

Pinterest 2023革新之作—基于Transformer用户实时兴趣建模



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

16 人赞同了该文章

Introduction

推荐系统是解决在线信息过载问题的有效工具，用于帮助企业个性化推荐相关的产品、图像、视频和音乐，从而提升用户满意度和企业业务发展。

Pinterest是一个大型的内容分享和社交媒体平台，它拥有数十亿的钉子，这些钉子包含丰富的上下文和视觉信息。当用户访问Pinterest时，他们首先看到的是主页，这是平台主要的流量来源，占了平台总参与度的大部分。

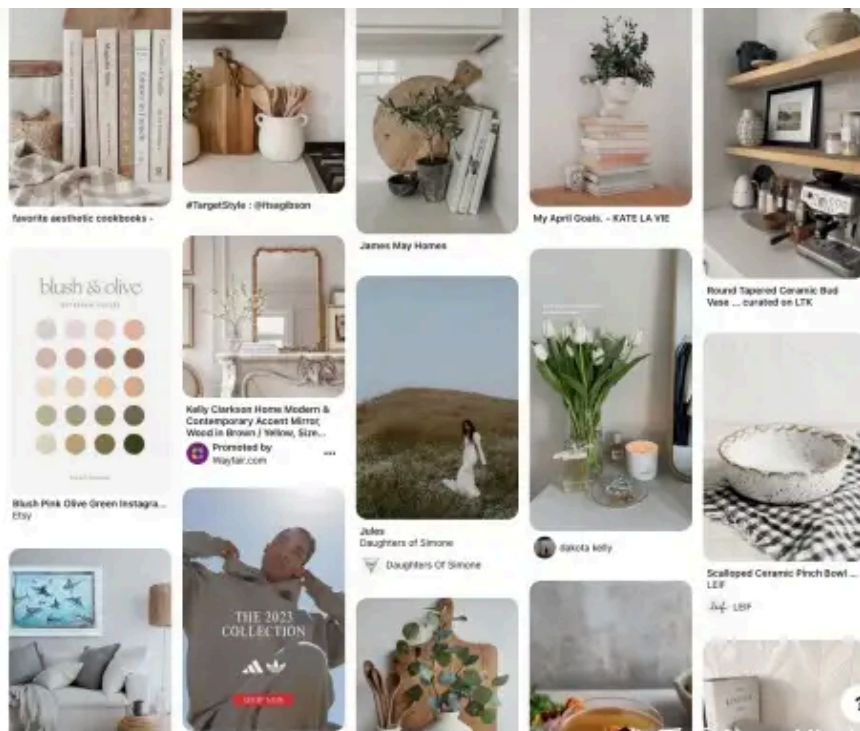


Figure 1: Pinterest Homefeed Page

主页由一个由用户兴趣和活动驱动的阶段推荐系统组成，在检索阶段，系统会根据用户的兴趣和其他因素（如关注的板）筛选出Pinterest上的钉子。然后，系统使用点对point-wise模型来预测每个候选钉子与用户的相关性。最后，系统使用融合层调整排序结果以满足业务需求。

实时推荐非常重要，因为它能给用户提供快速和最新的建议，从而提高用户体验和满意度。使用实时数据*进行实时推荐可以提供更准确的建议和增加发现相关项的可能性。然而，较长的动作序列会导致更好的用户表示和更高的推荐性能，但同时也可能增加计算资源的需求并导致延迟增加。

为了解决这个问题，有些方法使用哈希*和最近邻搜索在长用户序列中查找用户。其他工作则通过将用户的过去动作编码成用户嵌入来表示长期用户兴趣。通过将TransAct的表达力和批量用户嵌入相结合，混合排名模型可以提供实时的行为反馈并考虑用户的长期兴趣。实时和批量组件相辅相成，从而提高推荐准确度，进而改善主页用户体验。

- 描述Pinterest主页流推荐系统：Pinnability架构，主页流个性化推荐占总用户参与度高。
- 我们将服务优化应用于Pinnability，使在引入TransAct时能够承受计算复杂度增加65倍。通过优化，支持我们的CPU基模型使用GPU进行服务。
- 在真实推荐系统上使用TransAct解决在线A/B实验中的实际问题，如推荐多样性下降和参与度衰减。

Methodology

本节讨论实时批混合排名模型TransAct，首先简介Pinterest主页Feed排序模型Pinnability，然后说明如何使用TrancAct将Pinnability编码为实时用户操作序列特征。

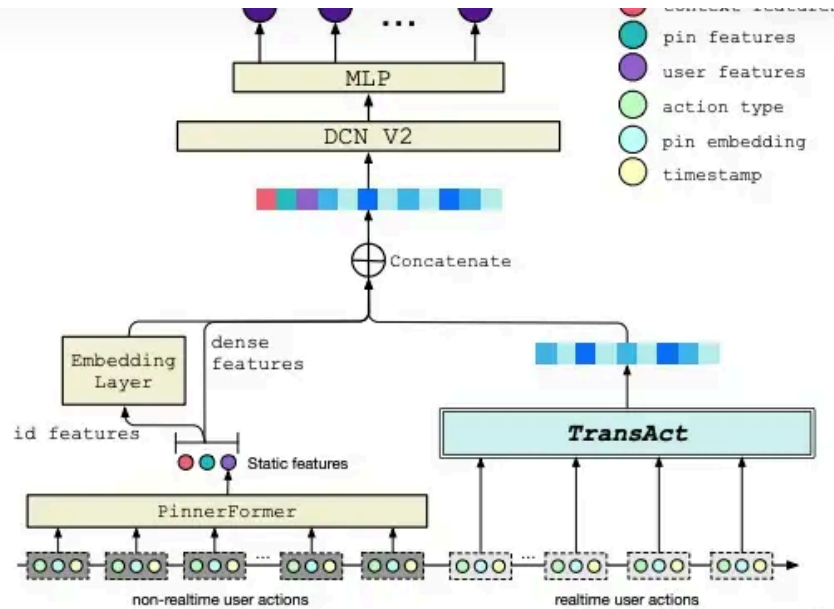


Figure 2: Pinterest Homefeed ranking model (Pinnability)

Preliminary: Homefeed Ranking Model

推荐任务是点对多任务预测问题，给定用户 u 和钉子 p ，我们预测用户 u 对候选钉子 p 执行不同动作的概率，包括正面行动和负面行动。我们提出并实现了Pinnability模型，采用宽和深度学习架构，使用多种类型输入信号，如用户信号、钉子信号和上下文信号，以及各种输入的表达方式，如分类形式、数值形式和嵌入特征形式。我们还使用全秩DCN V2进行特征交叉，使用全连接层⁺预测用户对候选钉子的操作。我们的模型是实时-批量混合模型⁺，通过TransAct和PinnerFormer方法同时编码用户操作历史特征，并优化排名任务。

$$L(\mathbf{y}, \hat{\mathbf{y}}) = - \sum_{k=1}^{|\mathbf{y}|} y_k \log \left(\frac{\hat{y}_k}{p(y_k)} \right) w_k$$

$$\mathcal{L} = w_u \sum_{h \in H} \{-w_h [y_h \log f(\mathbf{x})_h + (1 - y_h)(1 - \log f(\mathbf{x})_h)]\}$$

设 $f(\mathbf{x}) \in (0, 1)^H$ ，其中 $f(\mathbf{x})_h$ 为第 h 个头的输出概率。设 $y_h \in \{0, 1\}$ 为第 h 个头的真实标签。对于每个头 h ，采用一个权重 w_h 来计算其输出 $f(\mathbf{x})_h$ 的交叉熵⁺。 w_h 由真实标签 \mathbf{y} 和一个标签权重矩阵 $\mathbf{M} \in \mathbb{R}^{H \times H}$ 来计算：

$$w_h = \mathbf{M}[y_h, h]$$

$$w_h = \sum_{a \in H} \mathbf{M}_{h,a} \times y_a$$

给定的权重表达式为：

$$w_u = w_{\text{state}} \times w_{\text{location}} \times w_{\text{gender}}$$

Realtime User Action Sequence Features

为了提高用户体验和系统效率，我们在用户操作序列中选取了最近100个动作作为特征，并填充小于100个动作的用户特征为长度为0。这些特征按时间戳降序排序，即首先输入最recent的动作。用户操作序列中的所有动作都是钉级动作。对于每个动作，我们使用三个主要特征：动作发生时间，动作类型和针的32维PinSage嵌入。PinSage是一种dense的嵌入，可以编码钉的内容信息。

Our Approach: TransAct

实时用户动作序列特征 $\mathbf{S}(u)$ 是由特定子模块 TransAct 处理的动态特征。TransAct 从用户的过去行为中提取顺序模式，并预测 (u, p) 的相关性分数。

Feature encoding

其相关性应该极低。为了考虑到这些重要因素，我们将使用可训练的嵌入表将操作类型映射到低维向量。然后，用户的行为序列会被投影到一个用户行为嵌入矩阵

$$\mathbf{W}_{actions} \in \mathbb{R}(|S| \times d_{action})$$

其中 d_{action} 是操作类型嵌入的维度。正如之前所述，钉子的内容在用户的行为序列中由PinSage嵌入表示。因此，所有钉子内容的用户行为序列矩阵是

$$\mathbf{W}_{pins} \in \mathbb{R}(|S| \times d_{PinSage})$$

最后，将所有的用户动作和钉子内容的嵌入特征通过CONCAT函数连接在一起，得到最终的用户动作序列特征

$$\text{CONCAT}(\mathbf{W}_{actions}, \mathbf{W}_{pins}) \in \mathbb{R}(|S| \times (d_{PinSage} + d_{action}))$$

Early fusion⁺

在排名模型中使用用户操作序列特征的一个独特优势是能够模拟候选插件⁺与用户之间的交互。早期融合是推荐任务中提高排名性能的重要因素，包括用户和项目特性在模型早期阶段的合并。两种早期融合方法-----直接融合和混合融合-----已得到实验验证。

- 将PinSage中的候选pin插入到用户操作序列的末尾，如同BST一样。采用零向量作为模拟动作类型。
- 将用户操作序列与候选pin的PinSage嵌入连接。

我们选择concat作为早期融合方法，基于离线实验结果。最终的序列特征与早期融合构成一个二维矩阵

$$\mathbf{U} \in \mathbb{R}^{|S| \times d}$$

其中 $d = (d_{action} + 2d_{PinSage})$ 。

Sequence Aggregation Model

准备好用户动作序列特征 \mathbf{U} ，接下来的任务是有效聚合这些信息以表示用户的短期偏好。行业中的流行模型架构有CNN、RNN和Transformer等。我们试验了不同的序列聚合方法并选择了一种基于Transformer的架构。我们使用标准的Transformer编码器，包含2个编码层和1个头。前馈神经网络的隐藏维度为 d_{hidden} 。由于我们的离线实验发现位置信息无效（详情请参阅附录）。

Random Time Window Mask

兔子洞⁺效应 可能导致模型推荐内容相似，降低Homefeed的多样性，影响长期用户留存。为解决此问题，使用用户动作序列的时间戳构建时间窗口掩模，过滤部分位置。在前向传递中，随机选取0-24小时内的时间窗口 T ，忽略在此区间内进行的操作。训练时使用掩模，推理不应用。

Transformer Output Compression

Transformer编码器输出的矩阵

$$\mathbf{O} = (\mathbf{o}_0 : \mathbf{o}_{|S|-1}) \in \mathbb{R}^{|S| \times d}$$

的前 K 列

$$(\mathbf{o}_0 : \mathbf{o}_{K-1})$$

与 max 池化向量 MAXPOOL

$$(\mathbf{O}) \in \mathbb{R}^d$$

知乎

$$\mathbf{z} \in \mathbb{R}^{(K+1)*d}$$

这表示前 K 个输出列捕捉了用户的最新兴趣，而 $\text{MAXPOOL}(\mathbf{O})$ 则代表了用户对 $\mathcal{S}(u)$ 的长期偏好。由于输出足够紧凑，所以可以直接将其集成到 Pinnability 框架中的 DCN v2 特征交叉层中。

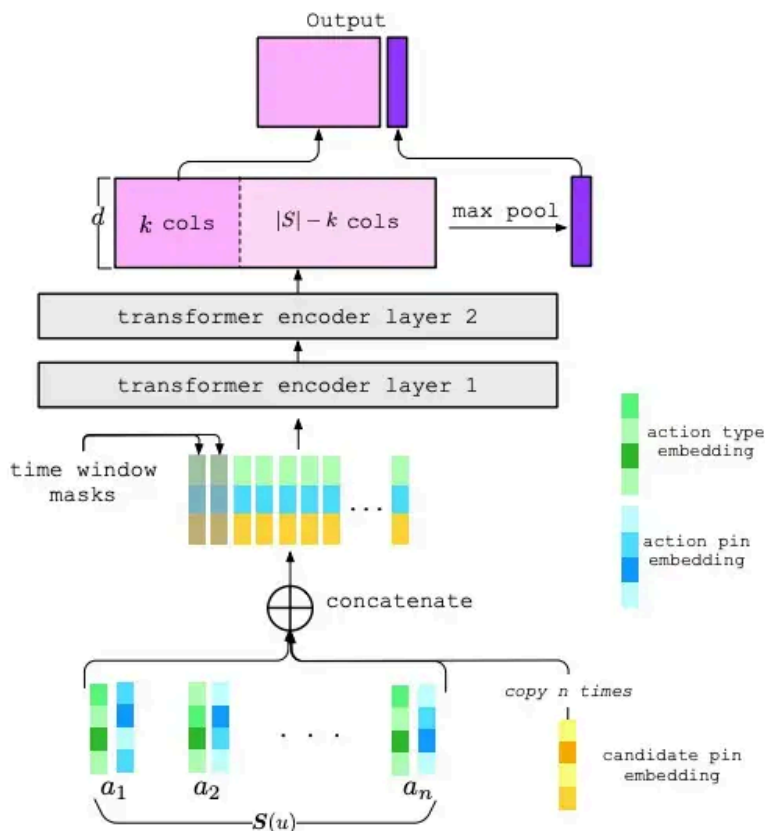


Figure 3: TransAct architecture. Note that this is a submodule that can be plugged into any similar architecture like Pinnability

Model Productionization

Model Retraining

重训练是推荐系统的重要组成部分，它有助于系统随着用户行为和偏好的变化而动态调整。如果没有定期重训，推荐系统的性能可能下降，导致推荐结果失准。特别是当使用实时特征时，由于模型具有较强的时效性，因此需要频繁地进行重训，否则可能导致模型在几天内过时，影响预测准确性。我们的研究表明，每周进行两次从头开始的重训可以保证参与率的一致性，并且维持可管理的培训成本。在后续章节中我们将深入讨论重训练的重要性。

GPU serving

一种有效的合并CUDA内核方法是尽可能多地合并操作。我们利用nvFuser进行编译，但由于许多剩余操作需要人工干预，我们使用了cuCollections来支持GPU上的哈希表⁺，以支持原始ID的查找，并实现了一个自定义的综合嵌入查找模块，将多个特征的查找合并成一个操作，从而将与稀疏特征相关的几百个操作减少为一个操作。优化内存复制和批次大小，使用CUDA Graphs捕获模型推理过程，减少小操作开销。

Realtime Feature Processing

当用户执行操作时，一个基于Flink (Apache Flink) 的实时特征处理应用消费来自前端事件产生的Kafka (Apache Kafka) 流。它验证每个动作记录，检测和合并重复项，并管理多个数据源之间的时间差异。然后，应用程序将特征材料化并存储在Rockstore (Apache Rockstore) 中。在服

Experiment

展示了TransAct的离线和在线A/B实验结果，比较了其与其[基线模型](#)在Pinterest内部训练数据上的表现。

Experiment Setup

Dataset

我们通过收集 Pinterest 主页流视图日志 (FVL) 中的数据来构建一个离线训练数据集。在第一到两周，我们将模型训练在这上面，在第三周进行评估。我们的训练数据集包含30亿个实例，涵盖了177百万用户和720百万 pins。为了平衡高度偏斜的数据集，我们对负样本进行了[下采样](#)，并设置了正负样本的比例固定。我们没有使用公共数据集，因为它们缺乏实时用户行为序列元数据特征，如项嵌入和行动类型，这些特性对于我们 TransAct 模型来说是必要的。此外，它们也不支持我们的实时批处理混合模型，这种模型需要同时考虑实时和批用户特征，并且不能在线进行 A/B 测试。

Hyperparameters

这段内容主要介绍了训练模型所需的一些参数设置。其中包括[序列长度](#) $|S| = 100$ ，动作嵌入维度 $d_{action} = 32$ ；序列特征编码使用一个由两个transformer块组成的transformer编码器处理，且Dropout率为0.1；transformer编码层中的前馈网络维度为 $d_{hidden} = 32$ ，不使用[位置编码](#)；使用的优化器是Adam，带有一个学习率调度器；学习率在开始时有一个预热阶段，持续5000步，然后逐渐增加到0.0048，最后通过[余弦](#)退火降低；批量大小为12000。

Offline Experiment

Metrics

离线评估数据采用FVL随机采样，具有代表性，减少了评估方差。该方法解决了位置偏斜问题，通过在推荐顺序前随机打乱，避免了推荐列表顶部的物品获得更多关注和参与度。模型在HIT@3上进行评估，按照用户ID、钉子ID和块ID分组。

$$\mathcal{S} = \sum_{h \in H} u_h f(\mathbf{x})_h$$

接下来，我们将从每个块中选择排名前 K 的钉子，并计算所有头的匹配次数@3，记作 $\beta_{c,h}$ 。具体来说，如果一个块 $\mathbf{c} = [p_1, p_2, p_3, \dots, p_n]$ 按照排序函数 \mathcal{S} 排列，当用户重新推文 p_1 和 p_4 时， $\beta_{c,repin} = 1$ ($K = 3$)。

$$HIT@3/h = \frac{\sum_{u \in U} \sum_{c \in C_u} \beta_{c,h}}{|U|}$$

注意：HIT@K评分越高，模型性能越好；HIT@K/隐藏分数越低，越受欢迎。非核心用户定义为过去28天内未积极保存帖子的用户，他们活动较少，增加推荐相关性面临挑战。虽然如此，保留非核心用户至关重要，因为他们维护多样性和繁荣的社区，有助于平台长期增长。所有结果均具统计学意义 ($p < 0.05$)。

Results

对比TransAct与其他序列推荐方法：1) WDL模型作为基础，将序列特征纳入宽特征；2) 利用PinSage嵌入平均池化处理用户操作序列。另外还研究了阿里巴巴行为序列Transformer(BST)的两种变体模型。

with TransAct. (statistically insignificant)

Methods	HIT@3/repin		HIT@3/hide	
	all	non-core	all	non-core
WDL + seq	+0.21%	+0.35%	-1.61%	-1.55%
BST (all actions)	+4.41%	+5.09%	+2.33%	+3.59%
BST (positive actions)	+7.34%	+8.16%	-1.12%*	-3.14%*
TransAct	+9.40%	+10.42%	-11.36%	-13.54%

Ablation Study

Hybrid ranking model

表1显示，随着删除模型中的每个组件，用户所有特征的相对下降量。TransAct最显著地提高了用户参与度，而PinnerFormer则依赖于用户的历史行为来预测长期偏好。尽管TransAct对于模型理解至关重要，但我们发现大规模训练和长期兴趣捕获也有价值，并可以为实时交互序列的推荐提供补充。实验结果表明，将实时交互序列模型与预训练批量模型结合非常有效。

Table 2: Ablation study of realtime-batch hybrid model

TransAct	PF	Other User Features	HIT@3/repin	HIT@3/hide
✓	✓	✓	—	—
✓	✗	✓	-2.46%	+3.61%
✗	✓	✓	-8.59%	+17.45%
✓	✓	✗	-0.67%	+1.40%

Base sequence encoder architecture

我们使用平均池化、1D CNN、2层RNN和2层LSTM以及vanilla Transformer来处理实时用户行为序列特征，并对这些模型进行离线评估。结果表明，虽然简单的平均池化+方法可以提高参与度，但更复杂的架构并不总是比平均池化性能更好。其中，vanilla Transformer实现了最佳性能，它不仅显著减少了HIT\@3/hide，还提高了HIT\@3/repin。

Table 3: Offline evaluation of sequence encoder architecture

Sequence Encoder	HIT@3/repin	HIT@3/hide
Average Pooling	+0.21%	-1.61%
CNN	+0.08%	-1.29%
RNN	-1.05%	-2.46%
LSTM	-0.75%	-2.98%
Vanilla Transformer	+1.56%	-8.45%

Early fusion and sequence length selection

在Section中，早期融合对排名模型至关重要。它能考虑不同项目间的依赖关系，并明确学习排名候选针和过去互动过的每个针的关系。长序列较短序列更有表现力，因此我们评估了模型在不同输入序列长度下的性能。

图 显示了序列长度与性能之间的正相关关系。性能随序列长度以线性比例增加。早期融合中，串联优于附加。因此，最优的参与增益可通过使用最大可用序列长度和串联实现。

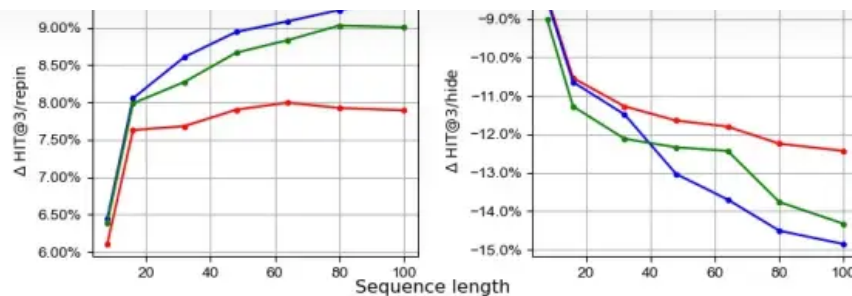


Figure 4: Effect of early fusion and sequence length on ranking model performance (HIT@3/repin, HIT@3/hide)

Transformer hyperparameters

我们通过对Transformer超参数的调整来优化TransAct的编码器。实验结果表明，增加Transformer层的数量和Feedforward维度可以提高性能，但也会增加延迟。其中，4个Transformer层*和384作为Feedforward维度可以获得最佳性能，但也带来了30%的延迟增加，不符合我们的延迟要求。因此，我们选择了2个Transformer层和32作为隐藏维度，以实现良好的性能与用户体验之间的平衡。

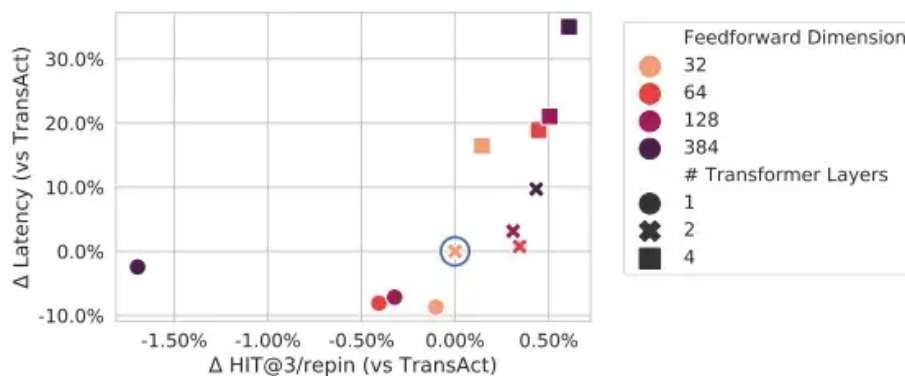


Figure 5: Effect of transformer hyperparameters on model performance and latency

Transformer output compression

如图2所示，在融合第一K列并且对整个序列执行最大池化之后，可以获得最高的HIT@3/repin。这个方法相比于使用所有列提供了更好的性能和更短的延迟。在TransAct中，我们选择K=10。

Online Experiment

相较于离线评估，在线实验的优势在于其可在实时用户数据上运行，使模型在更真实和动态的环境中进行测试。在线实验使用了在两周离线训练数据集上训练的排名模型，其中对照组是不包含实时用户序列特征的Pinnability模型，而实验组则是带有所谓的TransAct的Pinnability模型。每个实验组为总用户量的1.5%提供服务。

metrics

重要指标

Online engagement

在TransAct排名引入后，Homefeed页面上的重新标记数量提升了11%，同时隐藏数量也有所下降。另外，对于非核心用户，他们在较短的时间内没有稳定的操作历史，但是由于实时功能可以捕捉他们的兴趣，他们的参与度更高。使用TransAct后，Homefeed页面可以更快地响应用户行为，并及时调整排名结果，这导致了整体时间花费在Pinterest上的增加。

TransAct的交互指标随着时间推移呈现下降趋势。如图所示，我们对TransAct主页推荐卷积体积的增长与基准组的表现，发现如果没有进行重新训练，初始交互率明显高于基准组；但在实验进行至第14天，交互率下降至较低水平。相反，如果对新鲜数据进行重新训练，相比未重新训练模型，交互率有显著提升。这表明TransAct对用户行为变化十分敏感，需定期进行重训。实际应用中，我们设定重训频率为每周2次，并验证该频率可维持交互率稳定性。

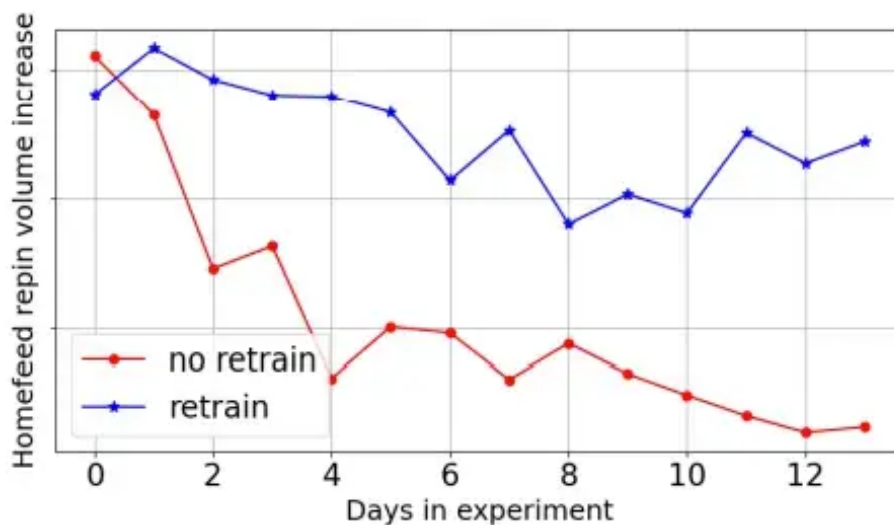


Figure 6: Effect of retraining on TransAct

Random time window masking

在Pinterest上，我们有一个由28,000个节点组成的分层兴趣分类体系（interest taxonomy）。最顶级的兴趣类别粗略，例如艺术、美容和体育。在这里，我们衡量“印象多样性”作为每个用户查看独特顶级兴趣数的总和。观察到，随着将TransAct引入Homefeed排名，展现多样性下降了2%至3%。解释是，通过添加用户行为序列特征，排名模型学习优化用户的短期兴趣。并且通过主要关注短期兴趣，推荐的多样性降低。

我们通过在Transformer中使用随机时间窗口掩模来缓解这种多样性的减少。正如Section所述。这种方法鼓励模型专注于除最近用户参与的物品以外的内容。在这种设计下，多样性的度量减少仅-1%，而不影响如重新发表量的推荐指标。我们也尝试使用更高的Transformer编码器层丢弃率和随机屏蔽固定比例的用户动作序列输入。但是，这两种方法都不如随机时间窗口掩模好。它们增加了多样性，但降低了参与度。

Discussion

Feedback Loop

在线实验揭示了TransAct的真实潜力并未完全发挥。当模型作为生产Homefeed排名模型时，在全流量情况下性能将更大提升。这是因为正向反馈循环效应：随着用户使用更响应性Homefeed，会吸引更多相关内容交互，进而引起行为改变（如更多点击或重新保存），这导致实时用户序列特征发生变化，新训练数据生成则可用来重新训练Homefeed排名模型。这样反复进行可以产生正向累积影响，从而提升参与率并强化反馈循环。类似文献中的直接反馈环，即一个模型可直接影响未来训练数据选择，并随着时间推移而逐渐显现的现象。

TransAct in Other Tasks

TransAct不仅用于任务排名，还在上下文推荐和搜索排序中有广泛应用。其个性化推荐模型（称为“相关帖子”排名）被用于Pinterest的“搜索”排名系统和“通知”排名系统。这些例子证明了TransAct的有效性和在实际应用中的可能性。

Application	Metrics	Δ
Related Pins	Repin Volume	+2.8%
Search	Repin Volume	+2.3%
Notification	Email CTR	+1.4%
	Push Open Rate	+1.9%

Conclusions

我们提出并验证了实时用户动作模型TransAct，它有效捕捉用户的短期兴趣，利用实时动作编码提高推荐效果。我们将实时与批处理编码的优点结合起来，并在Pinterest Homefeed推荐系统中取得成功。我们的离线实验结果显示，TransAct在推荐系统基准上显著优于竞争对手。

原文《TransAct: Transformer-based Realtime User Action Model for Recommendation at Pinterest》

编辑于 2024-02-20 16:01 · IP 属地北京

推荐系统 Transformer 工业级推荐系统

赞同 16 添加评论 分享 喜欢 收藏 申请转载 ...





理性发言，友善互动



还没有评论，发表第一个评论吧

推荐阅读



Google Colab 现已支持直接使用  transformers 库

Hugging Face



使用Transformer在Twitter数据集上进行情感分类

安大叔 发表于视觉项目聚...

简析阿里 BST: 当用户行为序列邂逅Transformer

本文介绍 阿里搜索推荐团队 2019 年 发布在 arXiv 上的文章《Behavior Sequence Transformer for E-commerce Recommendation in Alibaba》。文中提出BST模型，利用近年因...

刺猬 发表于咖啡与机器...



Google Transformer 模文详解

ZZFZXY