**特征工程**

***为什么说， 用物料的后验消费数据做召回存在“幸存者偏差“？能将这些消费数据用于排序吗？***

辨证的看，一个物料的后验指标好，只能说明推荐系统把他推荐给了合适的人，并不意味着把它推给任何人都能取得这个好的效果。这里面存在“幸存者偏差“；另一方面，如何这些后验指标参与精排，”幸存者偏差“影响不会很大，毕竟交给精排模型的物料已经经过了召回、粗排的筛选，或多或少适合当前用户相关的，这些物料的后验指标还是有参考意义的。

***使用物料的后验消费数据做召回， 会放大“马太效应”， 对新物料不友好， 如何缓解？***

后验指标好的物料会被排的靠前，获得更多的曝光与点击，后验指标会更好，形成正向循环。新的物料的后验指标不好甚至没有，排名靠后而较少获得曝光机会，后验指标迟迟得不到改善，形成负向循环。（如何缓解？冷启动问题？）

***解释什么是bias特征？你能举出哪些bias特征的例子？***

我们认为用户点击与否反映了用户正式的兴趣爱好，但严格来讲，以上假设并不成立。实践中，我们无法让所有物料在一个决定公平的环境中供用户挑选，这也就意味着用户的选择并非完全处于他的兴趣爱好，用户的点击未必是他喜欢的，未点击的不代表用户就一定不喜欢。这种不可避免引入的不公平因素叫做bias。例如1. Position bias：“视频5”是用户喜欢的，但是排名靠后，没有点击，将其作为负例就是bias。2. 视频年龄 = 当前时间-上传时间，上传早的后验指标更好，排名靠前，上传晚的没有积累起足够好的后验数据，排名靠后。

***bias特征怎样接入模型？能否和其他正常特征一起喂入DNN底层？为什么？***

解决方式是：1. Above Click, 只有位于被点击物料上方的未点击物料才被纳入负样本。2. 将偏差特征通过一个线性层接入模型，使得模型有足够多的信息来给用户反馈找出合理的解释，绝不能和其他正常特征一起喂入DNN，之后这样才能保证预测时无论伪特征的取值如何，都不会改变排序结果。P 24

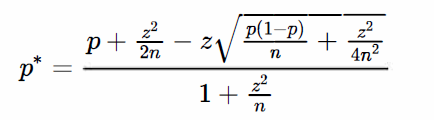
***某男性新用户对“体育”这个分类的喜好程度未知， 如何填充？***

最常规的做法是那所有样本在特征上的mean或者median替代。再精细些，用所有男性用户对“体育“类视频的喜好程度来填充。更合理的做法是训练模型来预测缺失值，利用比较容易得到的人口属性（性别，年龄等）预测新用户对某个内容分类、标签的喜爱程度。

***某新物料的后验指标未知， 如何填充？***

最常规的做法是那所有样本在后验指标上的mean或者median替代。更合理的做法是训练模型来预测缺失值。对于新物料，利用物料的静态画像（分类，标签，品牌，价位）预测它的动态画像（CTR, 平均观看时长，平均销售额）。

***某个物料曝光2次， 被点击1次， 如何计算它的CTR?***

使用“威尔逊区间平滑”“ 

p = 0.5, z = 1.96, n = 2 -🡪 p\* = (0.5+(1.96\*1.96)/2\*2 – 1.96 \* sqrt(0.5\*(1-0.5)+(1.96\*1.96)/(4\*2\*2))) / (1+(1.96\*1.96)/2) = 0.094

***有一个特征“某文章过去1天的点击率是10%", 如何将其构建成一个类别特征， 并喂入推荐模型？***

统计所有的文章过去一天的点击率，排序后，将整个值域的N个分位数（percentile）作为个桶的边界，确保落入各个桶的样本数大致相同。或者单独使用所有的点击率，与目标值拟合一棵决策树，然后，将每个点击率喂入决策树，这个值落入的最终节点的编号就是离散化的结果。

使用 Feature Hashing方法，将特征名字符串转化成Embedding下标，从Embedding矩阵中取得特征的embedding后，通过与其他特征做pooling或者 attention后，进入CTR模型。

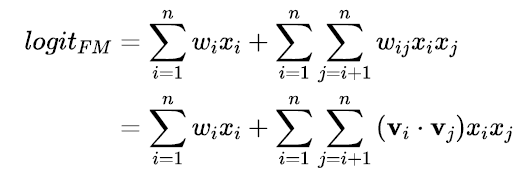
**Embedding**

***为什么说Embedding提升了推荐算法的扩展性？***

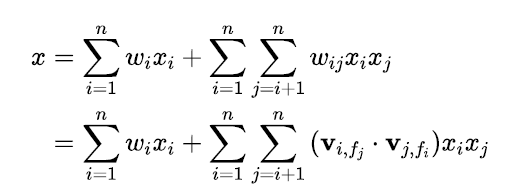
Embedding将推荐算法从“精确查找”变成“模糊查找”，让模型能够“举一反三”。以“科学”与“科技”两个词为例，查找包含“科学”这个词的文章，如果使用不进行同义词扩展，无法召回包含“科技”这个词汇的文章；但使用embedding进行计算，因为“科学”和“科技”这两个词的语义相关度较高，也能召回包含“科技”这个词的文章，这样就扩展了召回的规模。

***FFM针对FM的改进在哪里？***

FM中，每个feature与其他feature交叉式，使用的是同一个embedding，即embedding是共享的。这可能导致相互干扰问题，例如，模型调整Vi 为了把Wij学好，但是可能对另外一对特征组合的稀疏造成负面影响。参数量是n (featrue总数) \* k (Embedding size)



FFM的改进在于，每个特征和不同特征交叉时，根据对方特征所属的Field，使用不同的Embedding。如下公式所示，fj代表特征j所属的Field。Vi,fj 表示第i个feature针对第j个feature所属Field的embedding。说明第i个feature在和属于不同Field的feature交叉式，使用了不同的Embedding。参数量是n (featrue总数) \* f(feature所属filed数量) \* k (Embedding size) 。



***简述阿里Co-Action Network的基本思想？***

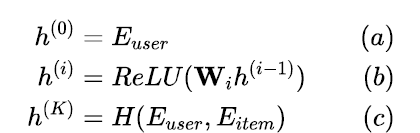
目标是：希望像FFM一样，让每个特征和其他特征交叉时，使用不同的embedding；避免引入FFM那么多的参数。

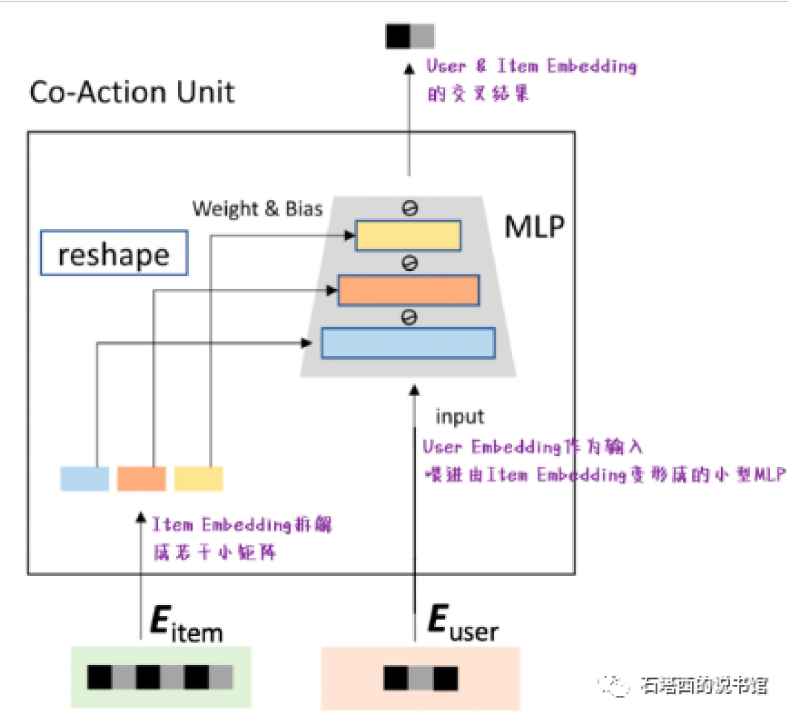
方法：1. 将物料embedding拆成K段，把这K段组织成一个MLP，使用ReLU作为每层的激活函数，将用户特征embedding喂入这个由物料特征embedding变化出来的MLP，MLP的输出就是这两个交叉的结果。

2. 使用ReLU作为激活函数，使得每一层的输出，有一些位置为0，也就是Wi中有一些神经元没有发挥作用；也就是物料特征的embedding和不同的用户特征交叉时，向量中的不同区域发挥作用；也就是同一个物料特征和不同的用户特征交叉时，使用了不同的embedding。

3.没有像FFM那样引入过多的参数，





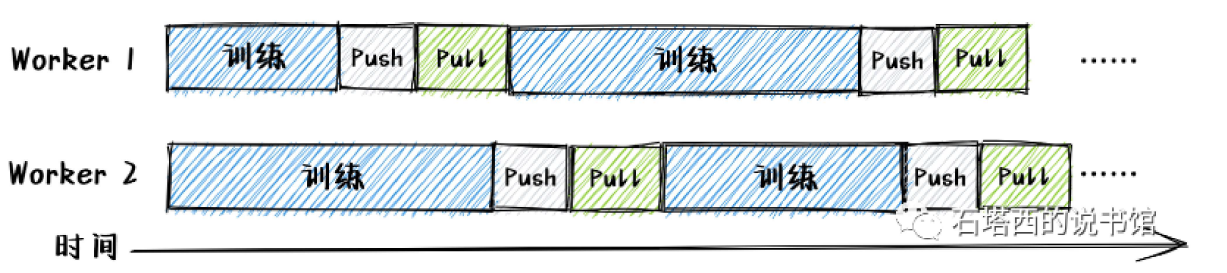


***简述Parameter Server是如何应对推荐系统“高维稀疏”的数据环境的？***

因为特征空间“高维稀疏”的特点，1. Worker节点先汇总当前batch中非零特征，再向Server提出pull参数请求，Server也只回复请求的这些特征（一介特征，Embedding）减少了数据网络传输。2. Server聚合汇总所有Worker发来的梯度，更新相应的参数。

***什么是异步并发(ASP) 中的梯度失效问题？即使如此， 为什么在推荐系统中仍然常用？***

ASP指的是每个Worker推送自己的梯度后，直接开始下一轮Batch训练，不用等待其他Worker。



由于缺乏同步机制，ASP存在“梯度失效”问题（Stale Gradient）,影响收敛速度。例如：

1. Server端有一个参数A\_0, Work\_0和Work\_1同时请求了这个参数，得到后，开始训练。
2. Work\_0训练的快，结束后向Server推送梯度g\_1.
3. Server接受到g\_1后，使用SGD，更新 A\_1 = A\_0 -lr \* g\_1.
4. Work\_1 训练结束，推送g\_2.
5. Server接收到g\_2后，如果使用SGD，这样更新 A\_2 = A\_1llr \* g\_2，就有可能损害收敛。因为g\_2是基于A\_0得到的梯度，但Server只能根据A\_1进行更新，g\_2其实已经失效了。

因为推荐系统的特征高维稀疏，在一轮迭代中，每个Worker分配的训练数据中，非零且重叠的很少。多个Worker同时跟新同一个特征的参数（一阶权重和Embedding）的可能性非常小，在Server端产生冲突的可能性也非常小，所以ASP在推荐系统中十分常用。