

微软 2023：推荐系统的秘密武器——排序与召回模型的协同作战，论文详解



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

51 人赞同了该文章

Introduction

在现代网络服务，如电子商务和在线广告中，**推荐系统**扮演着重要角色，它可以通过向用户推荐符合其兴趣的个性化内容来大大缓解信息过载问题。深度推荐系统需要两阶段工作流程：召回和排序。召回模型以高效率从整个内容中选取一小部分候选内容，并具有一定的上下文（例如用户画像和历史交互）。通常情况下，召回模型学习将上下文和内容表示为向量嵌入，这样就可以根据向量相似性来高效地估计用户对内容的偏好。相比之下，排序模型用于从召回结果中优化最优先的内容。为了获得最佳精度，它通常利用较复杂但耗时的网络，特别是那些在上下文和内容之间建立深度交互的网络（例如DIN、DIEN和DeepFM）。

Existing Problems

召回模型的独立训练仅利用了历史用户-物品交互，它可能受到训练数据稀疏性的限制，从而严重限制了召回质量。此外，召回模型可能会召回排序模型不喜欢的候选物品，从而影响下游性能。独立训练的排序模型通过随机或**启发式方法**收集的训练样本进行学习，这些样本可能过于简单，导致排序模型只能达到有限的鉴别能力。此外，物品分布可能在训练和推理阶段之间存在高度分化，导致排序模型无法有效地识别召回模型生成的高质量候选者。

最近关于召回模型和排序模型的协作的研究越来越多，例如提出了一种两阶段训练流程。然而，这种两阶段工作流程仍然存在几个关键限制。在正向传递中，排序模型仅根据召回结果的精确顶部-k截断进行训练，这可能导致假阴性。此外，当召回截断在推理阶段发生变化时，排序模型可能面临与训练阶段截然不同的物品分布，这会严重损害其预测准确性。在反向传递中，召回模型被训练以保持绝对排序得分的连贯性，并从召回结果中区分排序结果。然而，为了获得高质量的召回，相对排序顺序的一致性才是真正重要的。

Our Solution

在这项工作中，我们提出了一个新颖的框架，称为CoRR，用于召回模型和排序模型的协同训练。在我们的框架中，召回模型和排序模型在统一的流程中进行同时训练，两个模型可以相互强化。

2. 提出了一种新颖的KL散度的渐近无偏估计⁺，以补偿采样引入的偏差，从而在训练过程中实现更准确的知识蒸馏。

我们在四个基准数据集⁺上进行了全面的实验研究，发现CoRR可以显著提高推荐质量，同时对召回模型和排序模型具有独立有效性。

Cooperative Retriever and Ranker

我们的CoRR在训练数据集 $\mathcal{D} = \{(c, k)\}$ 上的训练过程。该数据集包含上下文 c 和正物品 k 的成对。上下文表示除物品以外的所有信息，例如用户、交互历史、时间、地点等。

Overview

为了实现可扩展的推荐，推荐模型依赖于两个模型之间的协作：召回模型(retriever)和排序模型(ranker)。Retriever的目标是从整个物品集中高效率地选择一小部分潜在的正面物品。通常，retriever由物品嵌入 $E_{\theta_1}(i)$ 和上下文嵌入 $E_{\theta_2}(c)$ 之间的相似性表示，即

$$M_{\theta}(i, c) = \text{sim}(E_{\theta_1}(i), E_{\theta_2}(c))。$$

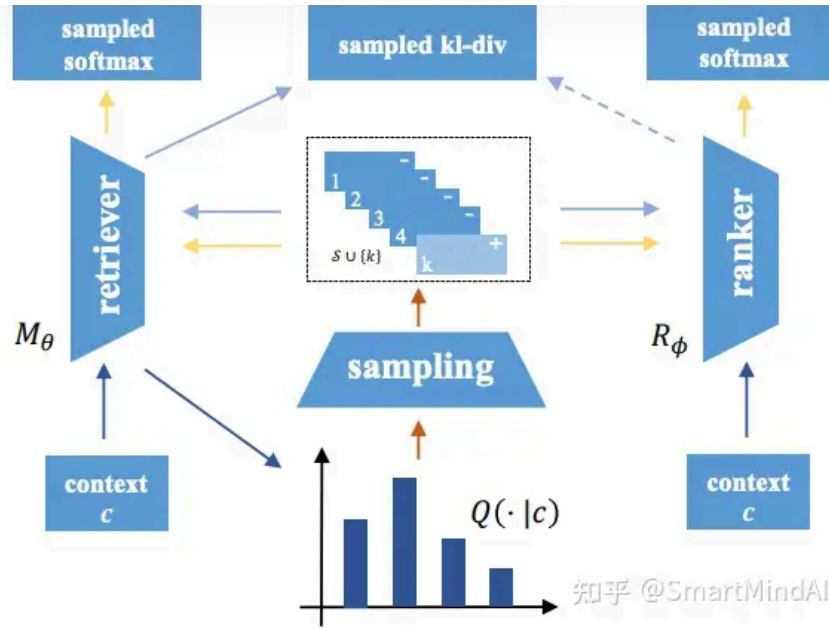
因此，我们可以在线性时间内使用现成的ANNs召回最相似的物品作为候选物品。Ranker旨在从召回结果中高精度地识别出最佳物品，通常由直接将物品 i 和上下文 c 作为输入的具有高准确率且耗时的网络 $R_{\phi}(i, c)$ 表示。由于ranker中的计算时间较长，因此在大多数情况下，传统训练方法学会了将正面物品与随机挑选的负面物品区分开。尽管这种方法很简单，但ranker是独立于retriever进行训练的。在一些情况下，ranker是通过将正面物品与retriever召回结果的top-k⁺区分开来训练训练的。

尽管这种方式与retriever建立了联系，但ranker容易遭受漏报问题 (false negative issue)，因为潜在的正面物品肯定包含在top-k召回结果中。此外，这些负面物品可能会引入大的梯度偏差，导致优化可能缓慢收敛甚至容易陷入局部最优解。为此，我们建议对ranker进行优化，采用基于重要性采样⁺的样本log-softmax的近似无偏估计 (sampled log-softmax)。样本log softmax依赖于从提议分布中随机抽样的样本。

为了将ranker与retriever联系起来，我们提出了一种基于自适应和可扩展策略的从retriever中采样负例的方法。

除了通过数据监督来训练召回模型外，还可以利用基于知识蒸馏的排序模型来进行指导，因为假设排序模型在排序内容时更为精确。同时，利用采样logsoftmax目标，以便使用与排序模型相同的建议来优化召回模型。另外，从排序模型中获取知识，大多数先前的方法从排序模型的top-k结果中获取知识，但此方法耗时。

本文通过基于KL散度直接对齐排序模型和召回模型的softmax归一化预测之间的排序顺序信息，从排序模型的预测中提取排序顺序信息。为了减少优化时间成本，我们提出了一个针对KL散度的渐近无偏估计，名为采样KL散度。召回模型和排序模型通过统一的流程同时进行训练，两个模型可以相互强化。随着训练的进行，排序模型变得越来越精确，这反过来为召回模型提供了更有信息的监督信号；与此同时，随着召回模型的改进，负样例变得越来越难，这有助于提高排序模型的鉴别能力。整体框架如图所示。



Loss Function

我们优化了样本 log-softmax 的排序函数，这是 log-softmax 的无偏估计。log-softmax 的使用源于其与平均倒数排序(MRR)和标准化折扣累积增益(NDCG)的 log 的密切关系，以及其出色的推荐性能。假设在上下文 c 中正项为 k ，则关于 k 的 log-softmax 目标如下所示：

$$\begin{aligned}
 \ell(k, c) &= R_\phi(k, c) - \log \left(\sum_i \exp(R_\phi(i, c)) \right) \\
 &= -\log \left(\sum_i \exp(R_\phi(i, c) - R_\phi(k, c)) \right) \\
 &\leq -\log \left(\sum_i \mathbb{I}_{\geq 0}(R_\phi(i, c) - R_\phi(k, c)) \right) \quad \text{由于 } \exp(\cdot) \geq \mathbb{I}_{\geq 0}(\cdot), \text{ 所以不等式成} \\
 &= \log \left(\frac{1}{\text{rank}_R(k, c)} \right) = \log \text{MRR}
 \end{aligned}$$

立。

然而，优化 log-softmax 在计算上具有挑战性，因为梯度计算与物品数量呈线性关系，即。虽然对于小的物品数量，线性关系可能不是一个严重的问题，但对于非常大的物品数量，这可能是一个相当大的计算挑战。此外，由于 log-softmax 的性质，当物品数量非常大时，计算精度和数值稳定性可能会受到影响。具体来说，为了优化 log-softmax 的计算，我们可以使用一种称为“近似 softmax”的方法。

该方法基于一种观察结果：对于高概率的物品， $\log(\text{softmax}(x))$ 和 $\text{softmax}(x)$ 之间的差异非常小；而对于低概率的物品，这个差异可能更大一些。因此，我们可以使用一个简单的近似来计算 $\log(\text{softmax}(x))$ ，而不需要显式地计算 $\text{softmax}(x)$ 。这个近似基于一个 log 求和公式，它允许我们在常数时间内计算 $\log(\text{softmax}(x))$ 的近似值。具体地，我们使用以下公式进行计算：

$$\log(\text{softmax}(x)) \approx \log(\max(x)) + \log(-\log(z))$$

其中 z 是 x 中最大的元素。这个公式在计算上非常有效，因为它只需要对 x 进行一次扫描，并且不需要任何嵌套循环或条件语句。此外，它也可以很容易地并行化，因为它只需要计算常数时间的最大值和最小值。我们还发现，这个近似方法对于处理非常大或非常小的 x 值也特别有效。例如，当 x 是一个非常大的向量时，使用这个近似方法可以避免计算 $\text{softmax}(x)$ 时可能出现的溢出问题。同样地，当 x 是一个非常小的向量时，使用这个近似方法可以避免计算 $\log(x)$ 时可能出现的下溢问题。因此，这个近似方法在处理各种不同类型的数据时都非常有效。

$$\nabla \ell_\phi(k, c) = \nabla R_\phi(k, c) - \mathbb{E}_{i \sim P_\phi(\cdot|c)} [\nabla R_\phi(i, c)]$$

知乎

$$P_{\phi}(j|c) = \frac{\exp(r_{\phi}(j,c))}{\sum_{i \in \mathcal{I}} \exp(r_{\phi}(i,c))}.$$

为提高效率，我们采用重采样近似，从提议分布 $Q(\cdot|c)$ 中抽取 L 个样本 $\mathcal{S} = \{o_1, \dots, o_L\}$ 。

$$\nabla \ell_{\phi}(k, c) \approx \nabla R_{\phi}(k, c) - \sum_{i \in \mathcal{S} \cup \{k\}} w(i, c) \nabla R_{\phi}(i, c)$$

$w(i, c)$ 是一个梯度，可通过以下采样log-softmax计算得到：

$$w(i, c) = \frac{\exp(R_{\phi}(i, c) - \log \tilde{Q}(i|c))}{\sum_{j \in \mathcal{S} \cup \{k\}} \exp(R_{\phi}(j, c) - \log \tilde{Q}(j|c))}$$

$$\tilde{\ell}_{\phi}^{\mathcal{S}}(k, c) = \log \frac{\exp(R_{\phi}(k, c) - \log \tilde{Q}(k|c))}{\sum_{i \in \mathcal{S} \cup \{k\}} \exp(R_{\phi}(i, c) - \log \tilde{Q}(i|c))} \quad \text{其中} \tilde{Q}(\cdot|c) \text{代表非规范化} Q(\cdot|c).$$

Sampling from the Retriever

之前的工作将召回结果的前 K 个结果视为负样本，这样做会导致漏报的错误，因为前 K 个结果中既包括难样本又包括潜在的正样本。本文中，我们提出从召回模型 M_{θ} 构建分布，

$$Q(i|c) = P_{\theta}(i|c) = \frac{\exp(M_{\theta}(i, c)/T)}{\sum_{j \in \mathcal{I}} \exp(M_{\theta}(j, c)/T)},$$

其中系数 T 控制负样本的难度和随机性的平衡。当 $T \rightarrow \infty$ 时，[抽样方法](#)⁺完全随机；当 $T \rightarrow 0$ 时，它变成完全确定性的，相当于[argmax](#)⁺。在这种情况下，采样的logsoftmax可以被改写为

$$\tilde{\ell}_{\phi}^{\mathcal{S}}(k, c) = \log \frac{\exp(R_{\phi}(k, c) - M_{\theta}(k, c)/T)}{\sum_{i \in \mathcal{S} \cup \{k\}} \exp(R_{\phi}(i, c) - M_{\theta}(i, c)/T)}.$$

1. 计算每个物品的负采样概率。2. 根据每个物品的负采样概率从其聚类中心进行采样。3. 将采样的物品与查询向量进行比较以计算损失。

- 在第一个子空间中采样簇 k^1 。簇 k 的采样概率定义为：
- $P(k) = \frac{\psi_k \exp(\langle \mathbf{z}_c^1, \mathbf{w}_k^1 \rangle)}{\sum_{k'} \psi_{k'} \exp(\langle \mathbf{z}_c^1, \mathbf{w}_{k'}^1 \rangle)}$ 其中： $\psi_k = \sum_{k'} \omega_{k, k'} \exp(\langle \mathbf{z}_c^2, \mathbf{w}_{k'}^2 \rangle)$ 而 $\omega_{k, k'} = |\Omega_k^1 \cap \Omega_{k'}^2|$ 。
- 在给定第一个子空间中的 k^1 条件下，第二个子空间中采样集群 k^2 的采样概率表达式为：

$$P(k^2|k^1) = \frac{\omega_{k^1, k^2} \exp(\langle \mathbf{z}_c^2, \mathbf{w}_{k^2}^2 \rangle)}{\sum_{k'} \omega_{k^1, k'} \exp(\langle \mathbf{z}_c^2, \mathbf{w}_{k'}^2 \rangle)}.$$
- 在交集内容集 $\Omega_{k_1}^1 \cap \Omega_{k_2}^2$ 内，我们随机且均匀地选取一个内容。

关于采样有效性的说明：尽管被分解，但采样程序实际上是从

$$\hat{Q}(i|c) = \frac{\exp(\langle \mathbf{z}_c, \mathbf{w}_{k(i)}^1 \oplus \mathbf{w}_{k(i)}^2 \rangle)}{\sum_j \exp(\langle \mathbf{z}_c, \mathbf{w}_{k(j)}^1 \oplus \mathbf{w}_{k(j)}^2 \rangle)}$$

进行采样。由于 $\hat{Q}(i|c)$ 和 $Q(i|c)$ 之间的上界偏差，采样有效性可保证，这取决于聚类的残差。当残差小时，近似采样更接近精确采样。关于采样效率的说明：由于 $\omega_{k, k'}$ 与查询无关，可聚类后预先计算。采样 T 个内容的总体[时间复杂度](#)⁺为 $\mathcal{O}(Kd + K^2 + T)$ ，其中 d 为嵌入维度， K 为[K-means](#)⁺中的聚类数。

Training Retriever with Ranker

尽管召回模型 $M_{\theta}(i, c)$ 用于训练排序模型，但应阻止来自排序模型目标的梯度，因为召回模型仅用于为排序模型提供负面信息。随后，使用来自推荐数据的监督损失和来自排序模型的蒸馏损失来指导召回模型的训练。

Supervision Loss

提案，因为它可以通过使用少数采样项来大大降低召回模型的前向推理和反向传播成本。因此，目标函数可以表示为：

$$\min_{\theta} \sum_{i=1}^N \sum_{c=1}^C -\ln p(y_i = c | x_i, \theta)$$

其中， $p(y_i = c | x_i, \theta)$ 表示模型预测样本 (x_i, y_i) 属于类别 c 的概率， θ 表示模型参数。

$$\tilde{\ell}_{\theta}^S(k, c) = \log \frac{\exp(M_{\theta}(k, c) - \log \tilde{Q}(k|c))}{\sum_{i \in \mathcal{S} \cup \{k\}} \exp(M_{\theta}(i, c) - \log \tilde{Q}(i|c))}$$

Distillation Loss

为了提高召回模型的效率，需要限制其表达能力。虽然排序模型可能更加精确，但知识蒸馏可以提供大量弱监督信号来缓解训练数据稀疏的问题。因此，本部分提出了一种基于排序模型的知识蒸馏损失函数^{*}。我们专注于蒸馏排序模型的预测结果，而不是特定的推荐器，为了不使用先前工作中排序模型的 top-k 结果，我们遵循 KD 的开创性工作，通过 KL 散度将排序模型和召回模型的 softmax 归一化预测结果进行直接匹配。即，

$$D_{KL}(P_{\phi}(\cdot|c) \| P_{\theta}(\cdot|c)) = \sum_{i \in \mathcal{I}} P_{\phi}(i|c) \log \frac{P_{\phi}(i|c)}{P_{\theta}(i|c)},$$

然而，直接计算KL散度及其梯度并不实际，因为其内容数量会线性扩展相对于每个上下文。为此，我们提出了一种用于加速前向推理和反向传播的KL散度的无偏估计。我们用

$\mathcal{S} = \{o_1, o_2, \dots, o_L\}$ 表示从提案 $Q(\cdot|c)$ 中抽取的样本集，并定义

$$P_{\phi}^S(j|c) = \frac{\exp(\tilde{R}_{\phi}(j, c))}{\sum_{i \in \mathcal{S}} \exp(\tilde{R}_{\phi}(i, c))} \text{ 和 } P_{\theta}^S(j|c) = \frac{\exp(\tilde{M}_{\theta}(j, c))}{\sum_{i \in \mathcal{S}} \exp(\tilde{M}_{\theta}(i, c))},$$

其中 $\tilde{R}_{\phi}(i, c) = R_{\phi}(i, c) - \log \tilde{Q}(i|c)$ 和 $\tilde{M}_{\theta}(i, c) = M_{\theta}(i, c) - \log \tilde{Q}(i|c)$ 。

根据以下定理， $P_{\phi}^S(\cdot|c) \| P_{\theta}^S(\cdot|c)$ 是无偏估计 $P_{\phi}(\cdot|c) \| P_{\theta}(\cdot|c)$ 。

在用于学习召回模型时，样本KL散度基于第几节中的采样器。下面我们将讨论一些特殊情况的采样器，以便进一步理解其通用性。

Special Cases

我们研究了两种特殊的提议：

$Q(\cdot|c) = P_{\theta}(\cdot|c) = \text{softmax}(M_{\theta}(\cdot, c))$ 和 $Q(\cdot|c)$ 是均匀分布。

当 $Q(\cdot|c) = P_{\theta}(\cdot|c)$ 时，我们可以优化采样 log-softmax

$$\tilde{\ell}(k, c) = \log \frac{\exp(\Delta_k^c)}{\sum_{i \in \mathcal{S}} \exp(\Delta_i^c)}$$

来优化排序模型。其中， $\Delta_k^c = R_{\phi}(k, c) - M_{\theta}(k, c)$ 。

证明见附录。推论表明，当从 $P_{\theta}(\cdot|c)$ 中精确抽样时，最小化KL散度等同于最大化一个由排序模型和召回模型之间的softmax归一化预测差异参数化的分类分布的熵。由此可知，最小化它们之间的KL散度等同于在随机样本集 \mathcal{S} 上匹配排序模型和召回模型之间的排序顺序。

Experimental Settings

Datasets

表显示了我们在四个真实世界数据集上对本方法进行的评估。这些数据集来自不同领域和平台，在大小和稀疏性上差异很大。Gowalla数据集包含用户在不同时间的签到数据。淘宝数据集是阿里巴巴集团收集的一个大型工业数据集，包含用户的点击、购买、添加商品到购物车和商品收藏等行

1000万次交互的子集来进行实验。然后我们过滤掉了所有数据集中互动次数少于10次的用户和物品（位置/产品/电影）。对于一般的推荐任务，每个用户的行为历史被按照0.8/0.1/0.1的比例分割成训练/验证/测试集¹。对于序列推荐任务，给定一个用户的行为历史为 $(i_1, i_2, \dots, i_k, \dots, i_n)$ ，目标是利用前 k 个物品预测第 $(k + 1)$ 个物品。在所有实验中，我们为所有用户生成了训练集，其中 $k = 1, 2, \dots, n - 3$ ，我们在验证集¹和测试集中分别给出了前 $n - 2$ 个和前 $n - 1$ 个物品，并预测下一个物品。此外，在所有数据集中，我们将用户行为序列的最大长度设置为20。

Table 1: Dataset Statistics.

	#users	#items	#interactions
Amazon	9,280	6,066	158,979
Gowalla	29,859	40,989	1,027,464
MovieLens	66,958	10,682	5,857,041
Taobao	941,853	1101,236	63,721,355

Metric

评估中常用的三种Top-K度量标准是NDCG、Recall和MRR。其中，Recall@k表示目标物品在top k物品中的比例；NDCG@k对更高的排序赋予更高的权重；MRR@k表示目标物品的倒数排序的平均值。这些度量标准的值越大，表示性能越好。

Implementation Details

我们选择MF（矩阵分解）和DeepFM分别作为召回模型和排序模型进行一般推荐。在序列推荐中，我们将SASRec设置为召回模型，将DIN设置为排序模型。在预测阶段，我们首先使用召回模型从所有物品中召回出前100个物品，然后使用排序模型对这些物品进行排序，得到最终输出。同时，我们也考虑了其他召回模型和排序模型的选择，如Section所示。源代码已在github上发布。

Comparison with Baselines

Baselines Methods

通过将我们的方法与三种不同的训练算法（即ICC、RankFlow和Independent）进行比较，我们能够评估所提出的方法在独立于特定训练算法的情况下是否有效。为了实现这一目标，我们使用了具有不同训练算法的四个基线召回模型，并将所提出的方法应用于这些基线召回模型。

实验结果表明，所提出的方法在所有基线召回模型上都取得了显著的性能提升，这表明我们的方法在不同训练算法之间具有广泛的应用价值。我们还进行了定性和定量分析，以了解所提出的方法在不同应用场景下的性能表现。在定性分析中，我们发现CoRR能够产生更具相关性和多样性的结果，而其他方法在某些情况下可能会产生冗余或不相关的结果。此外，我们还发现CoRR能够更好地处理查询与数据集之间的不平衡关系。

在定量分析中，我们使用了各种不同的指标来评估所提出的方法的性能，包括准确率、召回率¹、F1分数和AUC-ROC曲线等。实验结果表明，所提出的方法在所有指标上都取得了最佳性能，这进一步证明了CoRR的有效性和优越性。总之，我们的研究表明，所提出的方法在不同应用场景下都具有广泛的应用价值，并能够有效地优化查询性能和处理各种不同的排序问题。这些优点使得CoRR成为一种有前途的图像召回方法。

Results

1: 独立训练方法在表现上优于所有召回模型，无论是在一般推荐还是在顺序推荐中。具体来说，独立方法在Recall@10、NDCG@10和MRR@10上的提升幅度分别达到5.39%、5.16%、3.50%

据中的不同信息。

2：在三种联合训练方法中，CoRR在物品推荐方面表现最佳，这表明了所提出的联合训练框架的优势。与ICC相比，其中召回模型和排序模型仅使用训练数据进行学习，CoRR在一般推荐和顺序推荐的所有数据集上，NDCG@10的平均相对提升分别为21.65%和33.35%。与RankFlow相比，其中召回模型和排序模型通过相互强化进行联合训练，CoRR在这两项任务的所有数据集上，NDCG@10的平均相对提升分别为15.72%和11.62%。

Table 2: Comparisons with baselines($\times 10^{-2}$). Δ, \blacktriangle indicate the improvements of CoRR over the best results of baselines a statistically significant for $p < 0.05, p < 0.001$ based on t test.

Dataset	Metrics	BPR	NCF	LogisticMF	DSSM	Independent	ICC	RankFlow	CoRR
Amazon	Recall@10	4.86	4.51	3.74	4.55	4.97	4.88	5.06	5.76 Δ
	NDCG@10	2.72	2.52	2.03	2.52	2.68	2.74	2.82	3.22 Δ
	MRR@10	2.51	2.29	1.80	2.28	2.43	2.37	2.63	3.04 Δ
Gowalla	Recall@10	7.78	8.33	6.70	6.30	8.24	9.08	9.35	10.52 Δ
	NDCG@10	5.74	5.69	4.21	4.32	6.17	6.18	6.34	7.49 Δ
	MRR@10	7.91	7.41	4.99	5.88	7.96	8.14	8.65	10.17 Δ
MovieLens	Recall@10	18.23	18.63	10.79	14.83	19.54	20.22	20.72	22.44 Δ
	NDCG@10	16.54	16.82	9.44	13.88	17.49	17.96	18.73	21.59 Δ
	MRR@10	25.98	26.22	15.51	22.59	27.74	27.76	28.73	33.70 Δ
TaoBao	Recall@10	0.87	1.87	0.51	1.66	2.16	2.19	2.63	2.79 Δ
	NDCG@10	0.67	0.94	0.25	0.83	1.04	1.12	1.24	1.43 Δ
	MRR@10	0.62	0.65	0.17	0.58	0.72	0.79	0.88	1.02 Δ

Dataset	Metrics	GRU4Rec	BERT4Rec	Caser	SASRec	Independent	ICC	RankFlow	CoRR
Amazon	Recall@10	4.98	5.18	4.74	5.18	5.26	4.34	5.33	5.84 Δ
	NDCG@10	2.61	2.81	2.46	2.63	2.69	2.34	2.78	3.07 Δ
	MRR@10	1.89	2.00	1.77	1.85	1.94	1.73	2.06	2.23 Δ
Gowalla	Recall@10	8.43	13.61	7.88	9.29	12.50	9.28	12.68	14.46 Δ
	NDCG@10	4.75	7.74	4.67	5.42	7.06	5.28	7.09	8.31 Δ
	MRR@10	3.64	5.95	3.69	4.24	5.41	4.07	5.82	6.44 Δ
MovieLens	Recall@10	15.54	14.67	16.78	15.02	16.89	17.27	17.59	18.46 Δ
	NDCG@10	7.78	7.52	8.55	7.53	8.97	9.23	9.33	9.83 Δ
	MRR@10	5.45	5.37	6.07	5.09	5.96	6.21	6.60	7.25 Δ
Taobao	Recall@10	0.49	0.68	0.37	0.30	0.89	0.88	0.94	1.16 Δ
	NDCG@10	0.42	0.61	0.30	0.26	0.76	0.72	0.81	1.01 Δ
	MRR@10	0.88	1.14	0.65	0.57	1.58	1.55	1.96	2.13 Δ

结果表明，CoRR的合作训练比RankFlow更有效。特别是，在CoRR中，通过来自排序模型知识蒸馏，召回模型可以得到很大改进，而在RankFlow中，召回模型很少得到改进。

Extensive Retrievers and Rankers

本文提出了CoRR，它是一个与模型无关的训练框架。为了验证CoRR的普遍有效性，我们考虑了不同的召回模型和排序模型的组合。在三个数据集上，我们进行了实验，比较了四种不同的召回模型（SASRec和Caser用于顺序推荐，MF和DSSM用于一般推荐）和四种排序模型（DIN和BST用于顺序推荐，DeepFM和DCN用于一般推荐）。我们将独立和RankFlow作为比较的基准方法。

Table 3: Extensive Retriever and Ranker. ($\times 10^{-2}$)

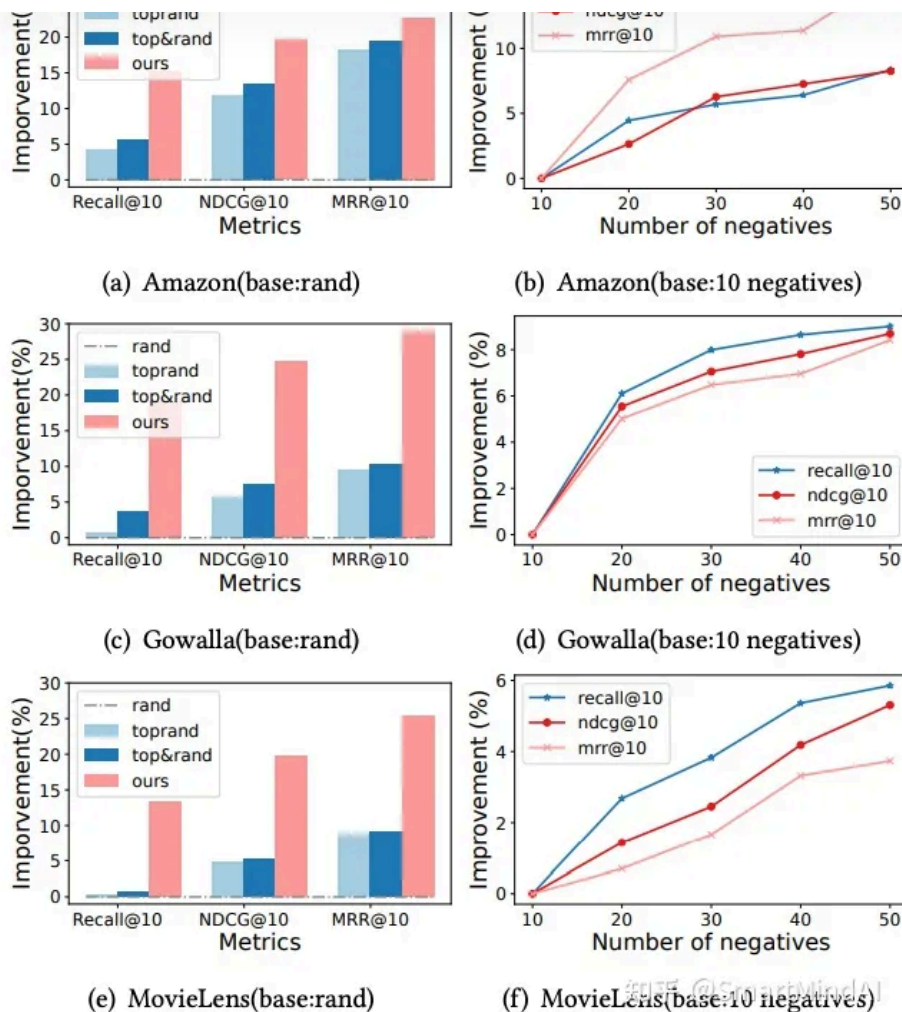
Dataset	Metric	MF+DeepFM			DSSM+DCN			SASRec+DIN			Caser+BST		
		Independent	RankFlow	CoRR	Independent	RankFlow	CoRR	Independent	RankFlow	CoRR	Independent	RankFlow	CoRR
Amazon	Recall@10	4.97	5.06	5.76	4.28	4.44	4.72	5.26	5.33	5.84	4.77	4.79	5.03
	NDCG@10	2.68	2.82	3.22	2.36	2.45	2.68	2.69	2.78	3.07	2.57	2.60	2.72
	MRR@10	2.43	2.63	3.04	2.12	2.24	2.45	1.94	2.06	2.23	1.83	1.87	1.98
Gowalla	Recall@10	8.24	9.35	10.52	7.01	8.12	9.69	12.50	12.68	14.46	10.99	10.68	13.90
	NDCG@10	6.17	6.56	7.54	5.17	6.10	6.73	7.06	7.09	8.31	6.03	6.11	8.27
	MRR@10	7.96	8.94	10.26	7.28	7.94	8.97	5.41	5.82	6.44	4.53	4.73	6.55
MovieLens	Recall@10	19.54	20.72	22.44	18.51	20.37	22.94	16.89	17.50	19.46	17.48	16.54	23.16
	NDCG@10	17.49	18.73	22.26	18.01	20.10	22.08	8.97	9.51	9.83	8.84	8.27	12.59
	MRR@10	27.84	28.73	34.07	29.02	31.53	34.24	5.96	6.60	7.25	6.24	6.22	9.39

Comparison of Different Negative Samplers

在排序模型的训练中，难负样本起着重要作用。我们在Section中比较了不同的基线策略和提出的采样策略，称为“我们的方法”。通常，召回模型的前k个召回项被视为硬负样本，因此我们比较了我们的方法与几个基于top-k的策略，以验证我们方法的有效性。它们包括

- TopRand：召回模型召回出前5n个内容，然后从召回出的内容中均匀地选出n个内容。
- Top&Rand: 召回模型召回前n/2个内容，并从所有内容中均匀抽取n/2个内容。
- 我们采用子空间聚类来近似 softmax 概率，从自适应且可扩展的采样器中采样 n 个内容。

知乎



我们发现随机性和难度在采样高质量负样本中都十分重要。在三个数据集上的实验结果显示，TopRand和Top&Rand在Rand上的表现突出，这说明难度在负采样中的重要性，因为top-k召回内容相比随机采样的内容更难且更具信息量。但要注意，Top-k召回内容也有可能是假阴性。Top&Rand在TopRand上的优势证明，在生成高质量负样本时随机性的重要性。

Effect of Knowledge Distillation

为了验证蒸馏损失在Section中的效果，我们在训练召回模型时去掉蒸馏损失，将CoRR与其变体进行比较。我们考虑这种情况下召回模型（召回模型列）和两阶段框架（CoRR列）的推荐性能。结果如表所示。结果表明，保持排序顺序的蒸馏损失确实显著提高了召回模型性能。使用采样的KL散度可以相对于NDCG@10、Recall@10和MRR@10带来10.89%、17.55%、8.18%的平均相对提升。根据表中的性能比较，CoRR中的排序顺序保持蒸馏优于RankFlow，而RankFlow中的导师学习方法效果甚微。这证明了所提出的采样KL散度对于知识蒸馏的有效性。

Dataset	Metric	w/o KL		KL	
		Retriever	CoRR	Retriever	CoRR
Amazon	R@10	4.88	4.96	5.25	5.87
	N@10	2.38	2.56	2.63	3.11
	M@10	1.84	2.14	1.90	2.27
Gowalla	R@10	13.10	13.42	13.90	14.61
	N@10	6.65	6.97	7.88	8.38
	M@10	5.69	6.00	6.05	6.49
MovieLens	R@10	15.96	17.80	16.66	18.77
	N@10	7.63	8.96	7.82	9.94
	M@10	5.69	6.60	5.31	7.28

Sensitivity to the Number of Negatives

我们提供了KL散度和logsoftmax的渐近无偏估计的理论结果，如第X和X节所述。为了进一步研究负数的影响，我们在三个数据集上进行了实验，其中负数在10, 20, 30, 40, 50之间变化。结果如图X所示右侧列。

原文《Cooperative Retriever and Ranker in Deep Recommenders》

编辑于 2024-02-19 14:02 · IP 属地北京

微软 (Microsoft)ctr检索系统



理性发言，友善互动

3 条评论

默认 最新



三杯通大道

召回模型的训练是相当于用召回模型召回的样本来训练自己吗😏

04-10 · 北京

回复 喜欢



米缸没米了

这偏论文本质不就是召回蒸馏精排模型？

2023-11-21 · 北京

回复 喜欢



Ametrin

原文：Cooperative Retriever and Ranker in Deep Recommenders

2023-11-11 · 韩国

回复 喜欢

推荐阅读

回顾微软的过去与现状：软硬件与云化启示

