**ctr预测模型设计**

目录

[1.0 模型简述 3](#_Toc463613656)

[2.0 GBDT 3](#_Toc463613657)

[2.1 简述 3](#_Toc463613658)

[2.2 回归树 4](#_Toc463613659)

[2.3 提升树 4](#_Toc463613660)

[2.4 梯度提升树 5](#_Toc463613661)

[3.0 FFM 6](#_Toc463613662)

[3.1 FM 6](#_Toc463613663)

[3.2 FFM 7](#_Toc463613664)

[4.0 总结 9](#_Toc463613665)

# 1.0 模型简述

本模型由GBDT (Gradient Boosting Decision Tree，决策提升树)模型和ffm（Field-aware Factorization Machine）模型组合。GBDT用于划分特征空间以及生成新的特征，ffm主要用于输出最后预测的ctr。

# 2.0 GBDT

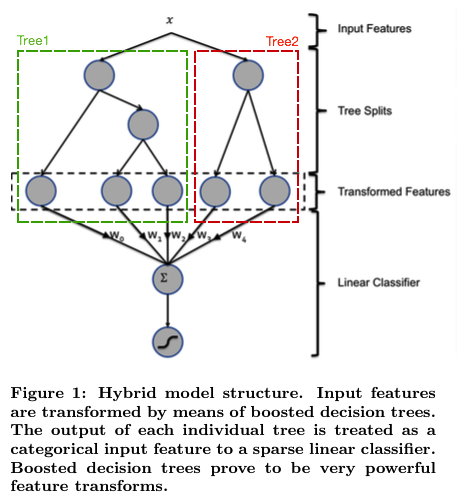
## 2.1 简述

GBDT模型应用于ctr预测首次出现在2014年[facebook的论文](http://www.herbrich.me/papers/adclicksfacebook.pdf)中。

GBDT（Gradient Boost Decision Tree）是一种常用的非线性模型，它基于集成学习中的boosting思想，每次迭代都在减少残差的梯度方向新建立一颗决策树，迭代多少次就会生成多少颗决策树。GBDT的思想使其具有天然优势可以发现多种有区分性的特征以及特征组合，决策树的路径可以直接作为LR输入特征使用，省去了人工寻找特征、特征组合的步骤。

先用已有特征训练GBDT模型，然后利用GBDT模型学习到的树来构造新特征，最后把这些新特征加入原有特征一起训练模型。构造的新特征向量是取值0/1的，向量的每个元素对应于GBDT模型中树的叶子结点。当一个样本点通过某棵树最终落在这棵树的一个叶子结点上，那么在新特征向量中这个叶子结点对应的元素值为1，而这棵树的其他叶子结点对应的元素值为0。新特征向量的长度等于GBDT模型里所有树包含的叶子结点数之和。

举例说明。右边的图中的两棵树是GBDT学习到的，第一棵树有3个叶子结点，而第二棵树有2个叶子节点。对于一个输入样本点x，如果它在第一棵树最后落在其中的第二个叶子结点，而在第二棵树里最后落在其中的第一个叶子结点。那么通过GBDT获得的新特征向量为[0, 1, 0, 1, 0]，其中向量中的前三位对应第一棵树的3个叶子结点，后两位对应第二棵树的2个叶子结点。



## 2.2 回归树

一个回归树对应着输入空间（特征空间）的一个划分以及在划分单元上的输出值。假设已将输入空间划分为M个单元,并且在每个单元上有一个固定的输出值，于是回归树模型可以表示为：

(2.0

其中为指示函数，

当输入空间的划分确定时，可以用平方误差来表示回归树对于训练数据的预测误差。

现在的问题是如何对于特征空间进行划分。这里采用启发式的方法，选择第j个变量和它取的值s，作为切分变量和切分点，并定义两个区域

然后寻找最优切分变量j和最优切分点s，具体地，求解

(2.1

对固定输入变量j可以找到最优切分点s

(2.2

遍历所有输入变量，找到最优的切分变量j，构成一个对(j, s)。依次将输入空间划分为两个区域。接着，对每个区域重复上述划分过程，直到满足条件停止为止。这样就剩成了一棵回归树。

输入：训练数据集D；

输出：最小二乘回归树f(x)

在训练数据集所在的输入空间中，递归地将每个区域划分为两个子区域并决定每个子区域上的输出值，构建二叉决策树：

1. 选择最优切分变量j与最优切分点，求解：

遍历变量j，对固定的切分变量j扫描切分点s，选择使上式达到最小的值的对(j, s)

1. 用选定的对(j, s)划分区域并决定相应的输出值：

1. 继续对两个子区域调用步骤(1),(2)，直至满足停止条件，
2. 将输入空间划分为M个区域，生成决策树

## 2.3 提升树

提升树(Boosting Tree)模型可以表示为决策树的加法模型：

(2.3

其中，表示决策树，为决策树的参数，M为树的个数。

提升树算法采用前向分布算法，首先确定初始提升树，第m步的模型是

(2.4

其中，为当前模型，通过经验风险极小化确定下一棵决策树的参数，

(2.5

回归问题的提升树算法

输入：训练数据集T={(

输出：提升树

1. 初始化
2. For m = 1, 2, …, M

a)计算残差

b)拟合残差学习一个回归树，得到

c)更新

1. 得到回归问题提升树

## 2.4 梯度提升树

提升树利用加法模型和前向分布算法实现学习的优化过程，当损失函数是平方损失和指数损失函数时，每一步优化很简单，但是对于一般损失函数而言，往往每一步优化算法并不那么容易。针对这一问题，产生了梯度提升算法（Gradient boosting）。这是利用最速下降法的近似方法，其关键是利用损失函数的负梯度在当前模型的值

(2.6

作为回归问题提升树算法中残差的近似值，拟合一个回归树。

下面是GBDT的大概框架

输入：训练数据集T={(； 损失函数

输出：回归树

1. 初始化
2. For m = 1 to M：

a)For i = 1,2,…,N

b)对拟合一个回归树，得到第m棵树的叶节点区域

c)For :

d)更新

1. 输出

# 3.0 FFM

## 3.1 FM

FM(Factorization Machine)，提出于2010年，旨在解决稀疏数据下的特征组合问题。假设一个广告分类问题，根据用户和广告位相关特征，预测用户是否点击广告，数据如下：

|  |  |  |  |
| --- | --- | --- | --- |
| Clicked | Country | Day | Ad\_type |
| 1 | USA | 2016.9.1 | Movie |
| 0 | China | 2016.10.1 | Game |
| 1 | China | 2016.10.2 | Game |

将其转化为数值型特征

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Clicked | Country=USA | Country=China | Day=2016.9.1 | Day=2016.10.1 | Day=2016.10.2 | Ad\_type=Movie | Ad\_type=Game |
| 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 |

可以看出特征空间并且数据稀疏。同时通过观察可以发现某些特征经过关联后与结果的相关性会提高。例如”USA”与”Thanksgiving”、”China”与”Chinese New Year”这样的关联特征，对用户的点击有正向的影响。因此引入两个特征的组合非常有意义。

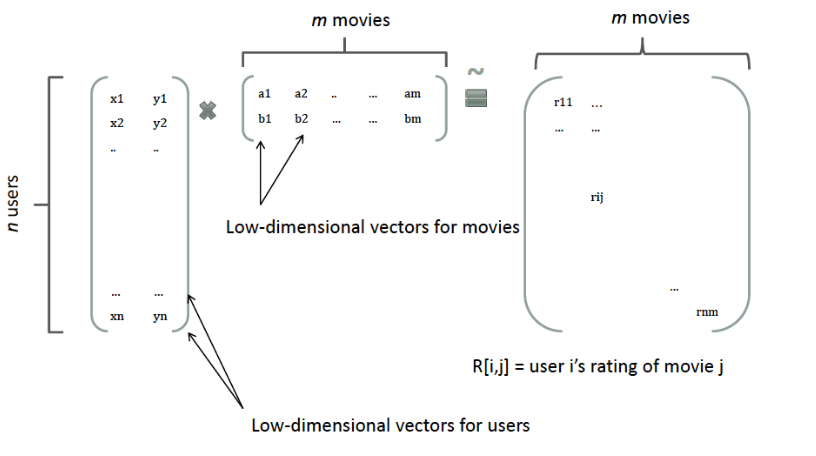
我们使用如下多项式模型：

(3.1

其中，n代表样本的特征数量，是第i个特征的值，代表第i个特征与第j个特征合成的组合特征，w是模型参数。

可以看出，组合特征的参数有个，任意两个参数都是独立的。然而，因为数据稀疏性的存在，二次项参数的训练是很困难的。

为了解决二次项参数训练的问题，可以从矩阵分解中得到思路。在model-based的协同过滤中，一个rating矩阵可以分解为user矩阵和item矩阵，每个user和item都可以采用一个隐向量表示。比如在下图中的例子中，我们把每个user表示成一个二维向量，同时把每个item表示成一个二维向量，两个向量的点积就是矩阵中user对item的打分。



类似地，所有二次项参数可以组成一个对称矩阵**W**，那么这个矩阵可以分解为，**V**的第j列便是第j维特征的隐向量，即>。因此FM的模型方程为

(3.2

## 3.2 FFM

FFM(Field-aware Factorization Machine)是FM的升级模型。通过引入field的概念，FFM把相同性质的特征归于同一个field。以上面的广告分类为例，”Day=2016.9.1”, “Day=2016.10.1”, “Day=2016.10.2”这三个特征都是代表日期的，可以放到同一个field中。在FFM中，每一维特征 ，针对其它特征的每一种field ，都会学习一个隐向量 。因此，隐向量不仅与特征相关，也与field相关。也就是说，“Day=2016.10.1”这个特征与“Country”特征和“Ad\_type"特征进行关联的时候使用不同的隐向量，这与“Country”和“Ad\_type”的内在差异相符，也是FFM中“field-aware”的由来。

假设样本的 n 个特征属于 f 个field，那么FFM的二次项有 nf个隐向量。而在FM模型中，每一维特征的隐向量只有一个。FM可以看作FFM的特例，是把所有特征都归属到一个field时的FFM模型。根据FFM的field敏感特性，可以导出其模型方程。

(3.3

假设我有如下数据：

|  |  |  |  |
| --- | --- | --- | --- |
| Field name | Field index | Feature name | Feature index |
| User | 1 | User=Su | 1 |
| Movie | 2 | Movie=3Idiots | 2 |
| Genre | 3 | Genre=Comedy | 3 |
| Price | 4 | Genre=Drama | 4 |
|  |  | Price | 5 |

那么，FFM的组合特征有10项，如下

其中，中i代表特征编号，j代表field编号。上式省略了特征的具体取值。

FFM模型采用logistic loss作为损失函数，和L2惩罚项，损失函数如下

(3.4

m为训练样本数量，是惩罚项系数。模型采用SGD优化，流程如下：

输入：训练数据集D，迭代次数n, 计算精度

输出：f(x)的极小点**x**

1. 初始化，随机生成模型参数
2. For k = 1 to n:

a)计算

b)计算梯度， 当，停止迭代；否则，令

c)一维搜索：求使得

d)更新

1. 输出x

如果要提升学习效率的话，算法应该可以改为L-BFGS优化。不过目前我见到FFM的实现都是使用SGD。

# 4.0 总结

模型由GBDT和FFM两个部分构成。GBDT用来挖掘有效的特征、特征组合。FFM模型既用来发现组合特征也用来预测最终点击的概率。FFM的输入有两种方式，一种是只输入GBDT的结果，另一种是输入GBDT的结果加上原本的特征。并没有找到比较两种方式的优劣的资料，只能通过后续实践来判断。