

微信-2023《STEAM: 探索自校正序列推荐模型》论文解读



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

12 人赞同了该文章

Introduction

在顺序推荐中捕捉用户偏好尽可能准确的一个重要挑战是如何通过**模型化**用户的交互序列的序列依赖来实现。早期的方法基于**蒙特卡洛**(MC)方法和深度学习模型(如RNN、CNN、记忆网络、transformers和GNN)已经被广泛应用。最近,SSL也被引入到顺序推荐中以解决顺序数据的不完全性和依赖问题。然而,先前的研究忽视了这些因素可能导致的不准确性。例如,项序列可能存在两种不完美性:缺失和噪声。缺失表示某些项在序列中不存在,而噪声表示存在但未被正确识别的项。

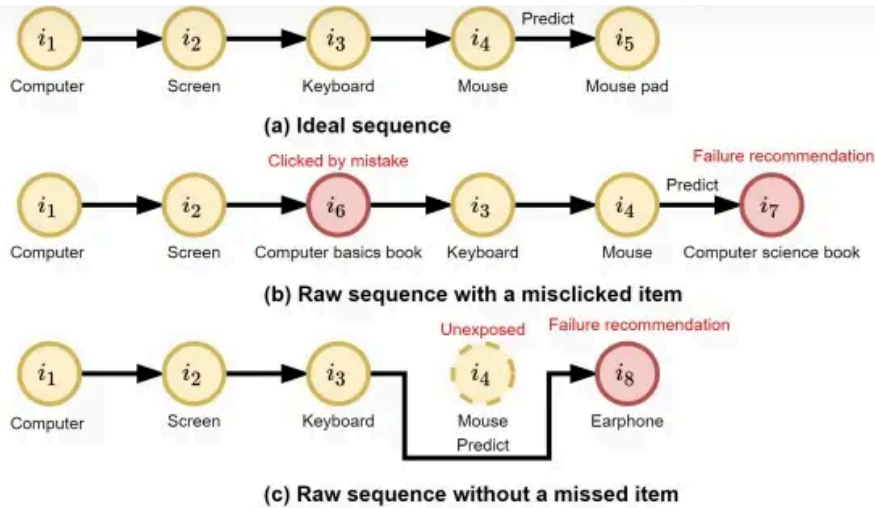


Figure 1: Examples for two kinds of imperfect item sequences. Sub-figure (a) is an ideal item sequence without any imperfection. Sub-figure (b) is an imperfect item sequence that contains a misclicked item (i.e., i_6). Sub-figure (c) is an imperfect sequence that lacks a missed item (i.e., i_4).

为了解决这个问题，提出了一种新的模型来处理序列数据的不完整性，并通过学习用户行为来捕获更复杂的转换关系。

本研究的主要成果如下：

- 提出修正原始物品序列的STEAM方法，推荐前修正原始序列⁺。
- 设计了项级校正器，并使用了删除校正和插入校正两种自我监督学习机制对其进行训练。
- 在GitHub上发布STEAM的代码和数据，展示了其最优性能，可供实验重现。

Method

Overview

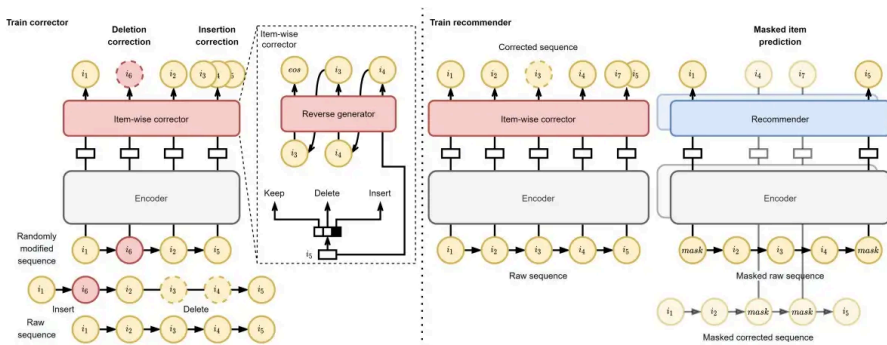


Figure 2: An overview of STEAM. For training the corrector, the item-wise corrector is asked to perform deletion correction and insertion correction on items to recover the raw item sequence that has been randomly modified. The raw item sequence with its corrected version are both used to train the recommender using the masked item prediction task. Finally, STEAM is optimized by the joint loss from the corrector and the recommender.

用 \mathcal{I} 表示物品集合， $|\mathcal{I}|$ 是该集合中的项数。用 $S = [i_1, \dots, i_{|S|}]$ 表示一个物品序列，其中 $i_t \in \mathcal{I}$ 是 S 中第 t 个位置的交互项， $|S|$ 是序列长度⁺。用 $S_{j:k}$ 表示 S 的一个子序列，其元素范围为索引区间 (j, k) 。

特别地，用 S^r 表示一个未经过处理的原始物品序列。在进行删除和插入纠正训练时，将随机修改 S^r ，然后让[STEAM]恢复它。对于 S^r 中的每个项，都以概率 p_k 保留它；以概率 p_i 在其之前插入一项；以概率 p_d 删除它。

需要注意的是，可以继续向序列添加更多项，这些项都是从 \mathcal{I} 中均匀采样得到的。将最后一步操作对应的 S^r 项存入 S_m 。为了避免混淆下一预测项，将此项保留下来。随机修改后的序列记作 S_m ，并标记其正确性校正操作序列 $O = [o_1, \dots, o_{|S_m|}]$ ，其中

$o \in \{ \langle \text{code} \rangle \text{ keep} \langle / \text{code} \rangle, \langle \text{code} \rangle \text{ delete} \langle / \text{code} \rangle, \text{'insert'} \}$ 。

其正确性校正操作为'insert'的项记作 I^{ins} 。对于每个项 $i \in I^{ins}$ ，将其正确的插入顺序记作 $S^{<i} = [i_1, \dots, i_{|S^{<i}|-1}, [eos]]$,

应在反向顺序前插入到 i ，其中 $[eos] \in \mathcal{I}$ 是一个特殊的结束符代表结束。插入后的顺序是 $[i_{|S^{<i}|-1}, \dots, i_1, i]$ 。为了训练STEAM，并用掩码后的项预测来进行掩码，随机在 S_r 或 S_c 中的一些项上应用一个特殊的结束符 $[mask] \in \mathcal{I}$ 的概率 p_m 。然后让STEAM去预测被掩码的项。被掩码的 S_r (S_c) 及其被掩码的项记作 \tilde{S}_r (\tilde{S}_c) 和 \tilde{I}_r (\tilde{I}_c)。

首先，对输入序列编码。然后，项级校正器先对所有项预测更正操作。接着，使用反向生成器⁺生成所有插入的序列。推荐器预测被遮挡的项。训练校正器时，输入随机修改的项序列，要求校正器恢复。训练推荐器时，首先使用校正器获取纠正后的项序列，然后随机在纠正后的项序列或原始项序列中掩盖一些项，使用推荐器来预测它们。最后，使用联合损失优化STEAM。在测试期间，在原始项序列或纠正后的项序列末尾附加 $[mask]$ ，使用STEAM预测下一个项。接下来提供STEAM详细信息。

Encoder

$$\mathbf{e}_t = \mathbf{E}\mathbf{i}_t,$$

对于每个物品，添加其在物品列表中的位置作为额外的特征。即：

$$\mathbf{x}_t = \mathbf{i}_t + \mathbf{e}_t \cdot \mathbf{p}_t \quad \mathbf{h}_t^0 = \mathbf{e}_t + \mathbf{p}_t,$$

给定位置嵌入 \mathbf{p}_t 和初始隐藏表示 \mathbf{h}_t^0 ，以及初始化为dropout的隐藏表示。将所有初始隐藏表示组成矩阵 \mathbf{H}_e^0 ，并通过具有 L_e 层的双向Transformer进行更新，如图所示。

$$\mathbf{H}_e^l = \text{Trm}_{\text{bi}}(\mathbf{H}_e^{l-1}),$$

双向变压器块表示为 Trm_{bi} ，其中 l 提供了详细信息。最后一层的隐藏表示矩阵 $\mathbf{H}_e^l \in \mathbb{R}^{|S| \times e}$ 是项级校正器和推荐者的输入。

Item-wise corrector

如果选择的纠正操作为插入，项目级别的校正器将使用生成器生成插入序列。根据隐藏表示 \mathbf{h}_t 在输入 \mathbf{H}_e 中的索引 i_t ，通过公式 () 获取对应纠正操作 o_t 的概率分布⁺ $P(\hat{o}_t | S)$ 。

$$P(\hat{o}_t | S) = \text{softmax}(\mathbf{W}\mathbf{h}_t),$$

$$\hat{o}_t = \text{predict}(o_t),$$

$\mathbf{W} \in \mathbb{R}^{3 \times e}$ 是投影矩阵。当测试时，从投影 \mathbf{i}_t 中选择概率最高的操作。反向生成器遵循编码器⁺中的相同步骤，获取所有项目嵌入。请注意，项目嵌入矩阵在编码器和校正器之间共享。然后，将隐藏表示 \mathbf{h}_t 与 \mathbf{H}_e 中的 t_i 以及来自 $S_{1:n-1}^{<i_t}$ 的所有项目嵌入 $\{\mathbf{e}_1, \dots, \mathbf{e}_{n-1}\}$ 堆叠在一起。

$$\mathbf{H}_c^0 = \begin{bmatrix} \mathbf{h}_t + \mathbf{p}_1 \\ \mathbf{e}_1 + \mathbf{p}_2 \\ \vdots \\ \mathbf{e}_{n-1} + \mathbf{p}_n \end{bmatrix}, \mathbf{H}_c^l = \text{Trm}_{\text{uni}}(\mathbf{H}_c^{l-1}),$$

该模型包括一个单向变压器块 Trm_{uni} ，以及用于计算插入项 i_n 概率分布的公式。隐藏表示矩阵 \mathbf{H}_c^l 和 $\mathbf{H}_c^{L_e}$ 分别第 l 层和最后一层，而 \mathbf{H}_c 则是它们的简写形式。

$$P(\hat{i}_n | S_{1:n-1}^{<i_t}, S) = \text{softmax}(\mathbf{E}^\top \mathbf{h}_n),$$

\mathbf{E} 是嵌入矩阵， \mathbf{h}_n 是隐藏表示在 \mathbf{H}_c 的最后一个位置。特别地，第一个插入项 i_1 是基于 $\mathbf{H}_c^0 = [\mathbf{h}_t + \mathbf{p}_1]$ 生成的。 \hat{i}_1 是在 $S_{1:0}^{<i_t}$, S 的基础上得到的。所以，在训练时 可以一次性使用 \mathbf{H}_c 中的所有位置的隐藏表示来计算 $P(\hat{i}_{n+1} | S_{1:n}^{<i_t}, S)$ 。在测试时，会逐个生成插入项，并在达到最大长度或遇到 $[eos]$ 为止。

目标是预测 \tilde{S}^r 或 \tilde{S}^c 中的被遮挡项并为其推荐下一个项目。给定输入序列 S 的隐藏表示矩阵 \mathbf{H}_e ，推荐者通过具有 L_r 层的双向Transformer更新 \mathbf{H}_r^0 。 $\mathbf{H}_r^l = \text{Trm}_{\text{bi}}(\mathbf{H}_r^{l-1})$,

\mathbf{H}_r^l 是一个在 $\mathbb{R}^{|S| \times e}$ 上定义的隐藏表示矩阵，其中 l 是层号。 $\mathbf{H}_r^{L_r}$ 也被称为 \mathbf{H}_r 。对于一个掩码集合中的项目 i_t ，推荐系统⁺会从 \mathbf{H}_r 中获取其隐藏表示 \mathbf{h}_t ，然后按照公式 $P(\hat{i}_t | S)$ 计算其概率分布。
 $P(\hat{i}_t | S) = \text{softmax}(\mathbf{E}^\top \mathbf{h}_t)$,

在训练和测试阶段，使用了不同的物品索引列表 \tilde{S}^r 和 \tilde{S}^c 以及 S^r 和 S^c 。通过比较这些序列表，可以计算出推荐下一个项目的概率值 $P(i_{|S^r|+1} | S^r)$ 和 $P(i_{|S^c|+1} | S^c)$ 。

Joint learning

使用STEAM模型训练时，采用三种任务：删除纠正、插入纠正和掩码项预测。首先，在原始序列 S^r 中随机插入或删除元素，得到修改后的序列 S^m ，然后让模型删除插入的元素并将其插入到 S^m 中，从而产生不需要人工标注就能修正原始输入序列的自我监督信号。这个目标可以通过最小化公式中的负对数似然来实现。

$$\begin{aligned} L_1 &= -\log P(S^r | S^m) \\ &= -\left(\log P(O | S^m) + \sum_{i \in I^{\text{ins}}} \log P(S^{<i} | S^m) \right) \\ &= -\left(\sum_{t=1}^{|S^m|} \log P(\hat{o}_t = o_t | S^m) + \sum_{i \in I^{\text{ins}}} \sum_{n=1}^{|S^{<i}|} \log P(\hat{i}_n = i_n | S_{1:n-1}^{<i}, S^m) \right), \end{aligned} \quad (10)$$

S^r 得到修正项序列 S^c ，然后随机遮罩 S^r 和 S^c 中的项并预测遮罩项。同时，最小化 \tilde{I}^r 和 \tilde{S}^r 以及 \tilde{I}^c 和 \tilde{S}^c 之间的负对数似然率，即方程 (9)

$$\begin{aligned} L_2 &= -\left(\log P(\tilde{I}^r | \tilde{S}^r) + \log P(\tilde{I}^c | \tilde{S}^c) \right) \\ &= -\left(\sum_{i \in \tilde{I}^r} \log P(\hat{i} = i | \tilde{S}^r) + \sum_{i \in \tilde{I}^c} \log P(\hat{i} = i | \tilde{S}^c) \right), \end{aligned}$$

最后，使用联合损失 L 来优化[STEAM]的参数。 $L = L_1 + L_2$ 。本文采用标准的反向传播算法⁺最小化损失函数 L 。

Experimental Setup

Datasets

评估了在三类数据集上的 STEAM，包括 Beauty 和 Sports 类别，来自 McAuley 等人 (2015 年) 的产品评论数据集。Beauty 和 Sports 分别被选作子集。使用 Yelp 数据集时，由于其规模巨大，仅选择自 2019 年 1 月 1 日至 2019 年 12 月 31 日的交易记录。数据集经过常见预处理步骤，包括删除交互次数少于 5 次的用户和项目，并将每个项目的序列划分为训练集、验证集和测试集⁺。训练集包含所有项目，除了最后一个作为测试点；验证集包含第二个最后一个项目；测试集包含其余所有项目。对于每个数据集，还构建了一个模拟测试集⁺，通过随机修改测试序列引入更多缺陷。在模拟测试集中，除了实际测试点外，保留每一项的概率为 0.8 或删除，概率分别为 0.1。此过程限制连续插入操作的数量少于 5 次，插入的项均从项集中随机抽取。处理后的数据集统计信息已在表中列出。

Dataset	#Users	#Items	#Actions	Avg. length	Sparsity
Beauty	22,362	12,101	194,682	8.7	99.93%
Sports	35,597	18,357	294,483	8.3	99.95%
Yelp	22,844	16,552	236,999	10.4	99.94%

Baselines

- 基础的序列推荐模型：
 - GRU4Rec使用GRU递归单元模拟物品间序列模式进行推荐。
 - SASRec使用单向Transformer模型预测下一个项目。
 - BERT4Rec利用双向Transformer模型进行训练，通过掩码项预测任务进行。
 - SRGNN模型利用Attention Network的Graph Neural Networks对项目序列建模。
- SSL序列推荐模型：
 - CL4SRec 通过三类自我监督任务（项目裁剪、项目遮罩和项目重新排序），基于变换器训练一个序列推荐模型。
 - DuoRec 是SSL中最先进的序列方法，通过dropout的模型级增强策略和新的采样策略生成自监督信号。

Overall performance

首先，STEAM 联合训练推荐器和项级别的纠正器，在大多数评价指标和所有数据集上均以较大优势显著优于所有基线。在 Beauty 数据集上，STEAM 的 MRR@5 和 MRR@10 性能仅略逊于最佳结果，但在几乎所有其他数据集上都表现出众。相较于单推荐模型使用的基线，STEAM 具有共享编码器和项嵌入的优点。

Table 2: Performance comparison of different methods on the real test sets. The best performance and the second best performance are denoted in bold and underlined fonts respectively. * indicates that the performance gain of STEAM against the best baseline is statistically significant based on a two-sided paired t-test with $p < 0.05$.

Model	Real Beauty				Real Sports				Real Yelp			
	HR@5	HR@10	MRR@5	MRR@10	HR@5	HR@10	MRR@5	MRR@10	HR@5	HR@10	MRR@5	MRR@10
GRU4Rec	32.95	42.59	21.63	22.90	30.58	42.85	18.35	19.97	55.40	76.57	32.23	35.05
SASRec	36.58	45.57	25.43	26.62	34.51	46.20	21.91	23.46	58.24	77.96	35.07	37.72
BERT4Rec	36.67	47.28	23.38	24.79	35.16	47.91	21.54	23.24	61.18	79.72	37.64	40.13
SRGNN	37.33	47.65	25.15	26.52	35.92	48.32	22.44	24.08	59.86	78.96	36.74	39.30
CL4SRec	39.29	48.75	27.59	28.84	37.91	49.83	24.53	26.11	62.15	80.16	39.29	41.70
DuoRec	<u>40.95</u>	<u>50.78</u>	28.84	30.15	<u>39.80</u>	<u>51.93</u>	<u>25.97</u>	<u>27.58</u>	<u>64.01</u>	<u>82.63</u>	<u>40.85</u>	<u>43.34</u>
FMLP-Rec	39.69	48.72	28.01	29.20	37.67	49.32	24.66	26.21	61.85	80.76	38.38	40.92
Recommender	35.73	46.47	22.84	24.27	35.02	47.78	21.34	23.03	61.47	77.37	37.44	40.12
STEAM	42.57*	52.89*	28.75	30.14	42.14*	55.16*	26.87*	28.61*	67.22*	84.49*	43.45*	45.77*

另一方面，纠正器通过对输入序列进行修正，可以减少缺陷，使STEAM 推荐器更准确地预测下一个项目。更详细的分析可在后续部分中找到。其次，基于SSL的模型CL4SRec和DuoRec在所有度量和数据集上都表现得更好。特别是，DuoRec实现了大多数度量的最佳性能，并且在Beauty数据集上实现了MRR@5和MRR@10的最佳性能。这证明了自监督学习通过从自身项序列中提取额外的监督信号来提高序列推荐的性能。

对比CL4SRec和DuoRec的性能发现，DuoRec明显优于CL4SRec，这验证了模型级别增强方法和在DuoRec中提出的采样策略的有效性。虽然Denoising Model FMLP-Rec是一个全MLP模型，但在处理序列数据时表现出色，优于采用更复杂架构如Transformer的vanilla模型。这是因为在Denoising Model FMLP-Rec中使用了傅立叶变换过滤掉了序列中的噪声信息，而vanilla模型可能因过度参数化而导致对噪声数据的过度拟合。相比DuoRec和CL4SRec，虽然FMLP-Rec的表现较差，但仍能与CL4SRec相媲美。通过自我监督学习，DuoRec和CL4SRec的鲁棒性*得到提高，能够在一定程度上抵抗包括噪声在内的不完美影响，使得Transformer能够更好地建模序列数据。

Benefits of the corrector

Overall-R的表现如何？相比所有基线，STEAM即使基于原始序列也显著优于其他基线。此外，发现联合学习的纠正器对于STEAM的大部分改进起到了关键作用。这是因为推荐器和纠正器共享编码器和项目嵌入。通过使用自我监督的删除和插入校正任务，项目表示和项目相关性可以转移到

表3展示了这个比例。结果显示，STEAM倾向于保留大多数序列和项目未修改。这符合预期，因为大多数项目的序列反映正常用户行为数据+而没有噪声。相比降噪方法，STEAM不仅能删除项，还能插入项，这也是其与降噪方法的主要区别。

Table 3: Performance analysis of STEAM on different groups of the real test sets. Overall-R (Overall-C) is the performance on all raw (corrected) test item sequences. Changed-R (Changed-C) is the performance on the raw (corrected) test item sequences of the changed sequence group. Unchanged is the performance on the test item sequences of the unchanged sequence group.

STEAM	Real Beauty				Real Sports				Real Yelp			
	HR@5	HR@10	MRR@5	MRR@10	HR@5	HR@10	MRR@5	MRR@10	HR@5	HR@10	MRR@5	MRR@10
Overall-R	42.21	52.75	28.27	29.68	42.03	55.04	26.75	28.48	67.19	84.49	43.42	45.75
Overall-C	42.57	52.89	28.75	30.14	42.14	55.16	26.87	28.61	67.22	84.49	43.45	45.77
Changed-R	41.35	51.59	27.04	28.40	35.04	47.64	21.56	23.23	56.05	74.19	34.36	36.80
Changed-C	42.56	52.06	28.66	29.94	35.54	48.12	22.08	23.76	57.44	74.20	34.45	37.00
Unchanged	42.58	53.25	28.79	30.22	44.21	57.36	28.37	30.12	67.44	84.71	43.62	45.95

Robustness analysis

本文进行了随机插入和删除项的实验，并通过Table展示了实验结果。结果显示，大多数基线方法在面对噪声和遗漏项时性能会下降。值得注意的是，在Beauty模拟测试集中，STEAM比DuoRec表现得更好，尤其是在处理不完美情况时。

此外，还通过对比真实测试集和模拟测试集，以及计算性能扰动距离来进一步分析各模型的鲁棒性。特别是对于STEAM，在考虑它如何处理真实测试集中的不完美情况时进行了特殊分析。大部分性能干扰值为负，表示序列不完美降低模型性能。

FMLP-Rec表现出较强鲁棒性，Yelp优于STEAM。STEAM多数Beauty/Sports优秀，少数次优Yelp。Beauty/Sports MRR@5/10 STEAM正干扰，说明STEAM除模拟不完美外，还纠正固有不完美。虽然FMLP-Rec鲁棒性优于STEAM，可能牺牲推荐性能。SASRec推荐性能好但鲁棒性较差。DuoRec鲁棒性较差但推荐性能最好。综上，STEAM推荐性能与鲁棒性间取得较好平衡，优于其它方法。

Table 5: Performance comparison of different methods on the simulated test sets.

Model	Simulated Beauty				Simulated Sports				Simulated Yelp			
	HR@5	HR@10	MRR@5	MRR@10	HR@5	HR@10	MRR@5	MRR@10	HR@5	HR@10	MRR@5	MRR@10
GRU4Rec	32.22	42.13	21.28	22.59	29.96	42.26	17.99	19.61	54.64	75.87	31.66	34.49
SASRec	35.97	45.26	24.97	26.20	33.63	45.23	21.47	23.01	57.71	77.12	34.64	37.23
BERT4Rec	35.83	46.79	22.79	24.25	34.10	46.49	20.62	22.26	59.46	78.07	36.36	38.85
SRGNN	36.64	46.81	24.50	25.85	35.39	47.55	22.00	23.60	57.55	76.82	35.09	37.68
CL4SRec	38.66	48.22	26.96	28.23	37.10	48.93	23.95	25.52	61.08	78.99	38.48	40.88
DuoRec	40.26	50.13	28.39	29.71	38.87	50.95	25.36	26.96	63.06	82.07	40.24	42.78
FMLP-Rec	39.38	48.47	27.85	29.06	37.23	48.86	24.33	25.87	61.17	80.37	37.97	40.56
Recommender	35.14	45.96	22.22	23.66	33.70	46.40	20.38	22.06	59.08	79.08	36.52	39.00
STEAM	42.09*	52.21*	28.45	29.81	41.72*	54.82*	26.43*	28.17*	66.46*	84.05*	42.83*	45.19*

发布于 2023-11-27 13:54 · IP 属地北京

微信 序列推荐 推荐系统

▲ 赞同 12 ▼ ● 2 条评论 ↗ 分享 ❤ 喜欢 ★ 收藏 📄 申请转载 ...



理性发言，友善互动

2 条评论

默认 最新



董棚

请问一下这篇文章的doi号是多少呢，我想看一下原论文，但是网上找不到资源🙏

05-14 · 黑龙江

● 回复 ❤ 喜欢



SmartMindAI 作者

A Self-Correcting Sequential Recommender

05-15 · 未知

● 回复 ❤ 喜欢

推荐阅读