准则应该选择什么?

7楼 [cjweffort](#) 2015-03-30 18:30发表



好文章啊, boosting和gradient boosting框架清晰明了~~尤其是根据损失函数来对boost体系的算法进行概括

6楼 [markvq](#) 2015-01-15 11:07发表



博主今年大四? 打算继续深造还是工作

Re: [Dark\\_Scope](#) 2015-01-15 19:30发表



回复markvq: 准备工作呢

5楼 [诚实的偷包贼](#) 2015-01-05 22:10发表



博主好...我的个人网站是[reetsee.com](#)。

然后, 我是来问问题的, 如果你觉得我不是伸手党的话有劳博主解答一下。

最近我在看机器学习有关的一些资料, 在看到Gradient Boosting Decision Tree的时候有几个疑问:

- 1.对于有很多feature的输入例如X, 输出Y是一个实数值(或离散值也没关系)。那么初始构建的决策树是传统的决策树吗(用C4.5, ID3, CART等方法构造的树)? 如果是的话那么是否需要引入剪枝、信息增益等东西呢? 如果不需要的话, 那么这个“决策树”要怎么构造呢, 就是应该选哪些feature呢? 因为我目前看到的例子都是X只有1个feature的。
- 2.当feature是非常离散的时候, 那么需要对算法做什么特殊的调整吗? 例如我一个输入里面feature可能有成千个, 但是实际不为0的feature可能只有几个, 而各个样本里面所有feature不为0的概率是一样的, 碰到这样的情况应该怎么办呢?

Re: [Dark\\_Scope](#) 2015-01-06 13:03发表



回复诚实的偷包贼: 1.是传统的决策树, 比如实数其实可以让目标值就是那个分支下的样例的均值。真的要实际用的时候肯定剪枝什么的都会尝试的2.这和NLP的情况很类似的, 你可以参考一下相关的资料, 一些处理方法也是可以用的

Re: [诚实的偷包贼](#) 2015-01-07 23:21发表



回复Dark\_Scope: 好多谢博主~!

4楼 [wangsiheaaaa](#) 2014-11-04 16:57发表



你好, 我对adaboost和Forward stagewise additive modeling有一些疑问:

对于adaboost, 每一个弱分类器都是用来最小化weighted errors的, decision stump的数量可以是任意的, 而且类型也可以是任意的。而对于与Forward stagewise additive modeling, 弱分类器的数量和类型在一开始就已经被决定了吗?

3楼 [zyjnju](#) 2014-10-21 19:54发表



这就是传说中的中国好博主吗? 基本上每条评论必回啊, 而且回的好耐心。。。。撿个评论少的第二评论, 哈哈~~

Re: [Dark\\_Scope](#) 2014-11-06 23:19发表



回复zyjnju: 多谢, ^\_^

2楼 [一条好汉](#) 2014-06-17 10:29发表



你好，请问GBDT的树的深度一般设置为多少合适？与特征的个数有关系吗？另，GBDT和RF相比，哪个更强一些？

Re: [Dark\\_Scope](#) 2014-06-17 13:10发表



回复一条好汉：我没有实际使用过，我觉得应该是看具体的实际情况决定哪个比较好

1楼 [ktt8868](#) 2014-05-15 15:05发表



well organized, coll

您还没有登录,请[\[登录\]](#)或[\[注册\]](#)

\* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

核心技术类目

全部主题   [Hadoop](#)   [AWS](#)   [移动游戏](#)   [Java](#)   [Android](#)   [iOS](#)   [Swift](#)   [智能硬件](#)   [Docker](#)   [OpenStack](#)  
[VPN](#)   [Spark](#)   [ERP](#)   [IE10](#)   [Eclipse](#)   [CRM](#)   [JavaScript](#)   [数据库](#)   [Ubuntu](#)   [NFC](#)   [WAP](#)   [jQuery](#)  
[BI](#)   [HTML5](#)   [Spring](#)   [Apache](#)   [.NET](#)   [API](#)   [HTML](#)   [SDK](#)   [IIS](#)   [Fedora](#)   [XML](#)   [LBS](#)   [Unity](#)  
[Splashtop](#)   [UML](#)   [components](#)   [Windows Mobile](#)   [Rails](#)   [QEMU](#)   [KDE](#)   [Cassandra](#)   [CloudStack](#)  
[FTC](#)   [coremail](#)   [OPhone](#)   [CouchBase](#)   [云计算](#)   [iOS6](#)   [Rackspace](#)   [Web App](#)   [SpringSide](#)   [Maemo](#)  
[Compuware](#)   [大数据](#)   [aptech](#)   [Perl](#)   [Tornado](#)   [Ruby](#)   [Hibernate](#)   [ThinkPHP](#)   [HBase](#)   [Pure](#)   [Solr](#)  
[Angular](#)   [Cloud Foundry](#)   [Redis](#)   [Scala](#)   [Django](#)   [Bootstrap](#)

[公司简介](#) | [招贤纳士](#) | [广告服务](#) | [银行汇款帐号](#) | [联系方式](#) | [版权声明](#) | [法律顾问](#) | [问题报告](#) | [合作伙伴](#) | [论坛反馈](#)

[网站客服](#)   [杂志客服](#)   [微博客服](#)   [webmaster@csdn.net](#)   400-600-2320 | 北京创新乐知信息技术有限公司 版权所有 | 江苏乐知网络技术有限公司 提供商务支持  
京 ICP 证 09002463 号 | Copyright © 1999-2014, CSDN.NET, All Rights Reserved 