

# ICSRec 论文复现

## 摘要

本次论文复现工作基于 2024 年发表在 WSDM 上的一篇文章: Intent Contrastive Learning with Cross Subsequences for Sequential Recommendation, 即交叉子序列意图对比学习用于序列推荐。该文章提出使用不同用户交互历史中的相同行为模式来构建意图监督信号, 构建一个粗粒度的意图对比学习辅助任务, 同时引入一个聚类来考虑全局的信息, 获取用户的细粒度的意图表示, 并通过构建一个细粒度意图对比学习任务来进行约束第一个任务引入的噪音问题。该文章提出的模型在使用的公开数据集上均取得了优于现有模型的表现, 本次复现工作也基本实现了文章所展示的结果, 验证了该模型的卓越性能。

**关键词:** 序列推荐; 意图表示; 对比学习

## 1 引言

推荐系统能够通过准确捕捉用户偏好并提供个性化推荐建议, 有效缓解数据过载, 其中, 序列推荐 (SR) 是动态建模用户偏好的最佳推荐模型之一, 专注于根据用户的时间行为预测用户可能交互的下一个项目。推荐系统面临数据集稀疏和噪音的问题, 这加大了对用户兴趣偏好建模的难度, 如何在有限的数据集中对用户兴趣偏好更好地进行建模是一个挑战。

用户的购买行为一般与其购买意图有关 (如购买手表用于装饰, 购买书籍用于阅读等), 利用这种意图进行序列推荐可以显著提高推荐性能和鲁棒性。不同的用户交互的序列中可能存在相同的行为模式, 这可以看作是一种粗粒度的意图监督信号, 不同用户可能出于同一目的购买了同一商品, 另外购买同一商品也可能出于不同的意图。文章通过对用户的意图进行建模, 实现了卓越的推荐性能, 在使用的公开数据集上都取得了优于现有模型的表现。

## 2 相关工作

### 2.1 序列推荐

序列推荐 (SR) 旨在根据一个人的历史交互关系预测连续的偏好。经典的 FPM-C [1] 同时融合了序列模式和用户的一般兴趣。FOSSIL [2] 通过结合基于相似性的方法和高阶马尔可夫链提高了对稀疏数据的鲁棒性。近年来 transformer [3] 的成功推动了 SR 的发展, SASRec [4] 首先利用 transformer 对用户有序的历史交互进行建模, 取得了巨大的成功。STOSA [5] 引入了随机嵌入, 并在 SR 中提出了 Wasserstein 自注意力。然而, 这些模型通常受到稀疏和噪声问题的限制。

## 2.2 基于自监督学习的序列推荐

受自监督学习 (SSL) 在自然语言处理 (NLP) 和计算机视觉 (CV) 领域的巨大成功及其在解决数据稀疏问题方面的有效性的激励, 越来越多的工作现在将 SSL 应用于推荐。BERT4Rec [6] 利用预训练的 BERT 从用户的历史交互中生成目标项目的表示。S3-Rec [7] 利用 4 个辅助自监督任务, 通过最大化不同视图的互信息来捕获序列信息。更有趣的工作将对比学习 (CL) 引入 SR 中, 利用增强视图的信息来提高学习表示的有效性, 以缓解噪声和稀疏问题。CL4SRec [8] 通过最大化同一用户按时间顺序交互的不同增强视图之间的一致性, 并同时优化与主任务的对比损失来学习用户的表示。CoSeRec [9] 通过引入两种更鲁棒的数据增强方法来生成对比对, 进一步改进了 CL4SRec。

## 2.3 意图引导的推荐系统

最近的许多方法都将注意力转向研究用户的意图, 以提高推荐系统的性能和鲁棒性。一些相关工作 [10,11] 通过引入辅助信息 (例如, 预测下一个物品的类别, 预测用户的下一个操作类型等), 构建了一个学习用户意图的补充任务。由于这些信息可能并不总是可用或真实地表达用户的意图, 因此有人提出在潜在空间中建模用户的意图。sin [12] 提出了一个稀疏兴趣提取模块, 以从大量意图组中推断用户的交互意图。ICLRec [13] 通过聚类从所有用户行为序列中提取用户的意图分布, 并通过对比 SSL 损失将学习到的意图集成到 SR 模型中。IOCRRec [14] 通过引入两个新的模块, 全局和局部模块, 来对 ICLRec 进行改进, 用于建模意图, 并利用这两个模块来缓解对比学习任务的噪声问题。然而, 这些方法都忽略了隐藏在用户交互序列中的意图监督信号。此外, 在意图建模中引入随机数据增强可能会改变隐藏在交互序列中的原始意图 [15]。相反, 所提出方法从用户交互中提取监督信号, 并利用对比学习 (CL) 来学习用户的意图。

# 3 本文方法

## 3.1 本文方法概述

ICSRec 利用交叉子序列模式构建意图监督信号进行意图表示学习。本文提出了两个模块如图 1 所示 (粗粒度意图学习 CICL 和细粒度意图学习 FICL), 从不同维度对用户意图进行建模。首先, 通过拆分操作将原始训练序列处理为多个子序列, 并将具有相同目标项的子序列放入同一集合中, 以构建粗粒度意图监督信号; ICSRec 假设具有相同目标项的两个不同子序列具有相同的意图。为此, 引入一种粗粒度意图对比学习方法, 使具有相同意图的两个子序列更加接近。为缓解同一物品在不同语境下可能代表不同意图的问题, 进一步提出细粒度意图对比学习, 使子序列更接近通过聚类得到的意图原型。最后, ICSRec 在所有数据集上与最先进的基线方法相比取得了显著的性能提升 (平均提升 24%)。

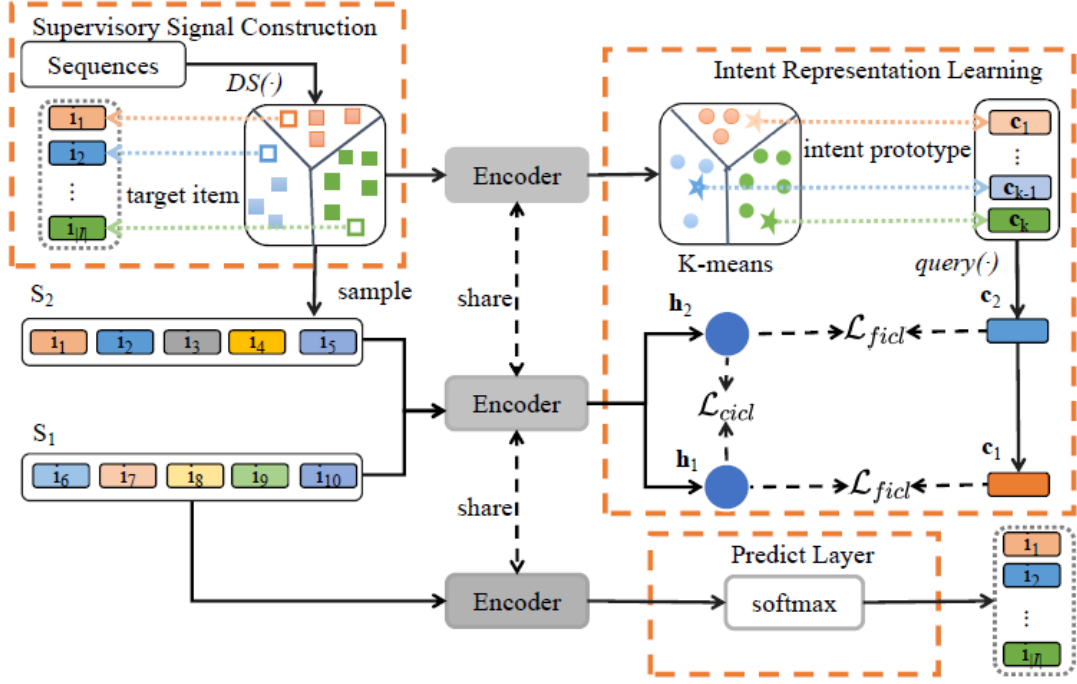


图 1. ICSRec 的模型体系结构

### 3.2 数据预处理

数据集：文章一共使用了四个公开的数据集，从 Amazon 数据集中选取 Sports, Beauty 和 Toys 作为三个不同的实验数据集。第四个数据集选择 MovieLens-1M (ML-1M) 版本作为实验数据集，其中包含 100 万个用户评分。

预处理：为了保证数据质量，出现次数少于 5 次的用户或项目将被删除。对于所有数据集，聚合每个用户的交互记录，并按操作时间戳按时间顺序对它们进行排序。出于评估的目的，根据数据集中给出的时间戳将数据分为训练、验证和测试数据集。具体来说，最后一个项目用作测试的标签，倒数第二个项目用作验证的标签，其他项目用于训练。

### 3.3 特征提取模块

首先通过  $DS(\cdot)$  操作 [16] 将原始序列拆分为多个子序列。得到的所有子序列放入不同的集合  $T = \{T_1, T_2, \dots, T_{|I|}\}$  中，其中  $T_a$  表示目标项  $i_a \in I (1 \leq a \leq |I|)$  的子序列集，以构建粗粒度的意图监督信号。

其次，将整个项集  $I$  嵌入到同一空间中，并生成项嵌入矩阵  $\mathbf{M} \in \mathbb{R}^{|I| \times d}$ ；给定输入序列  $S^u$ ，序列  $S^u$  的嵌入初始化为  $\mathbf{e}^u \in \mathbb{R}^{n \times d}$ ，可以表述如下：

$$\mathbf{e}^u = \mathbf{m}^u + \mathbf{p}^u, \quad (1)$$

其中  $\mathbf{m}^u \in \mathbb{R}^{n \times d}$  表示项的嵌入， $\mathbf{p}^u \in \mathbb{R}^{n \times d}$  表示嵌入的位置， $n$  表示序列的长度。文章选择了 SASRec [4] 作为骨干模型，表示为  $\mathbf{H}^u = f_\theta(\cdot)$ ，给定序列嵌入  $\mathbf{e}^u$ ，输出表示  $\mathbf{H}^u \in \mathbb{R}^{n \times d}$  计算如下：

$$\mathbf{H}^u = f_\theta(\mathbf{e}^u), \quad (2)$$

其中  $\theta$  表示序列模型的参数。 $H^u = [h_1^u, \dots, h_n^u]$  中的最后一个向量  $\mathbf{h}_n^u \in \mathbb{R}^d$  被选择作为序列的意图表示。

### 3.4 粗粒度意图对比学习

因为具有相同目标项的不同子序列可能代表相同的意图, 因此, 通过对比学习, 直接使具有相同目标项的两个子序列的意图表示在潜空间中更接近。给定带有相同目标项  $i_a$  的子序列  $S_1$ , 首先从  $T_a$  中随机采样一个子序列  $S_2$ , 然后分别计算两个子序列的粗粒度意图表示  $h_1$  和  $h_2$ 。然后, 利用对比学习使两个粗粒度意图表示在潜空间中相互靠近。对比损失可以表示为:

$$\mathcal{L}_{cicl} = \mathcal{L}_{con}(h_1, h_2). \quad (3)$$

而且

$$\begin{aligned} \mathcal{L}_{con}(x^1, x^2) = & -\log \frac{e^{s(x^1, x^2)/\tau}}{e^{s(x^1, x^2)/\tau} + \sum_{x \notin \mathcal{F}} e^{s(x^1, x)/\tau}} \\ & -\log \frac{e^{s(x^2, x^1)/\tau}}{e^{s(x^2, x^1)/\tau} + \sum_{x \notin \mathcal{F}} e^{s(x^2, x)/\tau}}. \end{aligned} \quad (4)$$

其中  $(x_1, x_2)$  表示一对正样本的嵌入,  $S(\cdot)$  表示内积,  $\tau$  为温度参数,  $\mathcal{F}$  为一组与 mini-batch 中两个正样本具有相同标签的负样本。

### 3.5 细粒度意图对比学习

假设用户的意图有  $K$  类型, 并且在不同环境中购买相同物品的意图可能属于不同类型。首先将所有子序列放入编码器中, 以获得所有粗粒度意图表示, 然后将所有获得的输出进行 K-means 聚类, 并将聚类中心作为细粒度的意图表示, 得到的意图原型表示为  $\mathbf{C} = \{\mathbf{c}^k\}_{k=1}^K$ , 其中  $\mathbf{c}^k \in \mathbb{R}^d$  表示  $k$ -th 意图原型, 然后通过  $\text{query}(\cdot)$  操作得到两个子序列的细粒度意图表示; 特别地, 选择意图原型集合  $\mathbf{C} = \{\mathbf{c}^k\}_{k=1}^K$  中离  $h_1$  最近的意图原型  $c_1$  和离  $h_2$  最近的意图原型  $c_2$  作为  $h_1$  和  $h_2$  的细粒度意图表示, 如下:

$$c_1, c_2 = \text{query}(h_1, \mathbf{C}), \text{query}(h_2, \mathbf{C}). \quad (5)$$

计算对比损失如下:

$$\mathcal{L}_{ficl} = \mathcal{L}_{con}(h_1, c_1) + \mathcal{L}_{con}(h_2, c_2). \quad (6)$$

### 3.6 预测层

在 SR 中, 对于一个物品的预测可以看作是一个基于所有物品集合的分类任务。因此, 使用学习到的意图表示来求计算用户可能与下一个物品交互的概率。给定学习到的意图表示  $\mathbf{h}^u \in \mathbb{R}^d$  和项目嵌入矩阵  $\mathbf{M}$ , Eq. (1) 等价于最小化交叉熵损失  $L_{Rec}$ , 如下:

$$\mathcal{L}_{Rec} = -1 * \hat{y}[g] + \log \left( \sum_i^I \exp(\hat{y}[i]) \right) \quad (7)$$

其中

$$\hat{y} = \text{softmax}(\mathbf{h}^u \mathbf{M}^\top). \quad (8)$$

其中  $\hat{y} \in \mathbb{R}^{|I|}$  表示所有项目的预测得分,  $g \in I$  表示序列的真实值。

### 3.7 总的损失函数定义

$$\mathcal{L} = \mathcal{L}_{Rec} + \lambda \cdot \mathcal{L}_{cicl} + \beta \cdot \mathcal{L}_{ficl}. \quad (9)$$

其中  $\lambda$  和  $\beta$  是需要调优的超参数。

## 4 复现细节

### 4.1 与已有开源代码对比

已有开源代码将随机种子设置为 2022，本次实验在复现时尝试将随机种子设置为 2023，最终发现结果出现了波动，与原论文相比，采用 SAS 作为编码器的实验结果大部分比原论文结果要低 0-0.5 个百分点，而采用 GRU 作为编码器的实验结果（除了 Toys 数据集）要比原论文结果高 0-0.7 个百分点。说明更换随机种子对最终实验结果有一定的影响，主要源于数据集的划分、训练时初始参数的随机初始化等因素。

### 4.2 实验环境搭建

使用 anaconda 搭建虚拟环境，按照所给开源代码中的 Readme.md 文件的要求，pip 下载安装依赖包，其中版本要求 Pytorch $\geq$ 1.7.0, Python $\geq$ 3.7, 另外 faiss-gpu 包需要用 conda 下载，其它包直接 pip 安装即可，注意 torch 需要下载对应 GPU 版本。本次复现实验在 NVIDIA GeForce RTX 4090 Ti GPU 上进行。

## 5 实验结果分析

项目主要复现结果如图2所示，最终各指标与原论文非常接近，仅相差 0-0.005，结果均比原论文要稍微低一点，可能的原因如下，第一个是随机数种子的原因，因为涉及到训练参数的随机初始化；第二个可能的原因是原作者所使用的 GPU 为阿里云服务器，GPU 型号为 NVIDIA GeForce RTX 2080 Ti，因为不同厂商搭建集群使用的策略还是略有不同，因此可能导致最终结果与原论文会有些许差异。另外，本实验还将编码器更换成 GRU，如图3所示，结果显示使用 SASRec 作为编码器效果会更好，这表明了 transformer 在建模序列任务上的优越性。

Dataset	metric	原论文结果	复现结果
Sport	HR@5	0.0403	0.0379
	HR@10	0.0565	0.0553
	HR@20	0.0794	0.0778
	NDCG@5	0.0283	0.0267
	NDCG@10	0.0335	0.0322
	NDCG@20	0.0393	0.0375
Beauty	HR@5	0.0698	0.0688
	HR@10	0.0960	0.0954
	HR@20	0.1298	0.1287
	NDCG@5	0.0494	0.0493
	NDCG@10	0.0579	0.0579
	NDCG@20	0.0663	0.0663
Toys	HR@5	0.0788	0.0778
	HR@10	0.1055	0.1030
	HR@20	0.1368	0.1348
	NDCG@5	0.0571	0.0560
	NDCG@10	0.0657	0.0641
	NDCG@20	0.0736	0.0721
ML-1M	HR@5	0.2445	0.2384
	HR@10	0.3368	0.3334
	HR@20	0.4518	0.4411
	NDCG@5	0.1710	0.1668
	NDCG@10	0.2007	0.1975
	NDCG@20	0.2297	0.2247

图 2. 复现结果与论文结果对比

Model	Dataset							
	Sports		Beauty		Toys		ML-1M	
	HR	NDCG	HR	NDCG	HR	NDCG	HR	NDCG
(A) GRU4Rec	0.0421	0.0186	0.0478	0.0186	0.0290	0.0123	0.2081	0.0834
(B) ICSRec <sub>GRU</sub>	0.0595	0.0280	0.1014	0.0506	0.0950	0.0510	0.4045	0.1964
(C) SASRec	0.0500	0.0218	0.0894	0.0386	0.0957	0.0397	0.2745	0.1156
(D) ICSRec <sub>SAS</sub>	0.0794	0.0393	0.1298	0.0663	0.1368	0.0736	0.4518	0.2297

Model	Dataset							
	Sports		Beauty		Toys		ML-1M	
	HR	NDCG	HR	NDCG	HR	NDCG	HR	NDCG
ICSRec <sub>GRU</sub>	0.0634	0.0296	0.1055	0.0515	0.0935	0.0491	0.4123	0.2006
ICSRec <sub>SAS</sub>	0.0762	0.0375	0.1287	0.0663	0.1348	0.0721	0.4411	0.2247

图 3. 采用不同编码器 GRU 和 SAS 的原论文（上）和复现结果（下）对比

## 6 总结与展望

ICSRec 是一种基于对比学习的序列推荐系统。ICSRec 从所有用户的历史交互序列中提取粗粒度的意图监督信号，然后利用这些意图监督信号构建两个辅助学习目标用于意图表示学习。该方法有助于缓解交互数据的稀疏性问题，为不同意图的用户呈现更合适的物品。最后，在 4 个真实数据集上的实验结果表明，所提模型优于其他方法。

本次实验的复现结果基本与原论文所展示的基本一致，验证了 ICSRec 模型的可行性和卓越性能。在未来的工作中，可以开发新的辅助学习对象来提高 ICSRec 的性能。此外，也可以将 ICSRec 应用于提高其他序列推荐模型的性能。

## 参考文献

- [1] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. Factorizing personalized markov chains for next-basket recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 811–820, 2010.
- [2] Ruining He and Julian McAuley. Fusing similarity models with markov chains for sparse sequential recommendation. In *2016 IEEE 16th international conference on data mining (ICDM)*, pages 191–200. IEEE, 2016.



- [3] A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.
- [4] Wang-Cheng Kang and Julian McAuley. Self-attentive sequential recommendation. In *2018 IEEE international conference on data mining (ICDM)*, pages 197–206. IEEE, 2018.
- [5] Ziwei Fan, Zhiwei Liu, Yu Wang, Alice Wang, Zahra Nazari, Lei Zheng, Hao Peng, and Philip S Yu. Sequential recommendation via stochastic self-attention. In *Proceedings of the ACM web conference 2022*, pages 2036–2047, 2022.
- [6] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM international conference on information and knowledge management*, pages 1441–1450, 2019.
- [7] Kun Zhou, Hui Wang, Wayne Xin Zhao, Yutao Zhu, Sirui Wang, Fuzheng Zhang, Zhongyuan Wang, and Ji-Rong Wen. S3-rec: Self-supervised learning for sequential recommendation with mutual information maximization. In *Proceedings of the 29th ACM international conference on information & knowledge management*, pages 1893–1902, 2020.
- [8] Xin Xia, Hongzhi Yin, Junliang Yu, Qinyong Wang, Lizhen Cui, and Xiangliang Zhang. Self-supervised hypergraph convolutional networks for session-based recommendation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 4503–4511, 2021.
- [9] Zhiwei Liu, Yongjun Chen, Jia Li, Philip S Yu, Julian McAuley, and Caiming Xiong. Contrastive self-supervised sequential recommendation with robust augmentation. *arXiv preprint arXiv:2108.06479*, 2021.
- [10] Jiacheng Li, Tong Zhao, Jin Li, Jim Chan, Christos Faloutsos, George Karypis, Soo-Min Pantel, and Julian McAuley. Coarse-to-fine sparse sequential recommendation. In *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval*, pages 2082–2086, 2022.
- [11] Md Mehrab Tanjim, Congzhe Su, Ethan Benjamin, Diane Hu, Liangjie Hong, and Julian McAuley. Attentive sequential models of latent intent for next item recommendation. In *Proceedings of The Web Conference 2020*, pages 2528–2534, 2020.
- [12] Qiaoyu Tan, Jianwei Zhang, Jiangchao Yao, Ninghao Liu, Jingren Zhou, Hongxia Yang, and Xia Hu. Sparse-interest network for sequential recommendation. In *Proceedings of the 14th ACM international conference on web search and data mining*, pages 598–606, 2021.
- [13] Yongjun Chen, Zhiwei Liu, Jia Li, Julian McAuley, and Caiming Xiong. Intent contrastive learning for sequential recommendation. In *Proceedings of the ACM Web Conference 2022*, pages 2172–2182, 2022.



- [14] Xuewei Li, Aitong Sun, Mankun Zhao, Jian Yu, Kun Zhu, Di Jin, Mei Yu, and Ruiguo Yu. Multi-intention oriented contrastive learning for sequential recommendation. In *Proceedings of the sixteenth ACM international conference on web search and data mining*, pages 411–419, 2023.
- [15] Wei Guo, Can Zhang, Zhicheng He, Jiarui Qin, Huifeng Guo, Bo Chen, Ruiming Tang, Xiuqiang He, and Rui Zhang. Miss: Multi-interest self-supervised learning framework for click-through rate prediction. In *2022 IEEE 38th international conference on data engineering (ICDE)*, pages 727–740. IEEE, 2022.
- [16] Yong Kiam Tan, Xinxing Xu, and Yong Liu. Improved recurrent neural networks for session-based recommendations. In *Proceedings of the 1st workshop on deep learning for recommender systems*, pages 17–22, 2016.