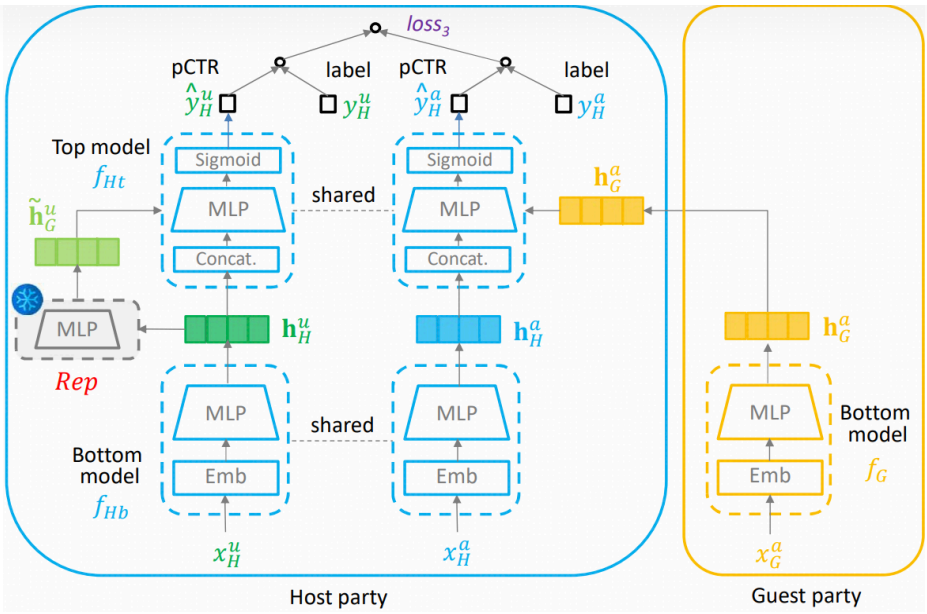
知乎



联邦学习：挖掘垮平台用户ID不匹配数据提升CTR预测

SmartMindAI
专注搜索、广告、推荐、大模型和人工智能最新技术，欢迎关注我
已关注

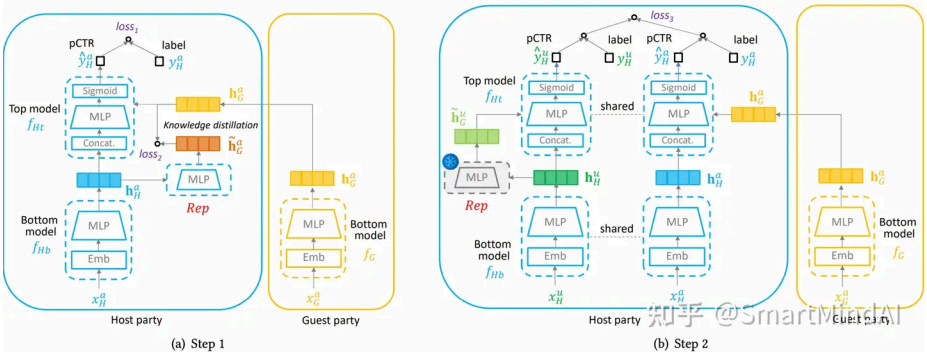
2 人赞同了该文章

Introduction

CTR 预测是在线广告平台的核心任务之一。目前大多数方法都使用来自平台本身的收集数据进行CTR预测，随着用户行为在其他平台上丰富起来，进一步利用这些补充信息以提高CTR预测性能是非常有用的。然而，由于用户隐私的顾虑，不同平台的数据无法集中上传到服务器进行模型训练。

然而，传统的联邦学习（VFL）方法仅使用具有共同id的对齐数据，这严重限制了其应用范围。在本文中，我们提出FedUD，它能够利用未对齐数据进行联邦CTR预测，而不仅仅是利用对齐数据。FedUD包含以下两个阶段。首先，FedUD像传统VFL一样利用跨平台的对齐数据，并增加了一个知识蒸馏模块。其次，FedUD应用学习到的知识来丰富主机平台的未对齐数据的表示。

Model Design



Step 1: Federated Learning using Aligned Data with Knowledge Distillation

我们通过图（a）来说明这一步骤。 为了保护隐私，每一方都有自己的本地模型。主方只有本地底层模型f_G，其目标是根据主方的本地数据提取高级表示。本地模型f_G包含一个嵌入层和一个包含多个全连接层的多层感知器（MLP），这些全连接层使用ReLU激活函数。 我们将客方数据的高级表示表示为：

$$\mathbf{h}_G^a = f_G(x_G^a) = MLP_G\left(Emb_G(x_G^a)\right)$$

知乎

$$\mathbf{h}_H^a = f_{Hb}(x_H^a) = MLP_{Hb}\left(Emb_H(x_H^a)\right)$$

主方拥有额外的顶级模型f_{Ht}，它以输入的h_H^a和h_G^a作为基础，输出点击率预测的logit[+]为：

$$z_H^a = f_{Ht}(\mathbf{h}_H^a, \mathbf{h}_G^a) = MLP_{Ht}\left(Concat(\mathbf{h}_H^a, \mathbf{h}_G^a)\right)$$

主方的顶级模型f_{Ht}，包含了拼接层和多层感知机[+]，由给出的模型f_{Ht}计算的预测点击率为：

$$\hat{y}_H^a = Sigmoid(z_H^a)$$

然而，利用主方的非对齐数据并不简单，因为这些数据在客方没有对应的特征。为解决这一问题，我们的目标是学习一个表示转换网络Rep，其输入是主方的高层表示$\mathbf{h}_H^a$，输出是

$$\widetilde{\mathbf{h}}_G^a = Rep(\mathbf{h}_H^a)$$

在某种度量下模仿客方的高层表示$\mathbf{h}_G^a$。这样$\mathbf{h}_G^a$中的知识被浓缩到$\widetilde{\mathbf{h}}_G^a$中，并且这种知识指导了表示转换网络的学习。 在FedUD的第一阶段，我们有两个目标：1）学习准确的高层表示和2）通过知识提炼学习一个准确的表示转换网络。为了实现第一个目标，我们优化了预测损失如下：

$$loss_1 = \frac{1}{|Y_H^a|}\sum_{y_H^a \in Y_H^a}\left[-y_H^a \log(\hat{y}_H^a) - (1-y_H^a)\log(1-\hat{y}_H^a)\right]$$

为了达成第二个目标，我们采用以下方法优化两个表示之间的均方误差[+]（MSE）损失：

$$loss_2 = \frac{1}{|Y_H^a|}\sum_{y_H^a \in Y_H^a}\left\|\widetilde{\mathbf{h}}_G^a - \mathbf{h}_G^a\right\|^2 = \frac{1}{|Y_H^a|}\sum_{y_H^a \in Y_H^a}\left\|Rep(\mathbf{h}_H^a) - \mathbf{h}_G^a\right\|^2$$

在FedUD的第一步中，总损失被定义为仅应用于各参与方之间的对齐数据的$loss_1$和$loss_2$。$loss_1 + \alpha\, loss_2$在其中，α是可调节的平衡超参数。

Step 2: Federated Learning using Both Aligned and Unaligned Data

我们用图（b）来说明这一步。对于主方的非对齐数据x_H^u，我们只能获得主方的表示$\mathbf{h}_H^u$，而无法获得客方的表示。这使得传统的VFL方法无法利用未对齐数据。 相比之下，我们已经学习了一个表示转移网络Rep，在第一步中。尽管Rep是基于对齐数据学习的，但我们将其应用于非对齐数据，通过$\mathbf{h}_H^u$来推断$\widetilde{\mathbf{h}}_G^u$这样做，我们可以在模型训练中，既可以使用对齐数据，也可以使用非对齐数据。 对于对齐数据：$\mathcal{D}^a = \{x_H^a, y_H^a, x_G^a\}$

主方的高层表示由其局部底层模型给出，客方的高层表示由其局部底层模型给出：

$$\mathbf{h}_H^a = f_{Hb}(x_H^a),\ \mathbf{h}_G^a = f_G(x_G^a)$$

主方的顶级模型随后生成了预测的点击率$\hat{y}_H^a$：

$$z_H^a = f_{Ht}(\mathbf{h}_H^a, \mathbf{h}_G^a),\ \hat{y}_H^a = Sigmoid(z_H^a)$$

对于未对齐的数据：$\mathcal{D}^u = \{x_H^u, y_H^u\}$

主方的局部底层模型提供了其高阶表示，而客方的高阶表示则由主方的代表转移网络Rep提供：

$$\mathbf{h}_H^u = f_{Hb}(x_H^u),\ \widetilde{\mathbf{h}}_G^u = Rep(\mathbf{h}_H^u)$$

主方的顶级模特随后计算出预测的点击率$\hat{y}_H^u$是：

$$z_H^u = f_{Ht}(\mathbf{h}_H^u, \widetilde{\mathbf{h}}_G^u),\ \hat{y}_H^u = Sigmoid(z_H^u)$$

我们然后根据对齐和未对齐的数据优化预测损失，如下：

$$+\beta\frac{1}{|Y_H^u|}\sum_{y_H^u\in Y_H^u}\left[-y_H^u\log(\hat{y}_H^u)-(1-y_H^u)\log(1-\hat{y}_H^u)\right],$$

其中Y_H^a和Y_H^u分别等于$\{y_H^a\}$和$\{y_H^u\}$，而β是一个可调平衡超参数。 当我们优化$loss_3$时，我们加载了表示迁移网络Rep的参数，这些参数在学习过程中被学习到，并保持它们处于冻结状态。

Privacy

FedUD 保留了传统VFL的所有隐私属性，因为客方仅向主方发送对齐数据的高层表示，而主方仅发送相应的部分梯度到客方。两方之间不进行原始数据的交流。主方的非对齐数据仅被主方自身使用。

Experiments

Datasets

（1）Avazu 数据集。我们将它分为训练、验证和测试数据集，每个数据集包含8天的数据。主方和客方分别有10和12个特征维度。样本键为设备ID的哈希值。 （2）商业数据集。我们将它分为训练、验证和测试数据集，每个数据集包含7天的数据。主方是一个新闻推送广告平台。客方是一个媒体平台。主方的特征维度有22个，客方的有12个。样本键为设备ID的哈希值。

Table 1: Statistics of experimental datasets.

Dataset	# Fields	# Train	# Val	# Test	# Show	# Click
Avazu	22	32.4M	3.83M	4.22M	40.43M	6.86M
Industrial	34	439.3M	62.5M	61.7M	563.5M	209.8M

Experimental Results

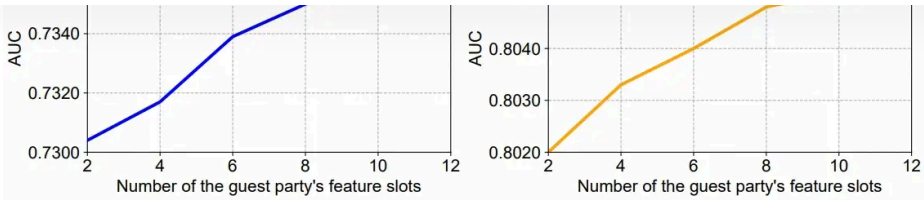
Effectiveness

	Avazu						Industrial					
	overall		aligned		unaligned		overall		aligned		unaligned	
	AUC↑	LogLoss↓	AUC↑	LogLoss↓	AUC↑	LogLoss↓	AUC↑	LogLoss↓	AUC↑	LogLoss↓	AUC↑	LogLoss↓
DNN	0.7186	0.4140	0.7203	0.4204	0.6997	0.3695	0.7996	0.5066	0.8032	0.4977	0.7909	0.5253
Wide&Deep	0.7178	0.4154	0.7191	0.4211	0.6994	0.3696	0.7997	0.5066	0.8034	0.4977	0.7909	0.5253
DeepFM	0.7185	0.4142	0.7202	0.4204	0.6995	0.3696	0.7997	0.5067	0.8033	0.4977	0.7910	0.5253
AutoInt	0.7196	0.4111	0.7212	0.4187	0.7003	0.3694	0.7997	0.5066	0.8033	0.4977	0.7912	0.5252
FedSplitNN	0.7283	0.4074	0.7330	0.4120	0.6932	0.3809	0.7990	0.5081	0.8102	0.4911	0.7761	0.5437
FedCTR	0.7294	0.4068	0.7346	0.4112	0.6935	0.3807	0.7998	0.5068	0.8099	0.4918	0.7786	0.5405
SS-VFL	0.7298	0.4065	0.7351	0.4109	0.6948	0.3797	0.8001	0.5063	0.8106	0.4908	0.7778	0.5412
FedHSSL	0.7302	0.4063	0.7358	0.4103	0.6941	0.3805	0.7999	0.5065	0.8105	0.4907	0.7771	0.5419
FedUD	**0.7355***	**0.4037***	**0.7370***	**0.4093***	**0.7083***	**0.3652***	**0.8057***	**0.5026***	**0.8121***	**0.4899***	**0.7952***	**0.5224***

在表中，"整体"，"对齐"，和"未对齐"分别表示在所有测试数据、仅对齐测试数据（包含客方的特征）和仅未对齐测试数据（不包含客方的特征）上计算AUC/LogLoss。观察到在对齐测试数据上，大多数联邦方法的表现比本地方法更好，这是因为包含了客方的特征。然而，大多数联邦方法在未对齐测试数据上的表现不如本地方法，这是因为缺少了客方的特征。本地方法完全不使用客方的特征。尽管SS-VFL和FedHSSL也利用了未对齐的数据，但它们仅用于自我监督学习，因此它们在未对齐测试数据上的表现也不好。相反，FedUD明确地从对齐数据向未对齐数据转移知识，并在两个数据集上都表现出最佳性能。 观察到FedUD在对齐数据上相对于DNN（一种本地方法）的AUC提升高于在未对齐数据上的提升。这是因为对齐数据包含真实的客方特征，而未对齐数据包含推断的客方表示，这些表示更不准确。

Effect of the Guest Party's Feature Slots

知乎

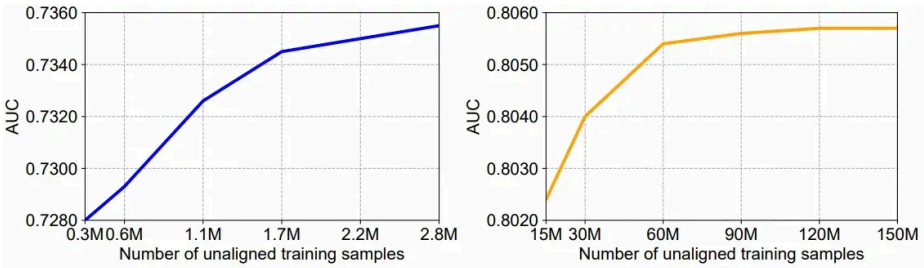


(a) Avazu (b) Industrial

Figure 2: Overall test AUC vs. the number of the guest party's feature slots. (a) Avazu dataset. (b) Industrial dataset.

图展示了主客体的特征槽数量与整体AUC值[+]的关系。对齐和未对齐测试数据的AUC趋势相近。观察到，更多客体特征槽位通常提供有用的额外信息，并提升性能。但是，增加更多的特征槽位也可能包含噪音，并可能导致性能平缓甚至轻微下滑。

Effect of the Host Party's Unaligned Samples



(a) Avazu (b) Industrial

Figure 3: Overall test AUC vs. the number of unaligned training samples. (a) Avazu dataset. (b) Industrial dataset.

在这个实验中，我们使用所有对齐样本进行训练，但使用不同数量的未对齐样本。图展示了总体AUC值与宿主派对的未对齐样本数量的关系。x轴最右边的点表示训练集中未对齐样本的最大数量。观察到，当数量达到一定水平时，更多的未对齐样本一般会提升预测性能指标。但在达到一定数量后，提升的效果不再显著。

原文《FedUD: Exploiting Unaligned Data for Cross-Platform Federated Click-Through Rate Prediction》

编辑于 2024-09-04 11:11 · IP 属地北京

联邦学习 ctr 工业级推荐系统

理性发言，友善互动

还没有评论，发表第一个评论吧