



第九章

特征工程模型化的开端



教学目标

- 理解GBDT为何可以作为特征组合
- 理解GBDT进行特征组合的过程
- 理解GBDT+LR组合模型的意义
- 理解LS-PLM模型的主要结构
- 了解LS-PLM模型的优点

目录

1、GBDT简单回顾

2、GBDT+LR组合模型介绍

3、LS-PLM模型介绍

4、LS-PLM模型的优点

5、从深度学习角度重新审视
LS-PLM模型



9.1 GBDT简单回顾

DT – Decision Tree决策树，GB是Gradient Boosting，是一种学习策略，GBDT的含义就是用Gradient Boosting的策略训练出来的DT模型。模型的结果是一组回归分类树组合 (CART Tree Ensemble): $T_1 \dots T_K$ 。其中 T_j 学习的是之前 $j-1$ 棵树预测结果的残差，这种思想就像准备考试前的复习，先做一遍习题册，然后把做错的题目挑出来，在做一次，然后把做错的题目挑出来在做一次，经过反复多轮训练，取得最好的成绩。而模型最后的输出，是一个样本在各个树中输出的结果的和：

$$\bar{y} = \sum_{k=1}^K f_k(x), f_k \in \Gamma, f_k \text{表示样本到树输出的映射}$$

9.1 GBDT简单回顾

● 那么GBDT中的树是怎么形成的呢？

(1) 初始化 $f_0(x) = 0$

(2) 对 $m=1, 2, \dots, M$

(a) 计算残差

$$r_{mi} = y_i - f_{m-1}(x_i), i = 1, 2, \dots, N$$

(b) 拟合残差 r_{mi} 学习一个回归树, 得到 $h_m(x)$

(c) 更新 $f_m(x) = f_{m-1} + h_m(x)$

(3) 得到回归问题提升树

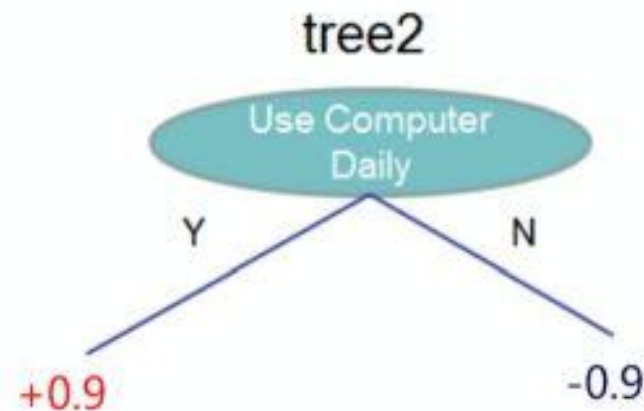
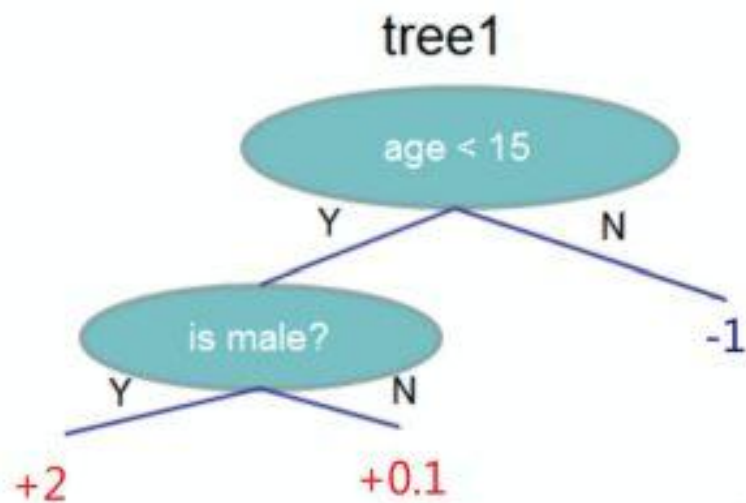
$$f_M(x) = \sum_{m=1}^M h_m(x)$$

对于回归问题, 我们一般采用平方损失作为损失函数, 目标是找到使损失函数值最小的最佳切分点。

对于分类问题我们一般采用两种损失函数: 指数损失函数和对数似然损失函数。我们用的是类别的预测概率值和真实概率值的差来拟合损失






9.1 GBDT简单回顾

假设我们要预测一个人是否会喜欢电脑游戏，特征包括年龄，性别是否为男，是否每天使用电脑。标记（label）为是否喜欢电脑游戏，假设训练出如下模型



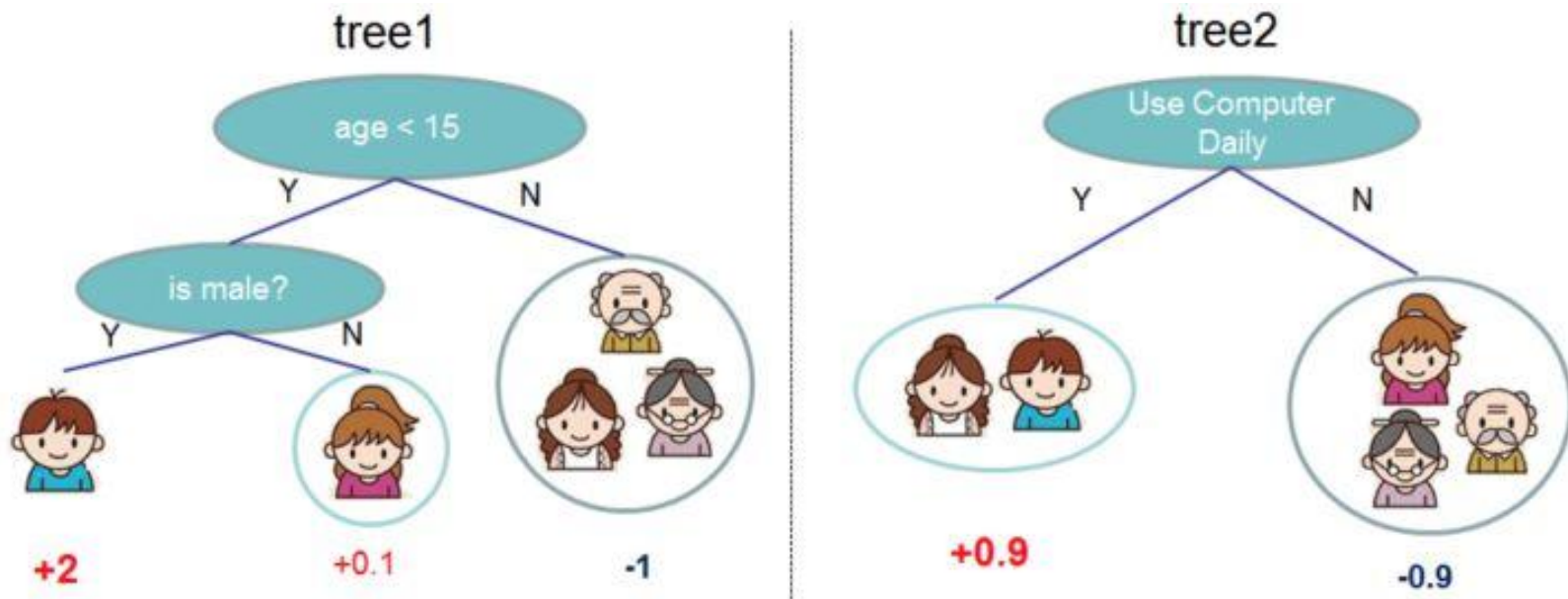
9.1 GBDT简单回顾

该模型又两棵树组成, T_1 使用 $\text{age} < 15$ 和 is male 作为内节点, 叶子节点是输出的分数。 T_2 使用是否每日使用电脑作为根节点。假设测试样本如下:

样本\特征	Age	is male	Use computer daily	Like CG
	79	1	0	?
	75	0	0	?
	29	0	1	?
	5	1	1	?
	8	0	0	?

9.1 GBDT简单回顾

样本在两棵树中所在的叶节点如下：



最后对某样本累加它所在的叶子节点的输出值，例如：

$$f(\text{boy icon}) = 2 + 0.9 = 2.9$$

$$f(\text{elderly man icon}) = -1 - 0.9 = -1.9$$

目录

1、GBDT简单回顾

3、LS-PLM模型介绍

5、从深度学习角度重新审视
LS-PLM模型



2、GBDT+LR组合模型介绍

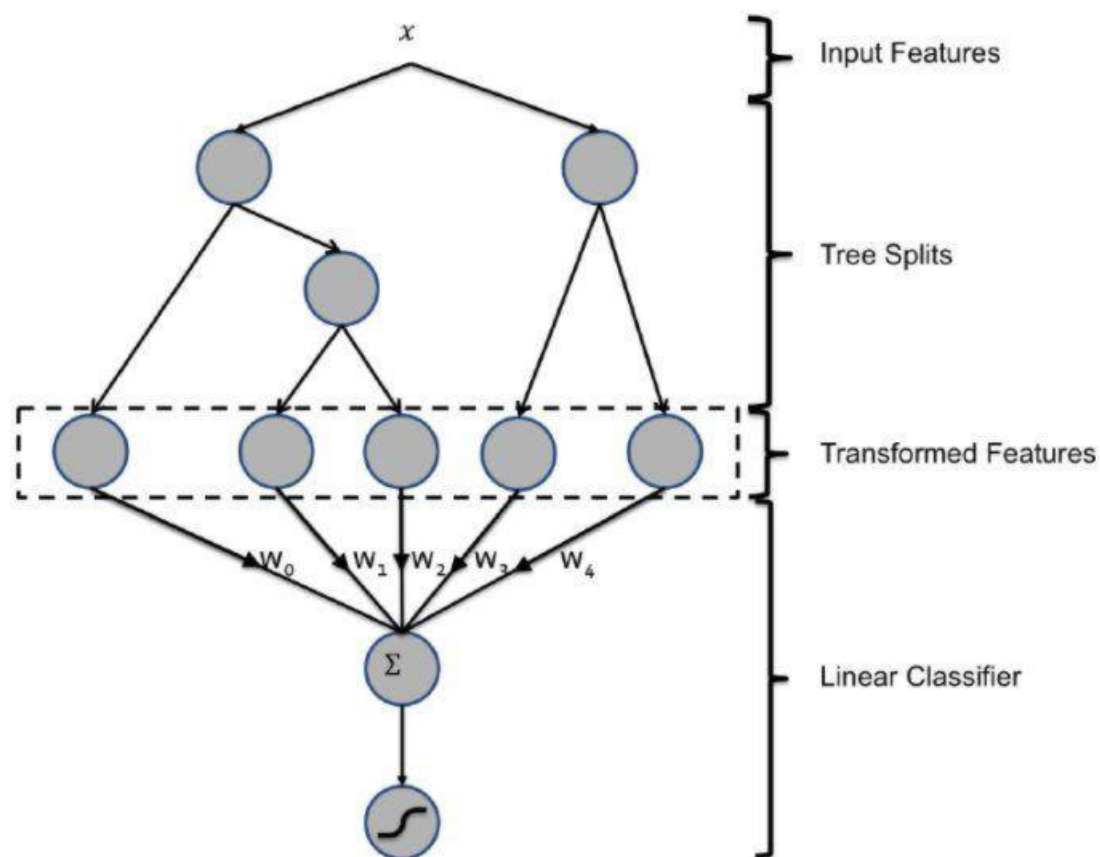
4、LS-PLM模型的优点

9.2 GBDT+LR组合模型介绍

FFM模型采用引入特征域的方式增强了模型的表达能力，但FFM只能够做二阶的特征交叉，如果要继续提高特征交叉的维度，不可避免的会发生组合爆炸和计算复杂度过高的情况。那么有没有其他的方法可以有效的处理高维特征组合和筛选的问题？2014年，Facebook提出了一种利用GBDT自动进行特征筛选和组合，进而生成新的离散特征向量，再把该特征向量当作LR模型输入，预估CTR的模型结构,也就是GBDT+LR。

9.2 GBDT+LR组合模型介绍

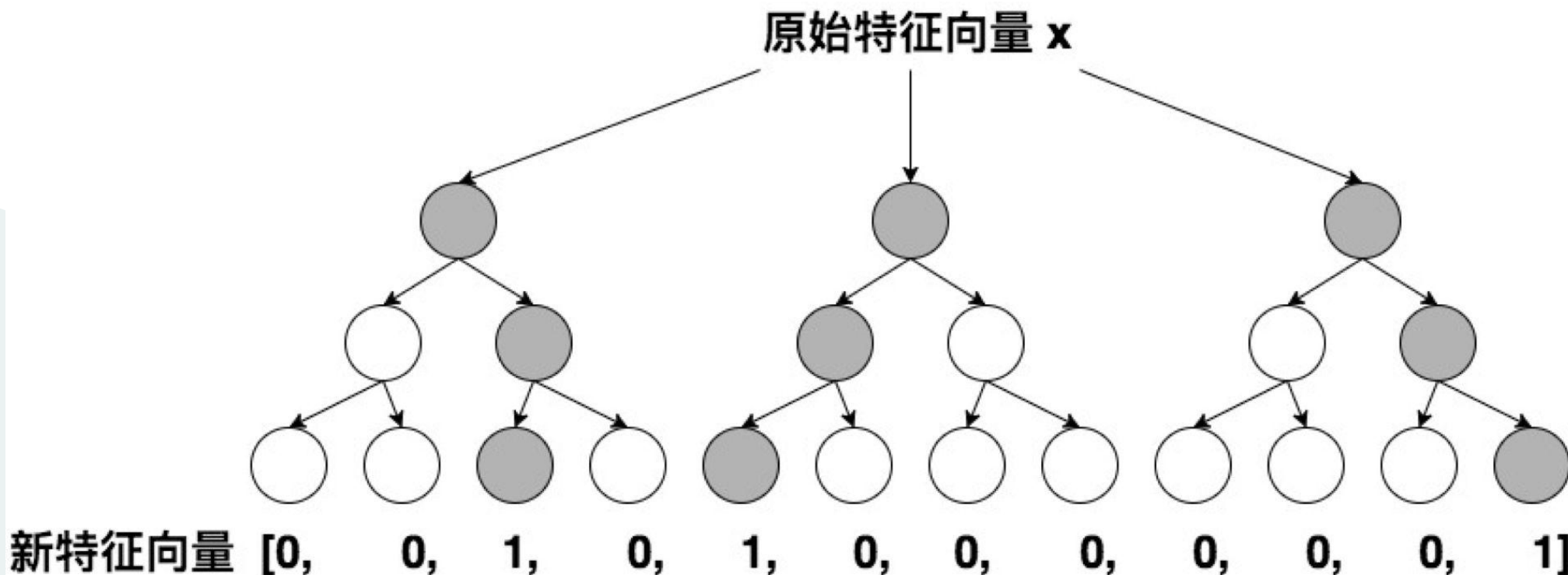
需要强调的是，用GBDT构建特征工程，和利用LR预估CTR两步是独立训练的。所以不存在如何将LR的梯度回传到GBDT这类复杂的问题



GBDT是由多棵回归树组成的树林，后一棵树利用前面树林的结果与真实结果的残差做为拟合目标。每棵树生成的过程是一棵标准的回归树生成过程，因此每个节点的分裂是一个自然的特征选择的过程，而多层节点的结构自然进行了有效的特征组合，也就非常高效的解决了过去非常棘手的特征选择和特征组合的问题。

9.2 GBDT+LR组合模型介绍

利用训练集训练好GBDT模型之后，就可以利用该模型完成从原始特征向量到新的离散型特征向量的转化。具体过程是这样的，一个训练样本在输入GBDT的某一子树后，会根据每个节点的规则最终落入某一叶子节点，那么我们把该叶子节点置为1，其他叶子节点置为0，所有叶子节点组成的向量即形成了该棵树的特征向量，把GBDT所有子树的特征向量连接起来，即形成了后续LR输入的特征向量。



9.2 GBDT+LR组合模型介绍

由于决策树的结构特点，事实上，决策树的深度就决定了特征交叉的维度。如果决策树的深度为4，通过三次节点分裂，最终的叶节点实际上是进行了3阶特征组合后的结果，如此强的特征组合能力显然是FM系的模型不具备的。但由于GBDT容易产生过拟合，以及GBDT这种特征转换方式实际上丢失了大量特征的数值信息，因此我们不能简单说GBDT由于特征交叉的能力更强，效果就比FFM好，在模型的选择和调试上，永远都是多种因素综合作用的结果。

GBDT+LR比FM重要的意义在于，它大大推进了特征工程模型化这一重要趋势，某种意义上来说，之后深度学习的各类网络结构，以及embedding技术的应用，都是这一趋势的延续。

9.2 GBDT+LR组合模型介绍

思考：

- 1、为什么使用GBDT作为LR的前置模型，为什么不选择随机森林或其他树模型？
- 2、既然点击率预估是一个分类问题，为什么不直接使用GBDT做分类器？

目录

1、GBDT简单回顾

3、LS-PLM模型介绍

5、从深度学习角度重新审视
LS-PLM模型



2、GBDT+LR组合模型介绍

4、LS-PLM模型的优点

9.3 LS-PLM模型介绍

LS-PLM (Large Scale Piece-wise Linear Model) 模型是阿里巴巴2012年的时候提出(2017年发表论文)来的点击率预估模型，它利用分段方式对数据进行拟合，相比LR模型，能够学习到更高阶的特征组合。它的另一个更广为人知的名字是MLR (Mixed Logistic Regression) 。

9.3 LS-PLM模型介绍

MLR就像它的名字一样，由很多个LR模型组合而成。用分片线性模式来拟合高维空间的非线性模式，形式化表述如下：

$$p(y = 1|x) = g\left(\sum_{j=1}^m \sigma(u_j^T x) \eta(w_j^T x)\right)$$

给定样本 x ，模型的预测 $p(y|x)$ 分为两部分：首先根据 $\sigma(u_j^T x)$ 分割特征空间为 m 部分，其中 m 为给定的超参数，然后对于各部分计算 $\eta(w_j^T x)$ 作为各部分的预测。函数 $g(\cdot)$ 确保了我们的模型满足概率函数的定义。当我们将softmax函数作为分割函数 $\sigma(x)$ ，将sigmoid函数作为拟合函数 $\eta(x)$ 的时候，该模型为：

$$p(y = 1|x) = \sum_{i=1}^m \frac{\exp(u_i^T x)}{\sum_{j=1}^m \exp(u_j^T x)} \cdot \frac{1}{1 + \exp(-w_i^T x)}$$

9.3 LS-PLM模型介绍

此时我们的混合模型可以看做如下数学形式的抽象：

$$p(y = 1|x) = \sum_{i=1}^m p(z = i|x)p(y|z = i, x)$$

目标损失函数为：

$$\arg \min_{\Theta} f(\Theta) = \text{loss}(\Theta) + \lambda \|\Theta\|_{2,1} + \beta \|\Theta\|_1$$

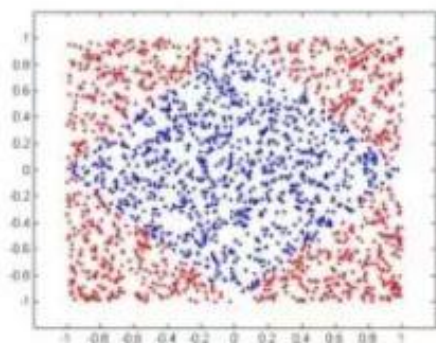
$$\text{loss}(\Theta) = - \sum_{t=1}^n \left[y_t \log(p(y_t = 1|x_t, \Theta)) + (1 - y_t) \log(p(y_t = 0|x_t, \Theta)) \right]$$

9.3 LS-PLM模型介绍

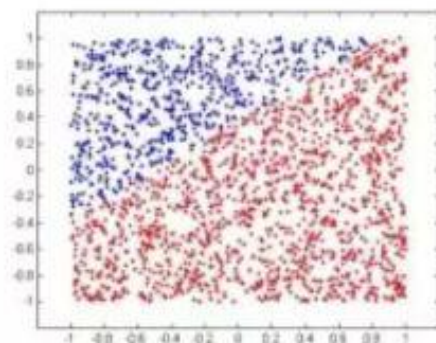
MLR模型首先用聚类函数 σ 对样本进行分类（这里的 σ 采用了softmax函数，对样本进行多分类），再用LR模型计算样本在分片中具体的CTR，然后将二者进行相乘后加和。

其中超参数分片数 m 可以较好地平衡模型的拟合与推广能力。当 $m=1$ 时MLR就退化为普通的LR， m 越大模型的拟合能力越强，但是模型参数规模随 m 线性增长，相应所需的训练样本也随之增长。在实践中，阿里给出了 m 的经验值为12。

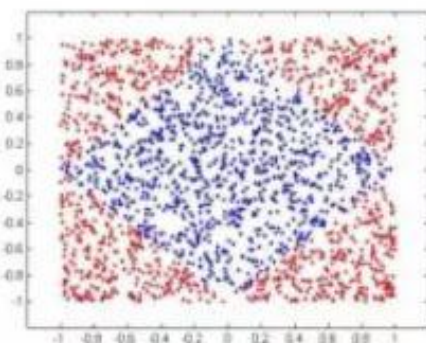
下图中MLR模型用4个分片可以完美地拟合出数据中的菱形分类面。



训练数据



LR模型



MLR模型

9.3 LS-PLM模型高级特性

在阿里妈妈团队写的MLR分享文章里面，提及了以下几点MLR的高级特性：

1、领域知识先验，这个分别体现在上文说到的dividing function和fitting function，他是基于领域的先验知识来设定dividing function和fitting function，比方说以用户的特征设计分片函数，以广告的特征设计拟合函数，这是基于不同人群具有聚类特性，同一类人群对广告有类似的偏好这样一个前提，此外这样的结构先验也有助于缩小解空间，更容易收敛

$$x = (x_1, x_2), \quad x_1 \in R^{d_1}, x_2 \in R^{d_2}, x \in R^{d_1+d_2}$$

$$f(x; \theta) = \sum_{j=1}^m \pi_j(x_1) \cdot \eta_j(x_2) \quad \leftarrow \text{结构先验/正则}$$

$$= \sum_{j=1}^m \left[\frac{\exp(\mu_j * x_1)}{\sum_{i=1}^m \exp(\mu_i * x_1)} \right] \cdot \left[\frac{1}{1 + \exp(-w_j * x_2)} \right]$$

x1 : 聚类参数
决定空间的划分

x2 : 分类参数
决定空间内的预测

2、加入线性偏置，因为特征的差异导致点击率天然存在一些差异，比如说位置和资源位，所以在损失函数中加入如公式所示的线性偏置，实践中对位置bias信息的建模，获得了4%的RPM提升效果。

$$x = (x_1, x_2), \quad x_1 \in R^{d_1}, x_2 \in R^{d_2}, x \in R^{d_1+d_2}$$

$$\begin{aligned} f(x; \theta) &\approx \left(\sum_{j=1}^m \pi_j(x_1) \cdot \eta_j(x_1) \right) \cdot \eta(x_2) \\ &= \left(\sum_{j=1}^m \frac{\exp(\mu_j x_1)}{\sum_{i=1}^m \exp(\mu_i x_1)} \cdot \frac{1}{1 + \exp(-w_j x_1)} \right) \cdot \frac{1}{1 + \exp(-w x_2)} \end{aligned}$$

✓ **Position bias**

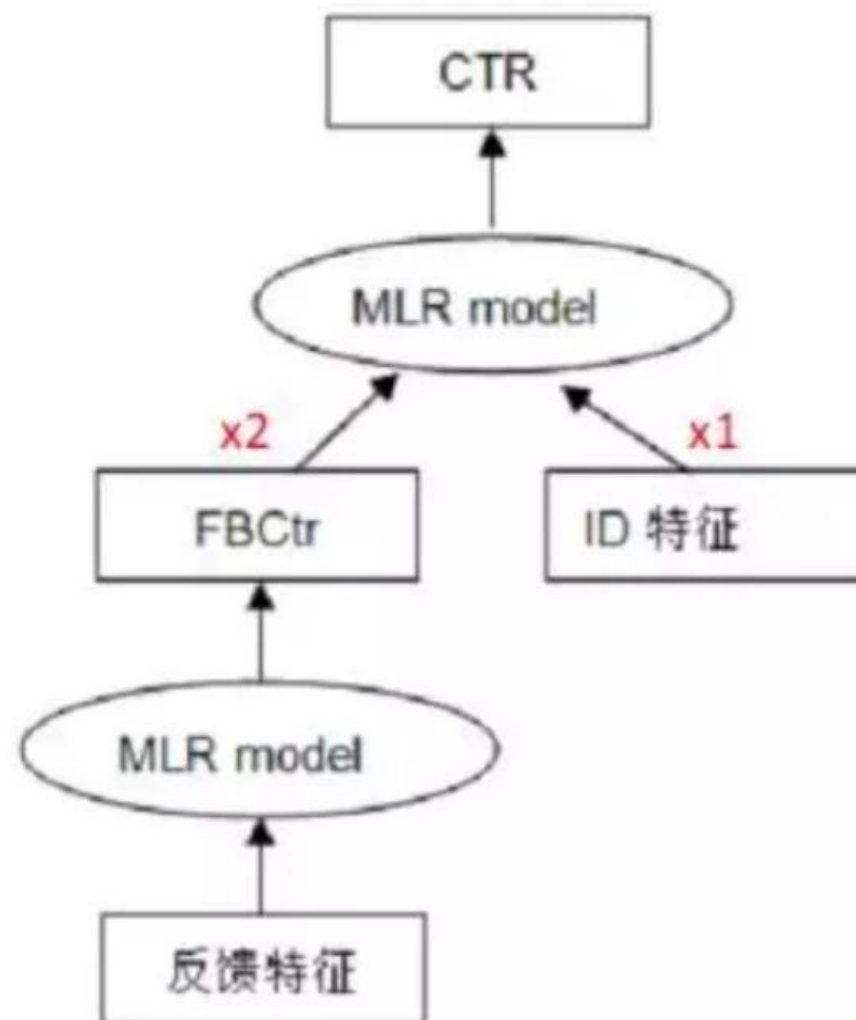
排名第1位和第5位的样本，点击率天然存在差异

✓ **Sample bias**

pc和mobile上的样本，点击率天然存在差异

9.3 LS-PLM模型高级特性

3、模型级联，与LR模型级联式联合训练，实践中发现一些强feature配置成级联模式有助于提高模型的收敛性，比如以统计反馈类特征构建第一层模型，它的输出(如下图中的FBCtr)级联到第二级大规模稀疏ID特征体系中去，这样能够有助于获得更好的提升效果。



反馈特征常用的如反馈CTR，是指系统上线一段时间之后得到的历史CTR值

9.3 LS-PLM模型高级特性

4、增量训练。实践证明，MLR通过结构先验进行pretrain，然后再增量进行全空间参数寻优训练，会获得进一步的效果提升。同时增量训练模式下模型达到收敛的步数更小，收敛更为稳定。在我们的实际应用中，增量训练带来的RPM增益达到了3%。

$$x = (x_1, x_2), \quad x_1 \in R^{d_1}, x_2 \in R^{d_2}, x \in R^{d_1+d_2}$$

Step 1: 结构化先验预训练

$$\begin{aligned} f(x; \theta_1) &= \sum_{j=1}^m \pi_j(x_1) \cdot \eta_j(x_2) \quad \leftarrow \text{结构先验} \\ &= \sum_{j=1}^m \boxed{\frac{\exp(\mu_j * x_1)}{\sum_{i=1}^m \exp(\mu_i * x_1)}} \cdot \boxed{\frac{1}{1 + \exp(-w_j * x_2)}} \end{aligned}$$

Step 2: 以 θ_1 为初值在全空间增量化训练

$$f(x; \theta_2) = \sum_{j=1}^m \pi_j(x) \cdot \eta_j(x) \quad \boxed{\theta_2 \text{ 初始化为 } \theta_1}$$

9.3 LS-PLM模型介绍

阿里妈妈团队主要将MLR主要应用于定向广告的CTR预估和定向广告的Learning to Match:

对于定向广告CTR, 他们将用户画像特征、用户历史行为特征、广告画像特征组成2亿为的embedding向量, 直接放到MLR模型中训练, 并且采用了线性偏置、领域知识先验、增量训练的高级特征, 提升了CTR预估的精度。

Learning to Match, 即基于用户的人口属性、历史行为等信息来猜测用户可能感兴趣的广告集合, 一般会使用规则匹配和协同过滤来做召回模块, 阿里妈妈研发了基于MLR的learning to match算法框架, 首先基于用户的行为历史来学习用户个性化兴趣, 召回候选集, MLR框架很容易融合将不同的特征源、标签体系, 省去交叉组合的成本, 灵活性很高。

目录

1、GBDT简单回顾

2、GBDT+LR组合模型介绍

3、LS-PLM模型介绍

4、LS-PLM模型的优点

5、从深度学习角度重新审视
LS-PLM模型



9.4 LS-PLM模型的优点

LS-PLM采用了分治的思想，先分成几个局部再用线性模型拟合，这两部都采用监督学习的方式，来优化总体的预测误差，总的来说有以下优势：

- ◆ 端到端的非线性学习：从模型端自动挖掘数据中蕴藏的非线性模式，省去了大量的人工特征设计，这使得MLR算法可以端到端地完成训练，在不同场景中的迁移和应用非常轻松。通过分区来达到拟合非线性函数的效果；
- ◆ 可伸缩性（scalability）：与逻辑回归模型相似，都可以很好的处理复杂的样本与高维的特征，并且做到了分布式并行；
- ◆ 稀疏性：对于在线学习系统，模型的稀疏性较为重要，所以采用了L1和L2,1正则化，模型的学习和在线预测性能更好。当然，目标函数非凸非光滑为算法带来了新的挑战。

目录

1、GBDT简单回顾

3、LS-PLM模型介绍

5、从深度学习角度重新审视
LS-PLM模型



2、GBDT+LR组合模型介绍

4、LS-PLM模型的优点

9.5 深度学习角度审视LS-PLM

LS-PLM可以看做一个加入注意力(Attention)机制的三层神经网络模型，其中输入层是样本的特征向量，中间层是由 m 个神经元组成的隐层，其中 m 是分片的个数，对于一个CTR预估问题，LS-PLM的最后一层是由单一神经元组成的输出层。

那么，注意力机制又是在哪里应用的呢？其实是在隐层和输出层之间，神经元之间的权重是由分片函数得出的注意力得分来确定的。也就是说，样本属于哪个分片的概率就是其注意力得分。

内容回顾

1、GBDT简单回顾

3、LS-PLM模型介绍

5、从深度学习角度重新审视
LS-PLM模型



2、GBDT+LR组合模型介绍

4、LS-PLM模型的优点



Thank you!