

赞同 67

分享

## 快手2023-用分而治之思想《从多任务角度实现更出色的向量召回效果》-论文深度解析



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

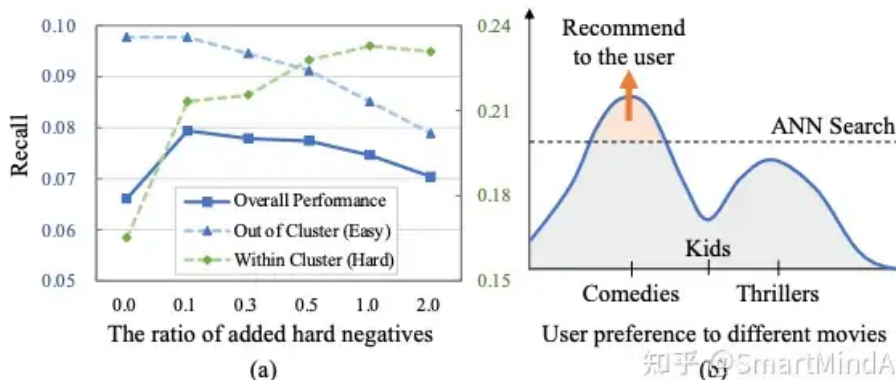
已关注

67 人赞同了该文章

### Introduction

在有限时间内，从庞大的内容库中精准找到相关内容是工业推荐系统<sup>+</sup>的一大挑战。常用的解决方法是两阶段方法，首先通过候选生成阶段缩小内容候选集，再进行排序。传统上，候选生成主要基于规则，如标签推荐和协同过滤<sup>+</sup>。但随着深度学习的兴起，基于嵌入的检索（EBR）方法变得更为普遍。这种方法通过深度神经网络<sup>+</sup>将用户和内容特征编码为嵌入，学习用户向量和内容向量间的距离以区分相关与否。预先计算的内容嵌入可通过ANN系统进行索引，实现高效检索。尽管EBR具有简单有效的优点，但仍存在一些根本性问题。

- **容易负样本和困难负样本的权衡** 在候选生成阶段，需面对整个物品语料库<sup>+</sup>，区分竞争性候选物品（正样本）和非竞争性候选物品（负样本），包括不相关物品（容易负样本）和有一定相关性但不具备竞争性的物品（困难负样本）。挑战在于，嵌入模型在两项任务上的能力常相互冲突（见图）。常用混合容易和困难负样本训练EBR模型，但需仔细调整比例。我们认为这种“最佳点”并非最优，而是一种妥协，限制了EBR模型的潜力。（EBR模型指双塔模型<sup>+</sup>，广泛用于推荐系统。）
- **无法控制的多样性与公平性**。用户对不同内容有不同兴趣，如常看喜剧的人偶尔也看惊悚片。理想的推荐系统应提供多样化结果，满足用户各种需求。但EBR中的ANN搜索算法<sup>+</sup>只贪婪搜索得分最高内容，忽视用户意图的多样性性质（见图）。因此，结果多样性取决于嵌入模型和长尾用户兴趣，如惊悚片，可能被低估。



受这些问题启发，我们提出了一种简易且高效的方法，能大幅提升EBR模型的效果，且几乎无需额外成本。该方法采用分治算法<sup>+</sup>的原理，将整体内容集划分为多个集群，假设同一集群内的内容大部分彼此相关。我们并非使用EBR模型整体搜索相关候选内容，而是在每个集群（或相关集群的子集）上并行运行EBR，随后使用特定策略合并结果。此方法排除了来自不相关集群的负面干扰，使

入了一个多任务学习 (MTL) 场景, 为每个集群检索相关候选者视为单个任务。这种视角为优化模型准确性提供了新的方向。然而, 发现最先进的MTL方法增加了**计算成本**<sup>+</sup>, 但效果提升未达到预期。因此, 我们汲取了基于提示的轻量级调整进展的灵感, 促进了输入层的任务适应, 实现了相当大的准确性改进。所提出的方法使得每个任务能够使用统一的模型架构进行更有效的微调。在线A/B测试和用户参与度指标显示, 我们的方法显著改善了关键指标, 召回率提高了高达40%。**消融研究**<sup>+</sup>进一步验证了分治方法和基于提示的任务适应技术的有效性。这项初步研究为EBR的改进展示了新的可能性和视角, 为未来的研究留下了开放的问题, 例如更好地划分候选空间和探索更先进的MTL方法。

## Methodology

### Background: Embedding-based Retrieval

假设有用户集 $\mathcal{U}$ 和物品集 $\mathcal{I}$ 。用户 $u$ 的特征为 $\mathbf{x}_u$ , 物品 $i$ 的特征为 $\mathbf{x}_i$ 。EBR方法通过用户编码器 $f$ 与物品编码器 $g$ 将它们转为嵌入向量 $\mathbf{e}_u = f(\mathbf{x}_u)$ 和 $\mathbf{e}_i = g(\mathbf{x}_i)$ 。物品与用户的相关性为 $r_{ui}$ , 通过用户嵌入 $\mathbf{e}_u$ 和物品嵌入 $\mathbf{e}_i$ 的距离计算 (如**内积**<sup>+</sup>)。物品嵌入 $\{\mathbf{e}_i\}_{i \in \mathcal{I}}$ 可预先计算并索引在ANN系统 (如FAISS)。实时评估用户编码器 $f(\mathbf{x}_u)$ 后, ANN能亚线性时间内检索最近物品。EBR的优势在于 $f$ 和 $g$ 可为任何**神经网络模型**<sup>+</sup>, 且部署成本低。本文以SASRec为例, 该模型采用Transformer作为用户编码器。

在训练过程中, 为了区分用户点击过的物品 ( $\mathcal{I}^u$ ) 和未点击的物品 ( $\mathcal{I} \setminus \mathcal{I}^u$ ), 我们训练了物品相关得分。具体而言, SASRec模型采用了二元**交叉熵**<sup>+</sup>损失函数进行训练。其中,  $\mathbf{s}_1^u, \mathbf{s}_2^u, \dots, \mathbf{s}_n^u$ 代表用户 $u$ 近期点击过的 $n$ 个物品的嵌入向量, 并且仅使用潜在物品ID的嵌入作为物品表示。在训练期间, 我们优化这个**损失函数**<sup>+</sup>以更新模型的参数。通过这种方式, 模型能够学习到如何更好地预测用户的点击行为, 并据此为用户提供更加精准的物品推荐。

$$\mathcal{L} = - \sum_{u \in \mathcal{U}} \sum_{i^+ \in \mathcal{I}^u} [\log(\sigma(r_{ui^+}))] + \mathbb{E}_{i^- \sim \mathcal{I} \setminus \mathcal{I}^u} [\log(1 - \sigma(r_{ui^-}))].$$

### Proposed Divide-and-Conquer Approach

如引言所述, 负项集合 $\mathcal{I} \setminus \mathcal{I}^u$ 的结构多样, 含不相关或竞争力不足的项。随候选集合 $\mathcal{I}$ 规模扩大, 此问题愈发突出。我们受“分治”思想启发, 将候选集划分为 $K$ 个语义相关的簇 $\mathcal{I} = \{\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_K\}$ , 用EBR模型在各簇内检索相关项 (见图)。聚类标准可依场景选取。实验表明, 基于Word2Vec训练的项嵌入的K-means聚类效果良好。生产系统中, 我们用内部视频类别预测内容特征, 解决冷启动问题并提高可解释性。划分检索空间后, EBR模型仅需处理每簇内的候选项, 训练时只需使用与正项来自同一簇的负项。此方式使EBR模型更聚焦, 从而可能更“高效”。训练损失相应重写为:

$$\mathcal{L} = - \sum_{k=1}^K \sum_{u \in \mathcal{U}} \sum_{i^+ \in \mathcal{I}^u \cap \mathcal{C}_k} [\log(\sigma(r_{ui^+}))] + \mathbb{E}_{i^- \sim \mathcal{C}_k \setminus \mathcal{I}^u} [\log(1 - \sigma(r_{ui^-}))].$$

一旦获得每个集群的最高得分项, 我们可为每个集群分配足够配额, 以满足定制要求。要生成大小为 $M$ 的最终候选集, 我们用用户编码器的特征来训练用户意图模型, 预测相关内容与用户 $u$ 可能属于集群 $\mathcal{C}_k$ 的概率 $p_{uk}$ 。接着, 组合每个集群的前 $M_k$ 个项目作为最终结果, 其中

$$M_k = M \cdot [(p_{uk})^\alpha / \sum_{k'=1}^K (p_{uk'})^\alpha], \alpha \text{ 为可调超参数} [\wedge 1].$$

由于EBR在不同内容集群上可并行运行, 且每次运行的**搜索空间**<sup>+</sup>缩小约 $K$ 倍, 此分治过程不会显著增加响应延迟。

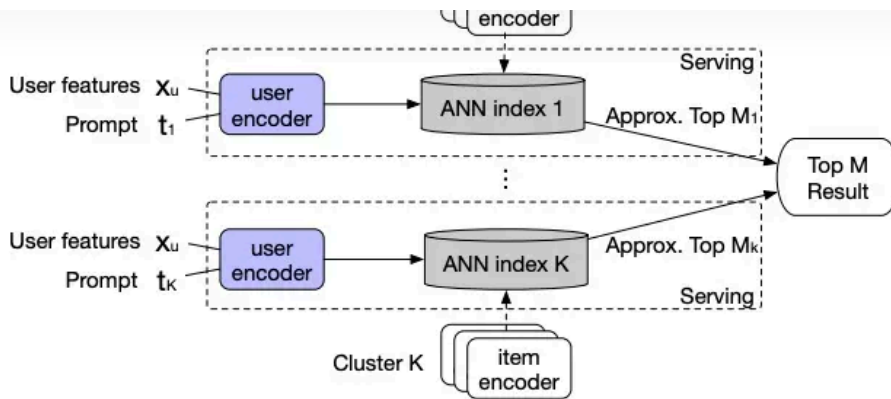


Figure 2: An overview of the system architecture.

### Prompt-like Multi-task Learning

通过观察训练损失，我们在训练 $K$ 个独立子任务，第 $k$ 个子任务对应检索 $C_k$ 中的相关项。因此，可能有机会进一步利用这些子任务之间的正向迁移，而不是简单地将所有任务样本混合在一起。这种多任务学习的观点促使我们尝试一些先进的MTL方法，例如MMoE。然而，如上所述，其有效性和效率都不如预期。相反，受基于提示的调优方法的启发，我们尝试将任务标识作为提示输入到用户编码器中，以促进任务适应的参数有效方式。也就是说，对于来自每个簇 $C_k$ 的样本，

$$e_u = f(x_u) = \text{Transformer}([t_k; s_1^u, s_2^u, \dots, s_n^u]), t_k \text{ 表示第 } k \text{ 个任务的可训练嵌入。}$$

尽管此方法缩短了训练时间，但效果并未得到显著提升。我们推测，这可能是因为推荐模型中的Transformer层数较少，远少于NLP场景中的常见层数 $[^2]$ 。因此，几层自注意力块可能无法充分捕捉提示与原始输入间的交互。为解决此问题，我们提议利用Hadamard积 $+$ 来加强特征交互。

$$e_u = f(x_u) = \text{Transformer}([s_1^u \odot t_k, s_2^u \odot t_k, \dots, s_n^u \odot t_k]).$$

此方法与HyperPrompt中的方法类似，均是将每个输入令牌嵌入与提示相连接。

### Offline Evaluation

#### Datasets

我们在两个公开数据集上进行了广泛实验，数据集统计数据如表。

- **ML-1M** 包含6,040个MovieLens用户对3千多部电影的1百万条匿名评价。用户观看过的电影都被视为相关的。
- **KuaiRand** 是一个公开数据集，来源于快手推荐系统的日志。在实验中，我们视点击过的物品为与用户相关的物品，仅保留主要推荐场景中的互动（即"tab"字段等于一）。同时，过滤掉了物品和用户频率分别小于70和10的数据。

Table 1: Statistics of datasets (after preprocessing).

Dataset	#Users	#Items	#Interactions	Density	#Clusters
ML-1M	6,040	3,706	1,000,209	0.045	10
KuaiRand	25,828	108,025	20,141,835	0.007	15

### Evaluation Protocol and Baselines

- 矩阵分解（MF）是一种经典的推荐方。

- **SASRec+**扩展了SASRec，结合了简单与困难负样本进行训练（困难负样本来源于与正样本相同的集群）。我们在验证集上优化了混合比率并呈现了最佳结果。
- **MIND**使用了胶囊网络中的路由机制，对用户行为进行分组，形成多个集群，并据此获得多个用户嵌入以进行检索。
- **ComiRec-SA**运用多头注意力机制为用户生成多嵌入，捕捉其多元兴趣。
- 在本文中，我们采用了MMoE这一广泛使用的多任务学习方法。具体地，我们的实验采用了自注意力块作为独立的专家，并设置了四个这样的专家。

Experimental results

我们进行了消融研究，评估了分治方法和类似提示的MTL方法。结果显示，删除提示模块后，ML-1M和KuaiRand上的效果分别下降了**5.6%**和**4.2%**，但仍显著提高了SASRec+的效果，证明了分治策略的有效性。为了进一步验证，我们评估了集群内检索效果，限制候选集为同一集群内的内容。如表所示，训练时关注集群内负样本，能提高检索效果，添加提示后效果更佳。

**Table 2: Performance comparison on ML-1M and KuaiRand. The last row reports the relative improvement of our proposed method over its base model, SASRec. R@M is short for Recall@M.**

Method	ML-1M		KuaiRand	
	R@20	R@50	R@500	R@1000
MF	0.120	0.231	0.086	0.132
SASRec	0.183	0.337	0.178	0.254
SASRec+	0.220	0.367	0.200	0.284
MIND	0.186	0.343	0.138	0.201
ComiRec	0.192	0.341	0.188	0.269
Ours	<b>0.234</b>	<b>0.375</b>	<b>0.254</b>	<b>0.359</b>
w/ Naive MTL	0.221	0.366	0.245	0.344
w/ MMoE	0.224	0.371	0.239	0.341
Improvement	+27.9%	+11.3%	+42.7%	+41.3%

同时，MMoE在ML-1M上结果相似，但从表中看到，其训练成本增加，吞吐量下降了约**70%**。相比之下，所提方法几乎无额外训练成本。

**Table 3: The retrieval performance, within the clusters of ground-truth items, of different MTL methods.**

Method	ML-1M			KuaiRand		
	R@5	R@20	R@50	R@100	R@500	R@1000
SASRec	0.191	0.424	0.639	0.172	0.399	0.543
Naive MTL	0.217	0.463	0.674	0.228	0.502	0.658
MMoE	0.223	0.482	<b>0.682</b>	0.218	0.496	0.655
Ours	<b>0.235</b>	<b>0.483</b>	<b>0.682</b>	<b>0.237</b>	<b>0.522</b>	<b>0.682</b>

Method	SASRec	MMoE	Ours (Prompt-like)
Throughput (samples/sec.)	25K	7.2K	25K

Live A/B Experiment

提出的方法在快手实时A/B实验中测试了三天（2022年7月3-5日），涉及4000多万用户，结果统计显著。实验组中，该方法作为候选生成阶段的候选源之一，多数关键用户\*参与指标都有显著改进，如表。与ComiRec来源相比，我们的内容参与率更高，如表。

Table 5: Results of the live A/B experiment in Kuaishou. All the performance gains are statistically significant at  $p < 0.05$ .

App Usage Time	Likes	Follows	Shares	Downloads
+0.096%	+0.75%	+1.01%	+1.04%	+2.40%

Table 6: Comparison between online engagement rates of the recommended items from ComiRec and our approach.

Metrics (%)	Click Rate	Like Rate	Follow Rate	Share Rate
ComiRec	56.1	2.85	0.283	0.209
Ours	60.5	2.92	0.431	0.239
Improvement	+7.8%	+2.5%	+52.3%	+14.4%

Conclusion

本文提出了一种经生产验证的解决方案，旨在克服基于嵌入的检索方法的限制并提高其效果。贡献包括分而治之方法和类似提示的MTL技术。离线和实时A/B实验证明了其有效性。

论文原文《Divide and Conquer: Towards Better Embedding-based Retrieval for Recommender Systems from a Multi-task Perspective》

跟踪最新技术，点击关注我  
www.zhihu.com/people/smartmindai



编辑于 2023-11-05 17:19 · IP 属地北京

推荐系统 工业级推荐系统 多任务学习



理性发言，友善互动

6 条评论

默认 最新



厚德载物、

快手这也能水一篇了么

01-02 · 上海

回复 喜欢



赵客缦胡缨

每个cluster训练单独的item塔吗  
样本不相关的cluster是否就不做负采样了



知乎

蜜桃小财主

这种思路多年前百度有类似的，不过是直接根据一级类目来划分cluster

2023-11-03 · 北京

回复

喜欢

SmartMindAI 作者

可惜百度很少发paper

2023-11-03 · 北京

回复

喜欢

章鱼哥

快手ab实验只看三天?

2023-11-01 · 广东

回复

喜欢

SmartMindAI 作者

最少3天

2023-11-01 · 北京

回复

喜欢

推荐阅读



关于搜索中的向量召回

量子侠 发表于量子侠的量...

推荐召回-向量召回之U2I

在上一篇文章 Blue：推荐召回-传统多路召回 中提到，推荐系统召回路径有如下几个：I2I：计算 item-item相似度，用于相似推荐、相关推荐、关联推荐； U2I：基于矩阵分解，直接给用户推荐i...

Blue



向量召回：深入评估离线体系，探索优质召回方法

汀、人工智... 发表于搜索推荐算...



阿里妈妈Dolphin分布式回技术详解

阿里妈妈技术