【模型融合】经典模型融合办法：线性模型和树模型的组合拳

推荐系统在技术实现上一般划分为三个阶段：挖掘、召回、排序。

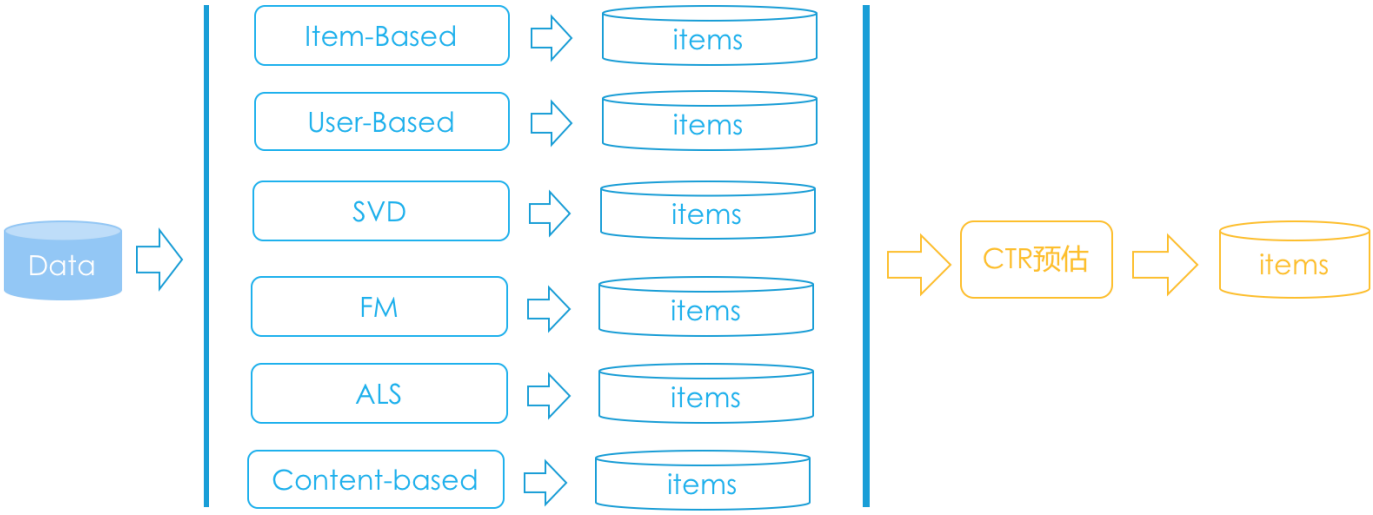
为什么要融合?

挖掘的工作就是对用户和物品做非常深入的结构化分析，庖丁解牛一样，各个角度各个层面的特征都被呈现出来，并且建好索引，供召回阶段使用，大部分挖掘工作都是离线进行的。

接下来就是召回，为什么会有召回？因为物品太多了，每次给一个用户计算推荐结果时，如果对全部物品挨个计算，那将是一场灾难，取而代之的是用一些手段从全量的物品中筛选出一部分比较靠谱的。

最后就是排序，针对筛选出的一部分靠谱的做一个统一的论资排辈，最后这个统一的排序就是今天要讲的主题：融合。

前面巴拉巴拉说了一段，画成图的话会好理解一些，示意图如下。



为什么要融合呢？这还得倒回去说一说召回是什么，以及这个阶段到底发生了什么？

在召回阶段，其实就是各种简单的、复杂的推荐算法，比如说基于内容的推荐，会产生一些推荐结果，比如基于物品的协同过滤会产生一些结果，矩阵分解会产生一些结果，等等。

总之，每种算法都会产生一批推荐结果，一般同时还附带给每个结果产生一个推荐分数，是各自算法给出来的。

于是问题就来了，这些不同算法产生的推荐分数，最后要一起排个先后，难道依据各自的分数吗？

这样是不行的，为什么？有几个原因：

1. 有个算法可能只给出结果，不给分数，比如用决策树产生一些推荐结果；
2. 每种算法给出结果时如果有分数，分数的范围不一定一样，所以不能互相比较，大家各自家庭背景不一样；
3. 即使强行把所有分数都归一化，仍然不能互相比较，因为产生的机制不同，有的可能普遍偏高，有的可能普遍偏低。

既然来自各个地方的状元凑在一起，谁也不服谁，那只能再举行一次入学考试了，这个入学考试就是融合模型。也就是，不同算法只负责推举出候选结果，真正最终是否推荐给用户，由另一个统一的模型说了算，这个就叫做模型的融合。

模型融合的作用除了统一地方军阀，还有集中提升效果的作用。在机器学习中，有专门为融合而生的集成学习思想。

今天要讲的一个典型的模型融合方案是：逻辑回归和梯度提升决策树组合，我可以给它取个名字叫做“辑度组合”。

“辑度组合”原理

在推荐系统的模型融合阶段，就要以产品目标为导向。举个简单的例子，信息流推荐，如果以提高CTR为目标，则融合模型就要把预估CTR作为本职工作，这个工作谁最能胜任呢，一直以来就是逻辑回归。

下面，我就来简单介绍一些常见的逻辑回归。

逻辑回归

CTR预估就是在推荐一个物品之前，预估一下用户点击它的概率有多大，再根据这个预估的点击率对物品排序输出。

逻辑回归常常被选来执行这个任务，它的输出值范围就是0到1之间，刚好满足点击率预估的输出，这是一个基础。因为逻辑回归是广义线性模型，相比于传统线性模型，在线性模型基础上增加了sigmoid函数。

下面就简单说说，逻辑回归如何做CTR预估？

我还是按照一直以来的套路来讲，先讲它在真正使用时怎么做的，再一步步往回看怎么得到所需要的条件。

在对召回阶段不同算法给出的候选物品计算CTR预估时，需要两个东西：

1. 特征；
2. 权重。

第一个是特征，就是用量化、向量的方式把一个用户和一个物品的成对组合表示出来。这里说的量化方式包括两种：实数和布尔。实数好理解，比如一个用户的年龄，一个用户平均在某个品类上每个月的开销，类似等等。

布尔，就是取值0或者1，针对两种类别形式的，比如用户所在的省、市，当时是白天还是晚上，物品的每一个标签。

用户和每一个候选物品都组一下CP，然后以这种特征化的方式表达出来，就可以计算了，否则类别形式的字段不能直接参与计算。

第二个是权重，每个特征都有一个权重，权重就是特征的话事权。在这场决定哪些物品最终有机会能走到前台的选秀过程中，用户和物品这对CP的所有特征都有投票权，只是同人不同命，每个特征的权重不一样，对最终计算CTR影响有大有小。

这个权重就很重要了，显然不能由愚蠢的人类来指定，需要模型自主从大量的历史数据中学习得到。

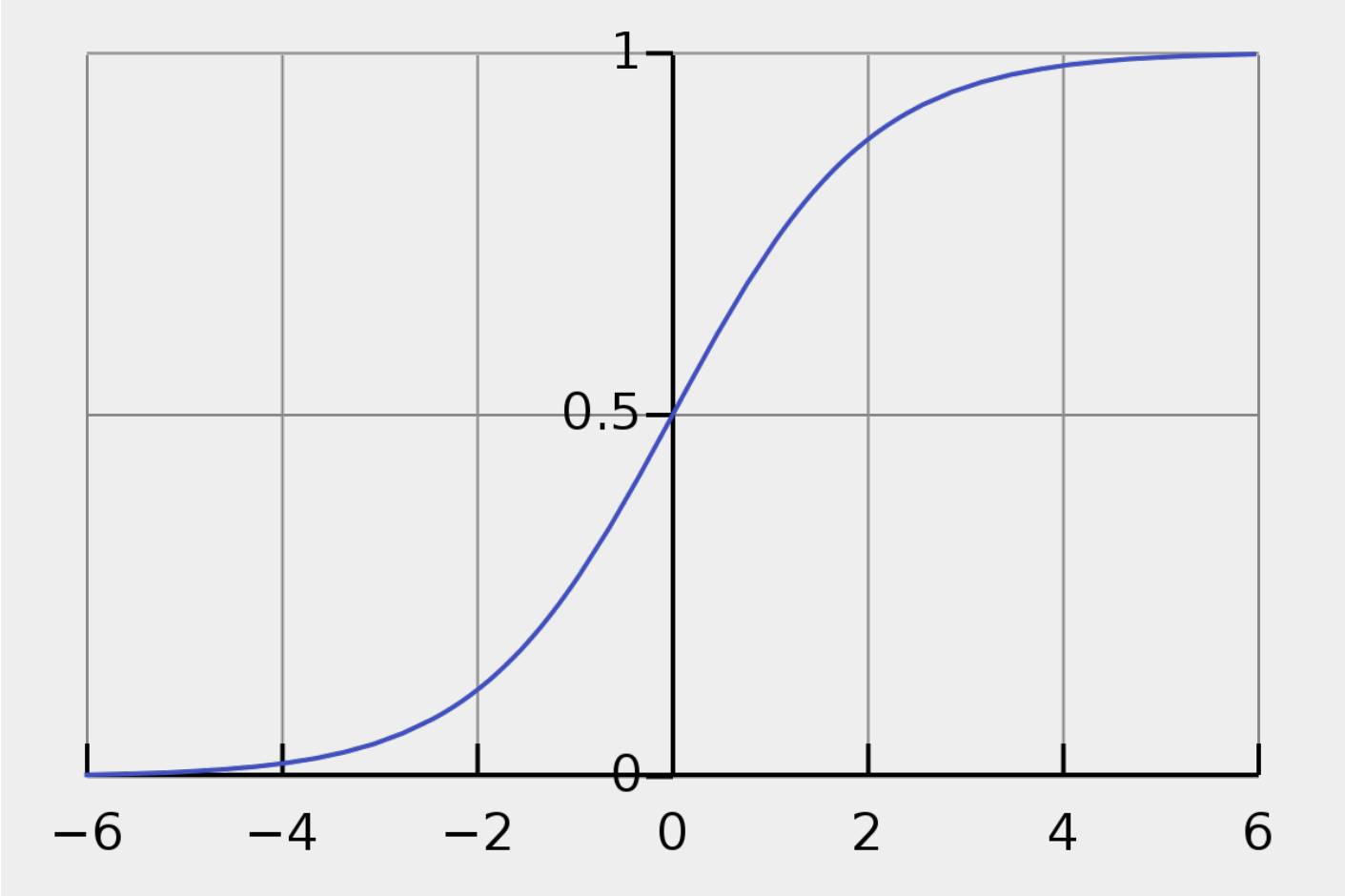
有了特征，它是一个向量，假如把它叫做x；还有特征的权重，也是一个维度和特征一样的向量，假如叫做w。

我们通过对x和w做点积计算，就得到了一个传统线性模型的输出，再用sigmoid函数对这个值做一个变换，就得到一个0到1之间的值，也就是预估的CTR。

这里所说的sigmoid函数长这个样子：

$$\sigma(w\times{x}) = \frac{1}{1+e^{-w\times{x}}}$$

这个函数曲线如图所示。



那看上去其实要做的就是两件事了：搞特征、学权重。

事实上的确如此，甚至前者占据更多的时间。逻辑回归特特征的取值都要求要在0到1之间。

甚至在一些领域，比如搜索广告，特征全都是布尔取值，只有出现和不出现两种，一旦遇到实数取值的特征，就将它划分成多个区间段，也变成了布尔取值。

除此之外，由于逻辑回归是广义线性模型，所谓广义就是因为加了sigmoid函数，所以很多非线性关系它无能为力。

比如说，有一天你发现“ID为233的用户喜欢买各种钢笔”这个事实，它可以有两个特征组合出来，一个是“ID为233”，是一个布尔特征，另一个是“物品为钢笔”，也是一个布尔特征，显然构造一个新特征，叫做“ID为233且物品为钢笔”。

只有两个原始特征都取值为1时，这个构造出的特征才会取值为1，这种组合就是非线性，逻辑回归本身对两个原始特征仅仅是线性加权，并不能很好地刻画这个组合关系，非得组合才能助它一臂之力。

类似这样的工作，行话都叫做特征工程，刚才举例所说的特征组合叫做二阶组合，还有三阶组合，只要你高兴，也没人拦着你搞四阶组合。

但是要注意，特征组合的难点在于：组合数目非常庞大，而且并不是所有组合都有效，只有少数组合有效。

需要不断去弄脏双手，脚上沾泥地从数据中发现新的、有效的特征及特征组合。

特征工程+线性模型，是模型融合、CTR预估等居家旅行必备。

权重那部分就是老生常谈了，简单说就是你准备好样本，喂给优化算法，优化算法再挤出新鲜的权重。

权重的学习主要看两个方面：损失函数的最小化，就是模型的偏差是否足够小；另一个就是模型的正则化，就是看模型的方差

是否足够小；都是希望模型能够有足够的生命力，在实际生产线上最好能和实验阶段表现一样好。

除了要学习出偏差和方差都较小的模型，还需要能够给工程上留出很多余地，具体来说就是两点，一个是希望越多权重为0越好，权重为0称之为稀疏，可以减小很多计算复杂度，并且模型更简单，方差那部分会可控。

另一个是希望能够在线学习这些权重，用户源源不断贡献他们的行为，后台就会源源不断地更新权重，这样才能实现生命的大和谐。

要学习逻辑回归的权重，经典的方法如梯度下降一类，尤其是随机梯度下降，这在前面讲矩阵分解时已经提到过，可以实现在实时数据流情形下，更新逻辑回归的权重，每一个样本更新一次。

但是随机梯度下降常被人诟病的是，它什么也表现不好，很难得到稀疏的模型，效果收敛得也很慢。

也就是模型预测结果在通往真正想要到达的靶心路上看上去像是喝醉了酒一样，歪歪斜斜，像是很随机，但是趋势上还是在朝损失函数下降的方向。

后来Google在2013年KDD上发表了新的学习算法：FTRL，一种结合了L1正则和L2正则的在线优化算法，现在各家公司都采用了这个算法。

这里也顺便提一句，这个专栏重点讲解的是推荐系统落地会用到的东西，尽量通俗易懂。如果深入到机器学习和人工智能其他分支，可以参考极客时间上洪亮劼老师的“AI技术内参”专栏。

对于我给你讲过的原理，希望可以让你有个直观的理解，在专栏结束后的图书出版计划中，我会在书中更加细致深入地讲原理，就有更多的代码和公式。

梯度提升决策树GBDT

前面提到，特征组合又能有效表达出数据中的非线性事实，但是发现成本却很高，需要花大量的人力和物力，那么有没有算法能够在这个阶段帮助到你呢?

答案是，有！就是用树模型。

树模型，可以理解为苏格拉底式的诘问，想象不断对一个样本提问：是男用户吗？是的话再问：是北上广的用户吗？不是的话则可以问：是月收入小于5000的用户吗?

这种不断提问按照层级组织起来，每次回答答案不同后再提出不同的问题，直到最后得出最终答案：用户对这个推荐会满意吗?

这就是树模型。树模型天然就可以肩负起特征组合的任务，从第一个问题开始，也就是树的根节点，到最后得到答案，也就是叶子节点，这一条路径下来就是若干个特征的组合。

树模型最原始的是决策树，简称DT，先驱们常常发现，把“多个表现”略好于“随机乱猜”的模型以某种方式集成在一起往往出奇效，所以就有树模型的集成模型。最常见的就是随机森林，简称RF，和梯度提升决策树，简称GBDT。

先讲一下剃度提升决策树的原理。按照其名字，我把它分成两部分：一个是GB，一个是DT。GB是得到集成模型的方案，沿着残差梯度下降的方向构建新的子模型，而DT就是指构建的子模型要用的决策树。

梯度提升决策树其实本意是用来做回归问题的，怎么回事呢?

举个例子好了。假如这里有以下这么几条样本。

喜欢养花	喜欢打游戏	喜欢帽子	年龄
0	1	1	13
0	1	0	14
0	1	0	15
1	1	1	25
0	1	1	35
1	0	0	49
1	1	1	68
1	0	0	71
1	0	1	73

现在有个任务是根据是否喜欢养花，喜欢打游戏，喜欢帽子来预测年龄，模型就是梯度提升决策树GBDT。假设我们设定好每个子树只有一层，那么三个特征各自按照取值都可以构成两分支的小树枝。

树根节点为：是否喜欢养花，左分支就是不喜欢，被划分进去的样本有13、14、15，35这四个年龄；右边的就是样本25、49、68、71、73。左边的样本均值是19.25，右边的样本均值是57.2。

树根节点为：是否喜欢打游戏，左分支是不喜欢，被划分进去就有49，71，73；右边是喜欢，被划分进去的样本有13、14、15、25、35、68。左边的均值是64，右边的均值是28.3。

树根节点为：是否喜欢帽子，左分支是不喜欢，被划分进去就有14、15、49、71；右边是喜欢，右边是13、25、35、68、73，左边均值是37.25，右边是42.8。

叶子节点上都是被划分进去的样本年龄均值，也就是预测值。这里是看哪棵树让残差减小最多，分别拿三个方案去预测每个样本，统计累积的误差平方和，三个分别是1993.55、2602、5007.95，于是显然第一棵树的预测结果较好，所以GBDT中第一棵树胜出。

接下来第二棵树如何生成呢？这里就体现出GBDT和其他提升算法的不同之处了，比如和Ada boost算法不同之处，GBDT用上

一棵树去预测所有样本，得到每一个样本的残差，下一棵树不是去拟合样本的目标值，而是去拟合上一棵树的残差。这里，就是去拟合下面这个表格。

喜欢养花	喜欢打游戏	喜欢帽子	年龄	上一棵树预测	残差
0	1	1	13	19.25	6.25
0	1	0	14	19.25	5.25
0	1	0	15	19.25	4.25
1	1	1	25	57.2	32.2
0	1	1	35	19.25	-17.75
1	0	0	49	57.2	8.2
1	1	1	68	57.2	-10.8
1	0	0	71	57.2	-13.8
1	0	1	73	57.2	-15.8

新一轮构建树的过程以最后一列残差为目标。构建过程这里不再赘述，得到第二棵树。如此不断在上一次建树的残差基础上构建新树，直到满足条件后停止。

在得到所有这些树后，真正使用时，是将它们的预测结果相加作为最终输出结果。这就是GBDT的简单举例。

这里有两个问题。

第一个，既然是用来做回归的，上面这个例子也是回归问题，如何把它用来做分类呢？那就是把损失函数从上面的误差平方和换成适合分类的损失函数，例如对数损失函数。

更新时按照梯度方向即可，上面的误差平方和的梯度就刚好是残差。对于CTR预估这样的二分类任务，可以将损失函数定义为：

$$-ylog§ - (1-y)log(1-p)$$

第二个，通常还需要考虑防止过拟合，也就是损失函数汇总需要增加正则项，正则化的方法一般是：限定总的树个数、树的深度、以及叶子节点的权重大小。

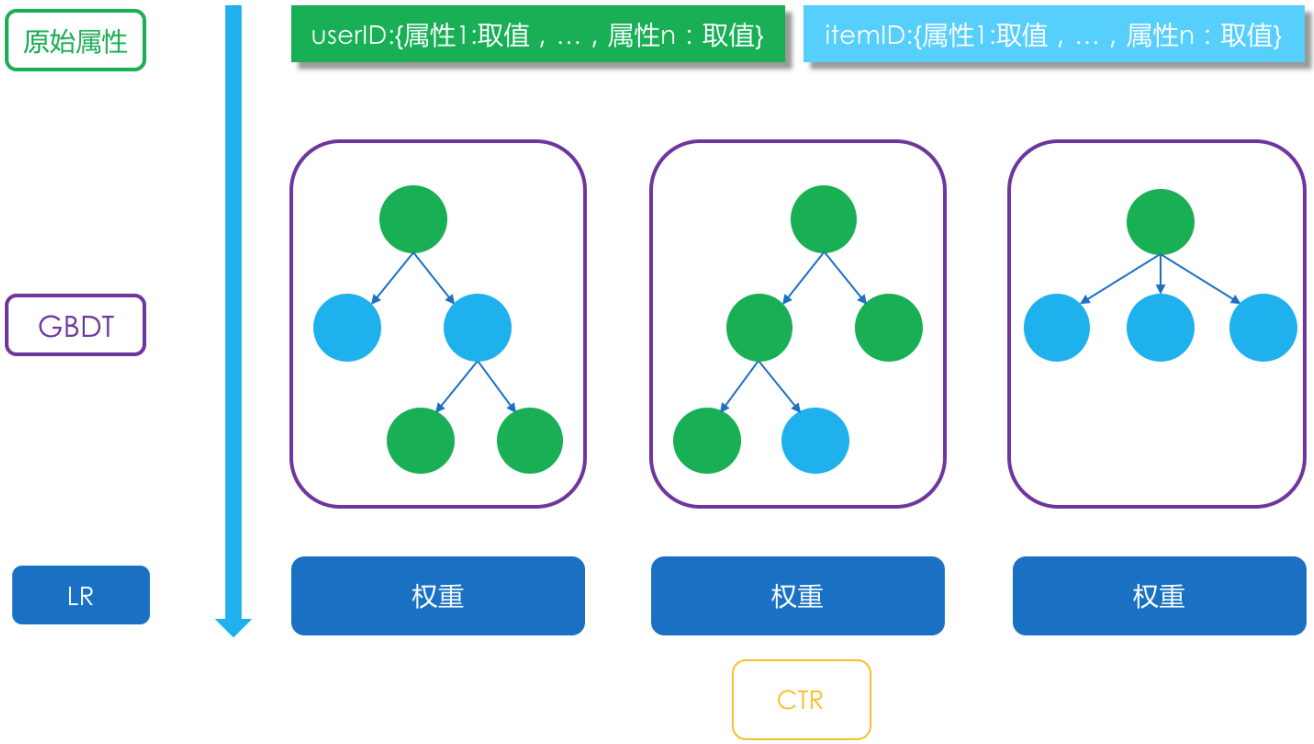
第三个，构建每一棵树时如果遇到实数值的特征，还需要将其分裂成若干区间，分裂指标有很多，可以参考xgboost中的计算分裂点收益，也可以参考决策树所用的信息增益。

二者结合

前面介绍了逻辑回归LR，以及剃度提升决策树GBDT的原理。实际上可以将两者结合在一起，用于做模型融合阶段的CTR预估。这是Facebook在其广告系统中使用的方法，其中GBDT的任务就是产生高阶特征组合。

具体的做法是：GBDT产生了N棵树，一条样本来了后，在每一棵树上都会从根节点走到叶子节点，到了叶子节点后，就是1或者0，点或者不点。把每一棵树的输出看成是一个组合特征，取值为0或者1，一共N棵树，每棵树i有 M_i 个叶子就相当于有M种组合，一棵树对应一个one-hot（独热）编码方式，一共就有$\sum_{i=1}^{N}{M_i}$个维度的新特征，作为输入向量进入LR模型，输出最终的结果。

示意图如下。



每一条样本，样本内容一般是把用户、物品、场景三类特征拼接在一起，先经过N棵GBDT树各自预测一下，给出自己的0或者1的预测结果，接着，这个N个预测结果再作为N个one-hot编码特征拼接成一个向量送入逻辑回归中，产生最终的融合预估结果。

另外，由于两者结合后用来做推荐系统的模型融合，所以也可以考虑在输入特征中加入各个召回模型产生的分数，也许会有用。

以上就是咱们的“辑度组合”原理，虽然简单，但在实际应用中非常的有效。

总结

今天我主要讲了简单的逻辑回归和梯度提升决策树，两者都是不太复杂的模型。并且无论是逻辑回归，还是梯度提升决策树，都有非常成熟的开源实现，可以很快落地。

由于篇幅限制，在梯度提升决策树那部分有一些细节被我略过了，你能自己手算出例子中的第二棵树是什么样的吗？欢迎留言一起讨论。感谢你的收听，我们下期再见。

精选留言

科技狗

真的太好了！还有一些困惑希望老师解答一下，gbdt的ntrees都是提前定好的，n颗树为什么产生的是n个特征，为什么不乘以叶子节点数？n颗树的建树过程和做特征组合的过程是浑然一体的还是先建树再做特征？

2018-04-02 12:28

漂浮

请教老师个问题，有必要区分用户群进行不同推荐策略模型的开发吗？比如，按照地域，长三角用户执行某种推荐策略，珠三角用户执行某种推荐策略？或者按照用户，学生执行某种策略，非学生执行另外一种？是不是在推荐效果有些明显差异的情况下才需要，差异不大的话，不太需要精细化做？

2018-04-03 14:32

明华

无刀老师，在实际情况下用户ID也需要作为一个特征吗?如果是，那这个特征学出来的权重的意义在哪?还有就是一个系统里如果用户ID用one hot编码那样特征不会很多吗?

2018-08-10 00:50

米乐乐果

楼上的几个朋友可以看看facebook那篇ctr预估的论文，更详细一点，是个不错的补充

2018-05-06 20:03

作者回复

对，那篇非常好！

2018-05-10 09:00

jacket

老师，最终送入逻辑回归的特征，仅仅是经过GBDT决策的结果，还是会加上原始特征向量呢？为什么？

2018-04-18 05:27

slvher

在Facebook那篇CTR论文中，GBDT起到特征变换器的作用，其每颗子树的叶子节点的输出把原始输入特征映射为以1-of-K方式编码的高阶组合特征（其中K为子树的叶子节点数）。也即，GBDT同时实现了高阶特征组合和特征值布尔化，故可提升LR模型效果。

本文对这个关键细节的解释不够清楚，感兴趣的话，一定要读原论文。

2018-09-17 15:17

作者回复

你说得对。

2018-10-30 14:39

Geek_b95d22

“除了要学习出偏差和方差都较小的模型，还需要能够给工程上留出很很多余地，具体来说就是两点，一个是希望越多权重为 0 越好，权重为 0 称之为稀疏，可以减小很多计算复杂度，并且模型更简单，方差那部分会可控。”

请问，如果很多的权重都是0，其实意味着这个特征对于结果是没有什么影响的，也就是这个特征其实是没啥意义的，特征工程的目的是生成很多特征，模型训练又希望大多数的特征权重为0，那这两个步骤似乎有些矛盾？

还是说特征工程是尽可能多地寻找特征，而训练是把其中海量特征里最有用的特征（训练前未知）找出来？

2018-09-12 09:35

arfa

老师好，请问用户id和itemid是否作为特征，多谢

2018-09-11 09:05

mgxs

你好，请问一下融合的时候，样本的标签是由召回阶段不同算法的预测结果构成的吗？比如某个算法预测某个样本喜欢，则该样本类别为1。

2018-06-14 11:52

kijiang

老师，请教一个问题，在本篇文章的图1里，就是排序，召回，融合示意图里，中间的挖掘算法部分，看到了svd，als，fm。我的理解是als是求解fm的一种手段，是fm求解的一部分，为何要独立出来呢？我发现不少资料都会单独把als放在一个挖掘算法中，是否是业内一种约定的写法？

2018-06-11 10:39

作者回复

你理解没错，als是求解svd及很多模型的一个方法。这里说的als通常值得als原始论文中那个模型，用于挖掘召回的，fm常用于CTR 预估。

2018-07-23 20:56

kijiang

老师，请教一个问题，在ctr阶段，使用逻辑回归，将用户与物品组成cp，然后抽取特征。这个特征，与挖掘时期对用户利用各种算法建立的特征，和对物品采取各种手段建立的特征，有何不同？这个阶段的特征提取，是否主要靠人工？

2018-06-07 14:09

atlas

老师，召回使用了多个算法召回一些内容，在召回的过程不需要考虑哪个算法最优吗？要是在召回的阶段已经考虑了最好的算法，那么该算法产出的内容应该是最好的，为什么还需要融合排序这个步骤？

2018-05-19 23:52

cook150

facebook 那篇写CTR预估的论文叫什么呢？

2018-05-12 07:44

zgl

请问逻辑回归重排序后，对所有用户特征权重岂不是都一样的？只不过每个人特征的值不一样，是不是这个意思？

2018-04-29 13:16

derek

老师，我在考虑一个问题，召回策略这么多，如果想要保证策略的多样性，同时又保证线上收益，这种问题有什么比较好的思考切入点吗？如果用规则做，那样非常不优雅

2018-04-25 22:16

卓越

GBDT送到LR的向量应该是叶子结点的个数吧？每个叶子结点表示一系列特征的组合。

2018-04-20 14:21

林彦

谢谢陈老师的无私和专业的分享。学到了好多。

1. 第二棵树用是否喜欢打游戏来作根节点，累积的误差平方和是1764.57，用是否喜欢戴帽子来作根节点，累积的误差平方和是1986.706。因此第二棵树选择用是否喜欢打游戏。49，71，73岁的人的预测值是-7.13，13、14、15、25、35、68岁的人

的预测值是3.57。所有样本按原始顺序的残差(保留小数点后2位）是-2.68,-1.68,-0.68,-28.63,19.32,-15.33,14.37,6.67,8.67。

2. 请问损失函数−ylog(p)−(1−y)log(1−p)中y是预测值，这个预测值是0或1对应点或不点，还是一个点击的概率值。这里的p对应什么值，如何计算？这个损失函数的公式看着和某些文章中逻辑回归的一类损失函数的计算公式有些接近。其中里面的p的位置对应的是文中提到的sigmoid 函数σ(w×x)。

3. N 棵 GBDT 树对应N个特征组合我的理解是每一个样本只会被分配到一棵树的某个叶子节点上，相应这棵树从树根到这个叶子的所有问题回答的是否组合就是这个样本的特征组合。“到了叶子节点后，就是 1 或者 0，点或者不变。”这句话我有点不太明白，是不是应该是“点或者不点”?

4. 文中有2个数值可能有点小的错误。(1)文中选择第一棵树时是否喜欢戴帽子的累积的误差平方和我算出来的是5125.55；(2)第2个数据表格里35岁样本的残差应该是-15.75。

2018-04-05 21:57