

腾讯2024年STEM新突破：重构MMOE网络结构，实现多目标学习效率飞跃



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

137 人赞同了该文章

Introduction

近期，多任务学习在推荐系统中成为焦点，优化诸如 喜欢、分享、完成 等多重目标。尽管MTR能利用多任务间知识，但其可能存在的负迁移问题，表现为多任务模型并非总优于单任务模型。为此，MMoE和PLE通过加入任务专属性门控网络、专家系统⁺，以及采用梯度剪枝和任务意识优化器等策略，来减轻不同任务间的潜在冲突，以改善多任务学习的效果。

当前的多任务学习方法在处理推荐系统⁺时，虽然能整合多种任务，但忽略了样本内部的差异，导致负迁移问题。为深入理解这一现象，我们在TikTok⁺的公开测试集⁺上，依据任务'Finish'和'Like'反馈的比例，将样本划分为三个子集进行研究。

- 我们聚焦于样本层面，探究多任务推荐模型中的负迁移⁺现象。在对TikTok的公开测试集进行分析时，我们发现那些反馈在'Finish'和'Like'任务上相当的样本中，负迁移现象普遍存在。
- 我们创新地提出STEM⁺，一个融合了共享与任务特定嵌入的多任务推荐新范式。为应对负迁移问题，我们构建了STEM-Net模型，它采用全向前向任务自适应后向门控网络，以强化任务特定信息的学习和跨任务知识的有效转移。
- 我们进行了实证研究⁺，包括在三个多任务推荐数据集上的详尽测试，以验证STEM-Net的有效性。结果证实，STEM-Net显著优于基线模型MMoE，体现在提升的GMV⁺（广告收入）。此外，我们的模型已在腾讯的在线广告平台实现了实际应用并取得了成功。

Delve into Negative Transfer in MTL Recommenders

我们定义了三个样本集：A-0 的压倒性优势子集 \mathcal{D}_{A-0} 和 B-0 的压倒性优势子集，分别包含任务A或B显著优势的积极反馈样本，以及对子集 \mathcal{D}_{Comp} ，它包含具有相似积极反馈的样本。我们量化单个任务模型对每个样本在各任务上的积极预测，通过比较任务间预测的差异来区分样本类别。我们运用TikTok数据集进行实证研究，该数据集涵盖了两个任务，以此来验证我们的观察。我们使用了A-和B-压倒性优势子集以及比较子集 \mathcal{D}_{Comp} ，通过分析单一任务模型对样本在各任务的积极反馈预测，来量化并分析任务间反馈差距，以识别可能存在的负迁移现象。

Method

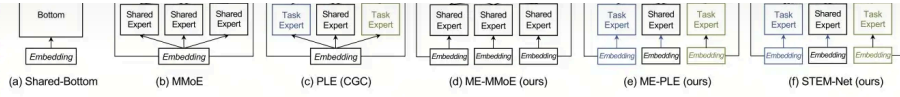


Figure 2: Comparison between representative MTL models and our proposed STEM-Net. Dot lines indicate connections with stop-gradient operation.

The STEM Paradigm

在推荐系统中，每个样本 \mathbf{x} 由 M 个特征 $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_M$ 组成，每个特征有特定的字段。传统方法采用共享嵌入模型⁺，所有特征共享一个 $N \times K$ 的嵌入表 E^S 。然而，我们提出STEM范式，引入 T 个任务特有的嵌入表 $\{E^1, E^2, \dots, E^T\}$ ，每个对应任务 t 。每个特征 \mathbf{x}_i 通过计算得到任务特定嵌入 \mathbf{v}_i^t 和共享嵌入 \mathbf{v}_i^S ：

$$\mathbf{v}_i^t = E^t \times \mathbf{x}_i \quad (\text{任务特定}) \quad \text{和} \quad \mathbf{v}_i^S = E^S \times \mathbf{x}_i \quad (\text{共享})$$

其中 E^t 是任务 t 的特定嵌入矩阵，通过与特征 \mathbf{x}_i 相乘生成任务相关特征表示。这种方法允许任务之间更好地处理和避免负迁移问题。

$$\begin{aligned} \mathbf{v}_i^t &= \text{Lookup}(\mathbf{x}_i, E^t), \\ \mathbf{v}_i^S &= \text{Lookup}(\mathbf{x}_i, E^S). \end{aligned}$$

在STEM范式中，我们结合共享-任务嵌入理念，通过计算每个特征的特定和共享嵌入。具体来说，对于特征 \mathbf{x}_i ，我们有：

$$\mathbf{v}_i^t = E^t \times \mathbf{x}_i \quad (\text{任务特有}) \quad \text{和} \quad \mathbf{v}_i^S = E^S \times \mathbf{x}_i \quad (\text{共享})$$

接着，我们将所有任务特定嵌入与共享嵌入串联起来，形成一个综合特征向量 \mathbf{h}_i ：

$$\mathbf{h}_i = \text{concat}(\mathbf{v}_i^1, \mathbf{v}_i^2, \dots, \mathbf{v}_i^T, \mathbf{v}_i^S)$$

这样 \mathbf{h}_i 包含了所有任务的信息，有助于在推荐过程中更有效地利用不同任务间的关联性和互补性。

$$\begin{aligned} \mathbf{h}_0^t &= [\mathbf{v}_1^t, \dots, \mathbf{v}_i^t, \dots, \mathbf{v}_M^t], \\ \mathbf{h}_0^S &= [\mathbf{v}_1^S, \dots, \mathbf{v}_i^S, \dots, \mathbf{v}_M^S], \end{aligned}$$

在这个设置中，我们以任务特定嵌入的串联版本 \mathbf{h}_0^t 和共享嵌入 \mathbf{h}_0^S 作为输入，分发给专属于每个任务的专家和一个共享专家。具体操作如下：

$$\mathbf{e}_0^t = \text{Expert}_t(\mathbf{h}_0^t) \quad \text{和} \quad \mathbf{e}_0^S = \text{Expert}_S(\mathbf{h}_0^S)$$

其中 Expert_t 由任务 t 的专业知识驱动，处理专属信息以产生专家特有的输出；而 Expert_S 利用共享的通用知识来辅助决策，生成共享专家的响应。这种专家网络的组合方式提供了全面的上下文理解，从而在多任务环境下能更精准地进行推荐。

The STEM-Net Model

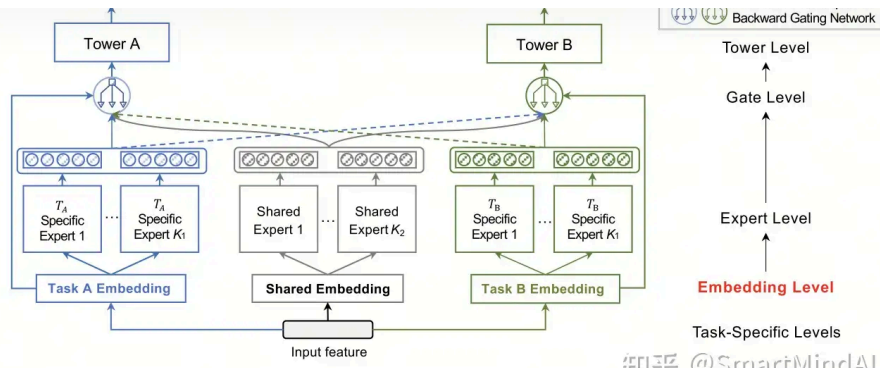


Figure 3: Overview of STEM-Net.

在STEM范式中，我们构建了STEM-Net，利用共享专家和任务特定专家的协同。共享专家处理通用特征和领域知识，而每个任务的专家则专注于任务特定的学习和优化。全向前向任务特定后向门控网络则确保信息流动，允许任务特定嵌入在共享基础上学习，并适时向其他任务传递有益知识。这样的设计促进了STEM-Net在多任务环境中的有效性和准确性，实现了更高效、精确的推荐。

Shared & Task-Specific Experts

共享专家 $(x) = \text{MLP}_{\text{shared}}(h_0^S)$ 和 任务 $t(x) = \text{MLP}_{\text{task}_t}(h_0^t)$

这里 $\text{MLP}_{\text{shared}}$ 和 $\text{MLP}_{\text{task}_t}$ 代表各自专家的网络结构，它们分别训练，既能学习通用知识，又能针对特定任务进行定制。这种设计保证STEM-Net在多任务环境下既能保持学习的灵活性，又能确保准确性。

$$h_i^t = \text{MLP}_i^t(h_0^t), \forall i = 1, \dots, K_1,$$

$$h_j^S = \text{MLP}_j^S(h_0^S), \forall j = 1, \dots, K_2,$$

接下来，我们结合专家的输出：

$$h_i^t = \text{Expert}_t(i) \quad \text{and} \quad h_j^S = \text{Expert}_S(j)$$

这些专家的个体贡献通过合并形成一个综合的上下文表示，STEM-Net据此提供多任务推荐。通过整合不同专家的专业知识，系统能够更好地理解和满足各类任务的要求，从而提升推荐的精确度和多样性。

All Forward Task-specific Backward Gating Network

在STEM-Net中，门控机制通过全向前向任务特定后向门控网络（All Forward Task-Specific Backward gating network）整合任务塔专家的输出，旨在最大化知识的跨任务转移。它连接每个任务塔到所有专家，但在计算非目标任务的梯度时执行止步操作，以避免过度干扰。

任务 t 的门控信号 g^t 由专家输出集合计算得出，包括任务 t 的特定专家 $\{h_i^t\}$ 和共享专家 $\{h_j^S\}$ 的输出。这个过程通过门控函数 'Gating' 实现，它动态调整每个任务的权重，确保既能从共享知识中学习，又能保留任务特有的信息，从而优化推荐的准确性和多样性。

$$o^t = \sum_i^{K_1} g_i^{t \rightarrow t} h_i^t + \sum_i^{K_2} g_i^{S \rightarrow t} h_i^S$$

$$+ \sum_{t' \in T, t' \neq t} \sum_i^{K_1} g_i^{t' \rightarrow t} \text{SG}(h_i^{t'}),$$

在这个表达中 $\text{SG}(\cdot)$ 表示停止梯度操作，用来阻止参数间的直接关联。针对任务 t $g^{t \rightarrow t}$ 是针对任务 t 内部专家权重 $g^{t \rightarrow t}$ 代表任务 t 从其他任务 t' 专家获得的影响力，而 $g^{S \rightarrow t}$ 则是共享专家对任务 t 的影响权重。

这些权重通过组合所有专家和任务塔的嵌入 ($\{h_i^t\} \{h_j^S\}$) 计算得出，然后通过softmax函数进行归一化处理。softmax函数确保权重分布均匀，加权和为1，这样STEM-Net能动态地依据这些

$$\text{Softmax}(W_g^t(h_0^t + h_0^s)),$$

在这个结构中 W_g^t 是维度为

$$d \times (K_1 \times T + K_2)$$

的权重矩阵⁺，其中 K_1 和 K_2 分别对应任务内和共享专家的权重维度。 W_g^t 的作用是将任务相关的注意力权重 $(g^{t \rightarrow t} g^{t' \rightarrow t} g^{s \rightarrow t})$ 与相应的专家嵌入相结合，以更新任务 t 的特定嵌入。通过这种矩阵乘法⁺，STEM-Net能够根据任务间的关系和共享知识动态调整每个任务的计算，从而更好地整合多元信息，提升推荐系统的精确性和多样性。

Towers and Loss Function

最后，STEM-Net通过独立的任务塔对每个任务进行处理，其输出由专门的输出塔Output Tower _{t} 计算得出，结合所有任务塔的专家输出和任务专属的门控信号 g^t ：

$$\hat{x}^t = \text{Output Tower}_t(\{h_i^t\}, g^t)$$

这个过程确保了个性化响应，同时门控机制⁺确保了对所有专家信息的有效整合，从而显著提升了推荐系统的总体效能。

$$\hat{y}_t = \sigma(\text{MLP}^t(o^t))$$

目标函数⁺定义为 L ，它包括三个部分：针对每个任务 t 的二元交叉熵⁺损失 L_t ，用以衡量预测输出 \hat{x}^t 与实际标签 y^t 的差异；正则化项，通过 λ 惩罚不同任务间专家权重的差异，用

$$\|sg(g^{t \rightarrow t}) - sg(g^{t' \rightarrow t})\|_2$$

度量；以及另一个正则化项，用 γ 约束共享专家权重 $sg(g^{s \rightarrow t})$

与全局共享权重 $sg(g^s)$ 的差异。停止梯度操作确保了权重的稳定更新。通过优化这个损失函数⁺，STEM-Net既能优化每个任务的学习，又能控制知识跨任务传播，从而提升推荐系统的整体表现。

$$\mathcal{L} = - \sum_t^T y_t \log(\hat{y}_t) + (1 - y_t) \log(1 - \hat{y}_t),$$

STEM-Net的目标函数 L 由任务 t 的二元交叉熵损失 $L_t(\hat{x}^t, y^t)$ 组成，加入两个正则化项，分别针对任务间专家权重(λ 控制)和共享专家权重(γ 控制)的差异。这里 $sg(\cdot)$ 确保了权重的稳定更新，避免梯度流动。该公式表明，模型通过这种方式优化每个任务的学习，同时通过门控机制保持知识在不同任务间的平衡，从而优化整体推荐系统的效能。 y_t 代表任务 t 的实际标签。

Comparison of Gating Networks

MMoE和PLE在门控策略上有所不同。MMoE使用全前全后的门控网络，每个任务塔会接收所有专家的输出并反馈梯度给所有专家，这意味着它可能难以直接学习到针对每个任务的专用嵌入。若采用多嵌入MMoE (ME-MMoE)，即使为每个专家分配了独立嵌入，这种设计也限制了任务间的直接知识传递。相反，PLE采取任务特定的前向任务特定的后向门控网络，每个塔仅与共享专家和自身专家互动，不考虑其他任务的专家。这使得PLE不能从其他任务中完全转移知识，即使使用多嵌入PLE (ME-PLE)，每个专家组（即独立嵌入）独立于其他任务。

Performance Evaluation

Public Datasets.

Table 1: Statistics of processed datasets.

Dataset	#User	#Items	#Samples	#Fields	#Tasks	Positive Ratio (%)
TikTok	560K	1800K	223.4M/24.8M/27.6M	8	2	28.31/1.60
QK-Video	970K	760K	95.9M/12.0M/12.5M	16	2	24.01/2.03
KuaiRand1K	1K	189K	10.9M/0.39M/0.42M	32	8	37.76/1.54/0.10/0.26/0.08/0.10/26.17/1.78

在实验设置中，我们构建了多任务模型对比基准，包括单任务模型（Single-Task）、Shared-Bottom、OMoE、MMoE和PLE。其中，Single-Task相当于只有一个塔的Shared-Bottom，仅使用MLPs。为更准确评估，我们在TikTok和QK-Video数据集上增加了ESMM和AITM两个额外基线。

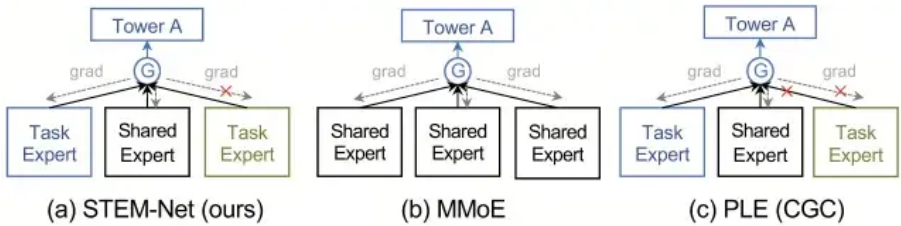


Figure 4: Comparison of gating networks of MMoE (All Forward All Backward), PLE (Task-specific Forward Task-specific Backward) and our STEM-Net (All Forward Task-specific Backward).

同时，我们提供了增强版的MMoE（ME-MMoE，图(d)）和PLE（ME-PLE，图(e)），分别研究仅增加嵌入对性能的影响。ME-MMoE通过为每个专家分配独立嵌入，遵循STEM-Net的架构，而ME-PLE则将任务特性和共享专家的嵌入分开。这两者都采用了相同的门控策略，即All Forward All Backward和Task-specific Forward Task-specific Backward，以确保公平对比。通过这些对比，我们将深入理解STEM-Net在多任务学习中的优势。

Hyper-Parameter Settings.

我们在Pytorch平台上实现了所有模型，利用Adam+作为优化器。我们设置了三个不同的学习率： $1e^{-3}$ $5e^{-4}$ 和 $1e^{-4}$ 。批大小为4096，并对嵌入施加 l_2 正则化，参数为 $1e^{-6}$ 。嵌入维数设为16，每个专家/基础结构（Shared-Bottom）由包含512、512和512个隐藏单元的MLP组成。所有塔和门控网络由具有128和64个隐藏单元的MLP构成。任务特性和共享专家的数量探索范围为1,2,4,8。通过网格搜索+，我们寻找每种方法的最佳超参数+组合。

Overall Performance

Table 2: Overall performance on TikTok.

Model	Finish		Like		Average AUC ↑
	Logloss↓	AUC ↑	Logloss↓	AUC ↑	
Single-Task	0.5111	0.7505	0.0558	0.9058	0.8281
Shared-Bottom	0.5112	0.7504	0.0560	0.9022	0.8263
OMoE	0.5103	0.7516	0.0559	0.9029	0.8273
MMoE	0.5105	0.7511	0.0560	0.9018	0.8265
PLE	0.5105	0.7511	0.0560	0.9016	0.8264
ESMM	0.5111	0.7503	0.0564	0.9012	0.8258
AITM	0.5109	0.7506	0.0560	0.9026	0.8266
ME-MMoE	0.5114	0.7502	0.0557	0.9045	0.8274
ME-PLE	0.5120	0.7492	0.0560	0.9058	0.8275
STEM-Net	0.5104	0.7513	0.0553	0.9095	0.8304

Table 3: Overall performance on QK-Video.

Model	Click		Like		Average AUC ↑
	Logloss↓	AUC ↑	Logloss↓	AUC ↑	
Single-Task	0.2826	0.9234	0.0378	0.9400	0.9317
Shared-Bottom	0.2857	0.9235	0.0380	0.9389	0.9312
OMoE	0.2826	0.9238	0.0373	0.9394	0.9316
MMoE	0.2813	0.9238	0.0379	0.9401	0.9319
PLE	0.2832	0.9238	0.0375	0.9399	0.9318
ESMM	0.2847	0.9208	0.0378	0.9368	0.9288
AITM	0.2836	0.9237	0.0386	0.9398	0.9318
ME-MMoE	0.2815	0.9239	0.0375	0.9407	0.9323
ME-PLE	0.2818	0.9238	0.0374	0.9410	0.9324
STEM-Net	0.2816	0.9237	0.0381	0.9426	0.9331

Table 4: Overall performance on KuaiRand1K.

Model	Task A	Task B	Task C	Task D	Task E	Task F	Task G	Task H	Avg. AUC	MTL Gain
Single-Task	0.7534	0.9293	0.8294	0.8943	0.8572	0.8821	0.7650	0.8358	0.8433	-
Shared-Bottom	0.7535	0.9261	0.8162	0.8881	0.8228	0.7820	0.7642	0.8340	0.8234	-0.0199
OMoE	0.7549	0.9273	0.8404	0.8923	0.8352	0.8750	0.7655	0.8349	0.8407	-0.0026
MMoE	0.7541	0.9278	0.8268	0.8901	0.8591	0.8908	0.7647	0.8360	0.8437	+0.0003
PLE	0.7537	0.9290	0.8362	0.8885	0.8449	0.8940	0.7643	0.8374	0.8435	+0.0002
ME-MMoE	0.7555	0.9288	0.8310	0.8912	0.8500	0.8668	0.7658	0.8385	0.8410	-0.0024
ME-PLE	0.7536	0.9294	0.8353	0.8970	0.8521	0.8871	0.7637	0.8351	0.8415	-0.0024
STEM-Net	0.7523	0.9282	0.8420	0.8910	0.8635	0.9070	0.7637	0.8359	0.8480	+0.0047

Which feature fields should be task-specific?

在TikTok数据集上，为了学习跨任务的用户偏好多样性，你们设计了STEM-Net的变体(STEM-Net-**F**)，这个变体专注于特定的特征字段**F**来定制任务特定的嵌入。

lected feature fields deploy task-specific.

Variants	Tiktok		KuaiRand1K	
	AUC	#Param	AUC	#Param
STEM-Net- \emptyset	0.8261	1.00x	0.8382	1.00x
STEM-Net-(user id, item id)	0.8302	1.95x	0.8448	1.36x
STEM-Net-(user side)	0.8301	1.23x	0.8437	1.00x
STEM-Net-(item side)	0.8260	2.62x	0.8193	2.05x
STEM-Net-(all features)	0.8304	2.85x	0.8480	2.06x

Effectiveness of the Proposed Gating Network

在STEM-Net模型中，你们通过探究All Forward Task-specific Backward gates的重要性，强调了stop gradients（SG）操作对于学习任务特定嵌入的必要性。实验中，对输入 $h_0^t + h_0^s$ 进行了三种变体的替换，分别是只用 h_0^t 、只用 h_0^s ，以及保留原输入。

Table 6: The effect of input and stop gradient to gating network.

Input	SG	Finish		Like		Average AUC
		AUC	Logloss	AUC	Logloss	
h_0^t	✓	0.7509	0.5107	0.9077	0.0555	0.8293
	✗	0.7511	0.5106	0.9023	0.0560	0.8267
	Δ	+0.0001	-0.0001	-0.0054	+0.0005	-0.0026
h_0^s	✓	0.7510	0.5106	0.9089	0.0553	0.8300
	✗	0.7511	0.5105	0.9037	0.0558	0.8274
	Δ	+0.0001	0.0000	0.0052	+0.0005	-0.0026
$h_0^t + h_0^s$	✓	0.7513	0.5104	0.9095	0.0553	0.8304
	✗	0.7510	0.5106	0.9008	0.0562	0.8259
	Δ	-0.0002	+0.0003	-0.0080	+0.0009	-0.0045

结果显示：(a) 停止梯度操作显著提高了'Like'任务的平均AUC⁺，提升幅度在5-8个basis points，这表明SG操作有助于防止嵌入在任务间无意识地共享，避免了ME-MMoE的行为。(b) 当输入包含混合了任务特定和共享嵌入时，性能最好，平均提升约1个basis point。这表明，结合两种类型的信息能帮助门控网络更有效地理解 and 处理不同任务之间的相关性，避免了过度依赖或忽视任务间的联系。

Contradictory User Preference Analysis

在STEM-Net中，我们通过观察在反馈丰富的相似子集中的用户-商品对，假设存在任务间矛盾的偏好现象。例如，用户可能对任务A的商品偏爱，但在任务B中可能持否定态度。共享嵌入方法对此问题无效，因为它仅基于一个共享的嵌入，无法捕捉这种矛盾。为验证这一假设，我们选择了40%具有矛盾偏好的用户-商品对，分别来自'Like'任务的前40%和'Finish'任务的后40%（共占9.63%）。

知乎

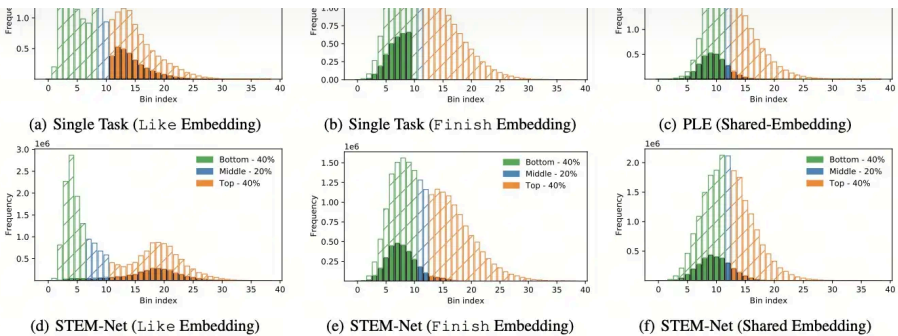


Figure 5: The distance distribution of the contradictory user item pair set S (with solid color) as well as the whole user item pair set (with slash lines) regarding: the single task Like (a) and Finish embedding (b), the PLE embedding (c), and the Like (d) and Finish-specific (e) embedding and shared embedding (f) in STEM-Net.

实验结果显示，与单任务'Finish'嵌入的学习结果相符，STEM-Net的嵌入表（特别是'Like'和'Finish'的任务特化部分）能够清晰地地区分这些差异。同时，与PLE的嵌入对比，STEM-Net的共享嵌入虽然与'Finish'任务关联，但与'Like'任务的分布形成反差，这进一步证实了STEM-Net能识别并处理用户在不同任务上的矛盾偏好。

Online A/B Test

Online Deployment

从2022年开始，STEM-Net已经在腾讯的展示广告+平台进行了广泛的应用研发，专注于多种场景。这些场景涵盖了诸如'Follow'（关注）、'Activation'（激活）、'Fulfill Sheet'（订单完成）和'Pay'（支付）等多种任务。通过STEM-Net，系统能够处理用户在这些不同场景下的复杂偏好，特别是在理解和适应用户在不同任务中的矛盾偏好方面，显示出其优越性。

Performance

在实际生产环境中，我们采用了MMoE架构，每个专家都采用NFwFM，这是一种NFM的扩展版本，通过用FwFM替换传统FM模块。模型设计包含了共享和任务特定的嵌入层以增强功能。实验结果显示，STEM-Net在所有任务上的表现优于基于MMoE的模型。具体成效体现在表中，针对'Follow'、'Activation'和'Fulfill Sheet'三个关键场景，STEM-Net分别提升了0.32%，0.24%，和0.48%的平均AUC。这直接导致在线A/B测试中，GMV（商品价值）的增长分别达到4.2%，3.9%，和7.1%。这充分证明了STEM-Net在广告推荐领域的显著优势。

Table 7: AUC Lift of Online A/B Test

Scenario	Follow	Activation	Fulfill Sheet	Pay	Avg.
Scenario 1	+0.29%	+0.33%	+0.29%	+0.35%	+0.32%
Scenario 2	+0.13%	+0.22%	+0.33%	+0.27%	+0.24%
Scenario 3	+0.47%	+0.39%	+0.28%	+0.78%	+0.48%

原文《STEM: Unleashing the Power of Embeddings for Multi-task Recommendation》

编辑于 2024-05-31 07:20 · IP 属地北京

STEM多目标优化推荐系统

赞同137

5条评论

分享

喜欢

收藏

申请转载

...

理性发言，友善互动

5条评论

默认最新

董德利团茨操

还看到有一篇对序列切成多个滑动子序列做的,计算成本真就一个比一个高

05-31 · 广东

回复

1

知乎

06-03 · 湖南

回复 喜欢



请叫我野狼

id类特征做多个embedding，开销也太大了

05-21 · 上海

回复 喜欢



王天枢

试了 那个task embedding对PS开销很大，目前还没收敛看到效果

05-17 · 美国

回复 喜欢



aliez

task embedding的输入特征得精简下，如果像user_id或item_id这种可能难收敛，反而影响模型效果，我觉得比较适合场景id或一些task特有的上下文特征

05-28 · 北京

回复 喜欢

推荐阅读

2024腾讯最新报告：做全域，你需要知道的2件事

2024腾讯最新报告：做全域，你需要知道的2件事

刘润



腾讯TGW网关团队最新研究成果入选NeurIPS 2023

腾讯技术工程

看完腾讯开年改革，咋感觉全员烧钱时代要结束了.....

据内部人士透露，腾讯IEG昨日（1月8日）召开了线上员工大会。会议上，除Mark、Steven等高层对未来的展望外，Colin、Jerry、Enzo、Ethan等一众工作室老板也逐一上台，向大家介绍了各自团...

游戏葡萄

发表于游戏葡萄



腾讯今晨公布第三次战略进军产业互联网

DeepT...

发表于C