

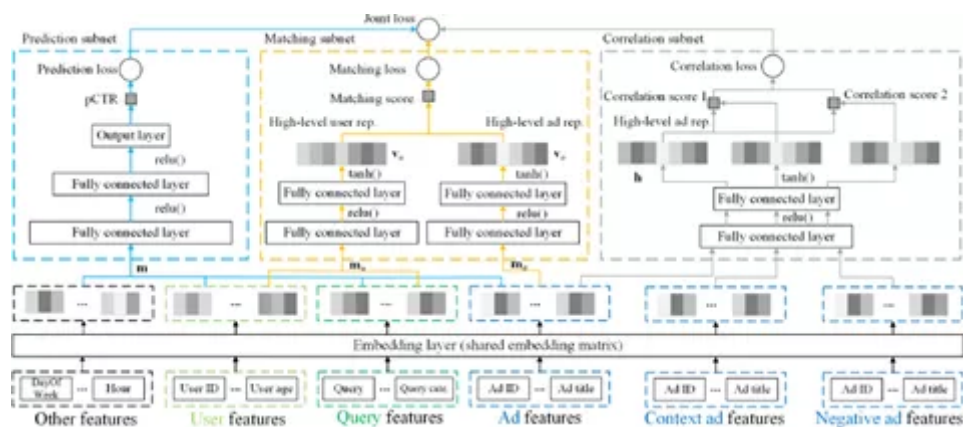
# 推荐模型之DeepMCP模型

原创 hellobill 比尔的新世界 8月23日

## 背景

DeepMCP模型是阿里19年提出的一个广告点击率预估，不同于传统的CTR预估模型刻画特征-CTR之间的联系，该模型采用多任务学习的方式进行联合训练进一步挖掘用户-广告、广告-广告之间的信息从而使得系统对于特征-CTR之间的联系刻画得更加准确。

## DeepMCP模型结构



DeepMCP包括三部分：prediction subnet、matching subnet和correlation subnet。这些子网络模型对特征-点击率，用户-广告，广告-广告关系分别建模。可以看出，这是一个多目标学习框架，所有子网在训练时都是激活的，预估的时候使用部分子网，比如点击率预估使用prediction subnet，而粗排和召回可以使用matching subnet。

输入特征分为四组，用户特征（如用户id，年龄），查询特征（用户历史行为），广告特征（如广告id），其他特征（时间等）。其中，prediction subnet使用所有的四组特征，matching subnet使用用户，查询和广告特征，corelation subnet 仅使用广告序列特征，所有这些子网共享embedding。

## prediction subnet

prediction subnet就是一个典型的点击率预估模型，可以替换为Wide&Deep、DeepFM等点击率模型。特征首先通过Embedding层转换为对应的Embedding，然后将特征进行横向拼接输入到DNN中，最后在输出层通过sigmoid函数来生成预测的点击率，损失函数使用logloss。

$$\text{loss}_p = -\frac{1}{|\mathbb{Y}|} \sum_{y \in \mathbb{Y}} [y \log \hat{y} + (1 - y) \log(1 - \hat{y})]$$

## matching subnet

matching subnet对用户-广告关系建模,学习用户与广告是否匹配。这个就是召回阶段使用的双塔DSSM模型,核心思想就是分别学习得到两个Embedding向量,分别是用户侧Embedding向量和广告侧Embedding向量,然后利用这两个Embedding向量计算得到一个matching score,具体matching score计算公式如下:

$$s(\mathbf{v}_u, \mathbf{v}_a) = \frac{1}{1 + \exp(-\mathbf{v}_u^T \mathbf{v}_a)}$$

损失函数和predictionsubnet一样是logloss。全连接网络的最后一层的激活函数是tanh而不是relu,主要是因为采用relu的话,最后一层的数据会包括很多零值,这会使得计算得到的matching score更趋近于零。

## corelation subnet

corelationsubnet 对广告-广告关系建模,基于对于一个用户的广告点击序列来说,在一定的时间窗口内部,广告之间存在一定的相关性的。这里用到了skip-gram的思路,对于一个广告点击序列来说,最优化的目标是最大化对数似然函数:

$$ll = \frac{1}{L} \sum_{i=1}^L \sum_{\substack{1 \leq i+j \leq L, j \neq 0 \\ -C \leq j \leq C}} \log p(a_{i+j} | a_i)$$

L是序列中广告的数量,C是上下文窗口的大小。对于上式中概率p的建模存在很多种方式,如softmax、hierarchical softmax、降采样等,文章采用的是降采样的方式,概率p如下式所示:

$$p(a_{i+j} | a_i) = \sigma(\mathbf{h}_{a_{i+j}}^T \mathbf{h}_{a_i}) \prod_{q=1}^Q \sigma(-\mathbf{h}_{a_q}^T \mathbf{h}_{a_i})$$

在此基础上correlationsubnet的损失函数被定义为最小化平均对数似然函数的负值，即：

$$\text{loss}_c = \frac{1}{L} \sum_{i=1}^L \sum_{\substack{1 \leq i+j \leq L, j \neq 0 \\ -C \leq j \leq C}} \left[ -\log \left[ \sigma(\mathbf{h}_{a_{i+j}}^T \mathbf{h}_{a_i}) \right] \right. \\ \left. - \sum_{q=1}^Q \log \left[ \sigma(-\mathbf{h}_{a_q}^T \mathbf{h}_{a_i}) \right] \right].$$

DeepMCP最终的损失函数为三个子模块损失函数加权和：

$$\text{loss} = \text{loss}_p + \alpha \text{loss}_m + \beta \text{loss}_c$$

其中 $\alpha$ 和 $\beta$ 是调节因子，用以调节不同子网络的权重。

## 线上Serving

从下表可以看出，单独使用DeepMP效果比base就要好很多，再加入Correlation 网络，效果提升有限，而数据和模型的复杂度都提升了很多。所以阿里最终线上采用的是DeepMP，在预估准确性和模型复杂度之间达到了最好的折衷。

	Avito		Company	
Algorithm	AUC	Logloss	AUC	Logloss
LR	0.7556	0.05918	0.7404	0.2404
FM	0.7802	0.06094	0.7557	0.2365
DNN	0.7816	0.05655	0.7579	0.2360
PNN	0.7817	0.05634	0.7593	0.2357
Wide&Deep	0.7817	0.05595	0.7594	0.2355
DeepFM	0.7819	0.05611	0.7592	0.2358
DeepCP	0.7844	0.05546	0.7610	0.2354
DeepMP	0.7917	0.05526	0.7663	0.2345
DeepMCP	<b>0.7927</b>	<b>0.05518</b>	<b>0.7674</b>	<b>0.2341</b>

多目标学习共享embedding具有迁移学习的性质，prediction subnet和matching subnet两个任务之间相关性较高，是可以相互促进提升彼此单独模型的效果的。prediction subnet的预测能力能够有助于matching subnet提升表达能力，而matching subnet的表达又有助于prediction subnet提升预测能力，所以该模型除了在排序阶段使用（包括粗排和精排）prediction subnet，还可以在召回阶段使用matching subnet。