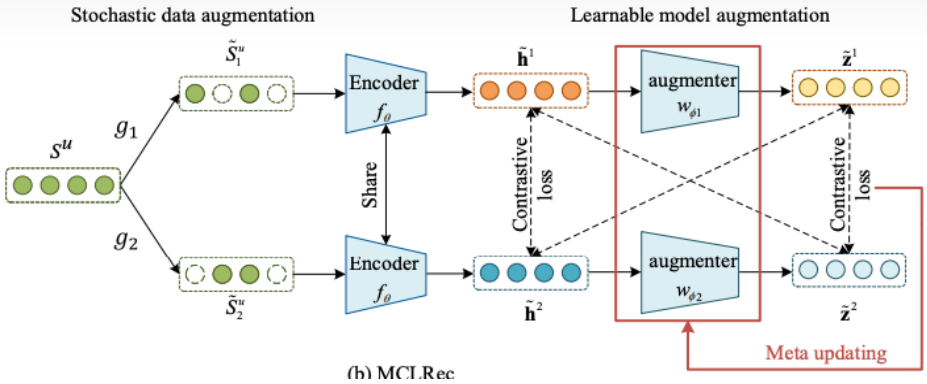
知乎



中国苏大-创新融合：元学习与对比学习在序列推荐中的深度探索

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

5 人赞同了该文章

Introduction

SR模型预测用户下一次交互项，捕捉长期和短期动态兴趣以及隐藏模式。GRU4Rec, SASRec, BERT4Rec等模型已显著提升性能，但受稀疏和嘈杂数据限制。最近，对比学习推荐系统+利用多种视角增强学习表示有效。对比学习是通过最大相似性和最小差异性增强表示的学习。选择哪种增强操作对模型性能至关重要。通常，CL推荐系统可分为三类：数据级随机掩码、裁剪和重排序；模型级丢弃；数据和模型增强结合使用。大部分是辅助任务以提高主要任务推荐准确性。

- MCLRec是通过数据增强和可学习模型来提高序列推荐性能的方法。
- MCLRec使用元优化引导学习可学习的模型增强器，以提高推荐的区分性。
- MCLRec在多个公共基准数据集上的大规模实验表明其优于当前最先进的序列方法。

Preliminaries

Problem Definition

给定用户历史交互数据，使用序列推荐方法为用户推荐下一步可能交互的物品。用户集合为$\mathcal{U}$，物品集为$\mathcal{I}$，用户u的交互物品序列为$S^u = \{i_1^u, \ldots, i_{|S^u|}^u\}$

其中$1 \leq k \leq |\mathbf{S}^u|$，目标是在用户$u$可能在第$|\mathbf{S}^u|$+步与之交互的物品集中推荐一个物品。

$\arg\max_{i \in \mathcal{I}} P(i_{|S^u|+1}^u = i | S^u)$

Sequential Recommendation Model

我们的模型主要由三部分组成：嵌入层、表示学习层和下一个项目预测层。

Embedding Layer.

给定项集$\mathcal{I}$，将其嵌入到同一空间中并生成项嵌入矩阵$\mathbf{M}$. 对于输入序列S^u，其初始嵌入是$\mathbf{e}^u$. 这个初始嵌入由$\mathbf{m}_{s_k}$和$\mathbf{p}_k$组成，其中$\mathbf{m}_{s_k}$是序列中位置k处的项的嵌入，$\mathbf{p}_k$是序列中的位置向量+，n是序列的长度。

Representation Learning Layer.

给定序列嵌入$\mathbf{e}^u$，一种深度神经网络+模型（如SASRec ）表示为$f_\theta(\cdot)$被用来学习序列的表示。其中θ代表序列模型的参数。输出表示$\mathbf{H}^u \in \mathbb{R}^{n \times d}$被计算为：

$\mathbf{H}^u = f_\theta(\mathbf{e}^u).$

知乎

$$\mathbf{H}^u = [\mathbf{h}_0^u, \mathbf{h}_1^u, \ldots, \mathbf{h}_n^u]$$

中被选择为序列的表示。

Next Item Prediction Layer.

最后的预测概率为

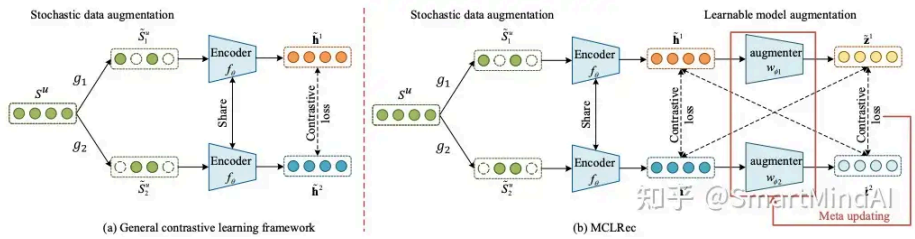
$$\hat{\mathbf{y}} = \mathbf{softmax}(\mathbf{h}_n^u \mathbf{M}^\top)$$

优化目标为最大化预测正确的概率，即使用交叉熵损失[+]。

$$\mathcal{L}_{rec} = -1 * \hat{\mathbf{y}}[g] + \log(\textstyle\sum_{i \in \mathcal{I}} \exp(\hat{\mathbf{y}}[i]))),$$

用户的真实标签为$g \in \mathcal{I}$.

Methodology



通用对比学习框架包括随机数据增强模块、用户表示编码器和对比损失函数。不同于一般的CL方法，MCLRec利用可学习增强器来选择最适合的增强方式。MCLRec的框架在图(b)中呈现，由增强模块、元学习训练策略和对比正则化组成，并将在后续部分中详细讲解这三部分的内容。

Augmentation Module

我们的增强模块由两部分组成：随机数据增强模块和学习型模型增强模块。前者从一个序列生成两个不同的增强序列，后者则捕获这些增强序列中的信息性特征。

Stochastic Data Augmentation Module.

$$S_1^u = g_1(S^u), S_2^u = g_2(S^u),\ s.t.\ g_1, g_2 \sim \mathcal{G},$$

利用函数g_1和g_2从集合$\mathcal{G}$中抽取数据增强方法，并将它们应用于不同的增强序列S_1^u和S_2^u。通过公式生成增强后的视图 $\mathbf{h}^1$和$\mathbf{h}^2$。

Learnable Model Augmentation Module.

使用两可学习增强器捕获随机增强视图中的信息特征，改进了CL框架。该方法在实验中表现出了较好的性能。

$$\mathbf{z}^1 = w_{\phi 1}(\mathbf{h}^1), \mathbf{z}^2 = w_{\phi 2}(\mathbf{h}^2),$$

对于增强器参数ϕ_1和ϕ_2，可以通过端到端的学习和适应不同的数据集。通过新的生成的$\tilde{\mathbf{z}}^1$和$\tilde{\mathbf{z}}^2$，可以得到更多的增强视图，而无需增加批大小。由于函数逼近的能力，简单的多层感知机[+]（MLP）被选为MCLRec的增强模型。未来的研究将探索其他神经网络模型，如自注意力[+]。

Meta-Learning Training Strategy

首先，引入学习型模型增强器后，需要更新两个模块：编码器和增强器。每个模块都有自己的目标，并且它们之间存在可能的差距。直接将两者参数联合作为联合学习可能会导致不理想的解决方

（即 $\mathbf{h}^1$，$\mathbf{h}^2$，$\mathbf{z}^1$ $\mathbf{z}^2$ ）

并将推荐损失结合起来，用于更新编码器$f_\theta(\cdot)$的参数。

在第二阶段，我们通过编码器$f_{\theta'}(\cdot)$重构序列，并使用增强器相关对比损失优化更新$w_{\phi 1}(\cdot)$和$w_{\phi 2}(\cdot)$。编码器

$f_\theta(\cdot)$，$w_{\phi 1}(\cdot)$和$w_{\phi 2}(\cdot)$

在迭代训练中收敛。

特别地，在第一阶段，我们随机初始化编码器$f_\theta(\cdot)$和两个增强器$w_{\phi 1}(\cdot)$和$w_{\phi 2}(\cdot)$。获取所有四个增强视图后，我们通过公式（）计算推荐损失，并使用联合对比损失更新编码器$f_\theta(\cdot)$，可以使用反向传播算法*计算。

$$\mathcal{L}_0 = \mathcal{L}_{rec} + \lambda \mathcal{L}_{cl1} + \beta \mathcal{L}_{cl2},$$

$$\mathcal{L}_{cl1} = \log \frac{\exp(-\frac{1}{2}\mathbb{E}_{\mathbf{x},\mathbf{s}}[\|\mathbf{h}^1 - \mathbf{h}^2\|^2]}{\exp(-\frac{1}{2}\mathbb{E}_{\mathbf{x},\mathbf{s}}[\|\mathbf{h}^1\|^2])}$$

$$\mathcal{L}_{cl1} = \mathcal{L}_{con}(\mathbf{h}^1, \mathbf{h}^2),$$

$$\mathcal{L}_{con}(\mathbf{x}^1, \mathbf{x}^2) = -\log \frac{e^{s(\mathbf{x}^1, \mathbf{x}^2)}}{e^{s(\mathbf{x}^1, \mathbf{x}^2)} + \sum_{\mathbf{x} \in neg} e^{s(\mathbf{x}^1, \mathbf{x})}} - \log \frac{e^{s(\mathbf{x}^2, \mathbf{x}^1)}}{e^{s(\mathbf{x}^2, \mathbf{x}^1)} + \sum_{\mathbf{x} \in neg} e^{s(\mathbf{x}^2, \mathbf{x})}},$$

$$\mathcal{L}_{cl2} = \mathcal{L}_{con}(\mathbf{z}^1, \mathbf{z}^2) + \mathcal{L}_{con}(\mathbf{h}^1, \mathbf{z}^2) + \mathcal{L}_{con}(mathbfh^2, \mathbf{z}^1)$$

固定θ，优化$w_{\phi 1}$和$w_{\phi 2}$提高编码器性能。用学习到的编码器$f_{\theta'}(\cdot)$重编码增强序列，通过方程重新计算$\mathcal{L}_{cl2}$，然后反向传播更新增强器。损失函数为：

$$\mathcal{L}_1 = \mathcal{L}_{cl2}.$$

然后我们得到了$w_{\phi_1'}(\cdot)$和$w_{\phi_2'}(\cdot)$这两个增强器，其中ϕ_1'和ϕ_2'是通过反向传播在第二阶段学习得到的参数。

- **Input:** 训练数据集 $\{S_u\}_{u=1}^{|U|}$，学习率 l 和 l'，超参数* λ, β, γ;
- **Initialize:**θ 表示编码器 $f_\theta(\cdot)$ 的参数，$\phi 1$ 表示增强器 $w_{\phi 1}(\cdot)$ 的参数，$\phi 2$ 表示增强器 $w_{\phi 2}(\cdot)$ 的参数;

$$\mathcal{L}_0 = \mathcal{L}_{rec} + \lambda \mathcal{L}_{cl1} + \beta \mathcal{L}_{cl2} + \gamma \cdot \mathcal{R}$$

$$\theta \leftarrow \theta - l \bigtriangleup_\theta \mathcal{L}_0$$

Update encoder $f_\theta(\cdot)$ by minimizing $\mathcal{L}_0$

$$\mathcal{L}_1 = \mathcal{L}_{cl2} + \gamma \cdot \mathcal{R}$$

$$\phi 1 \leftarrow \phi 1 - l' \bigtriangleup_{\phi 1} \mathcal{L}_1$$

$$\phi 2 \leftarrow \phi 2 - l' \bigtriangleup_{\phi 2} \mathcal{L}_1$$

通过优化损失函数 $\mathcal{L}_1$ 更新 $w_{\phi 1}(\cdot)$ 和 $w_{\phi 2}(\cdot)$，使得参数 $\theta, \phi 1, \phi 2$ 趋于收敛。

该元学习范式使不同维度间的差异更显著，增强了对比学习效果。编码器与增强器紧密耦合，更多信息及区别性特征促进了其有效性。实验详情见正文。

Contrastive Regularization

知乎

$s(\tilde{\mathbf{z}}^1, \tilde{\mathbf{z}}^2) = \sum_{i=1}^{D} \tilde{\mathbf{z}}_i^1 \cdot \tilde{\mathbf{z}}_i^2$

其中D是特征向量的数量。 然后我们将其分为正分组和负分组σ^+和σ^-，计算方式为：

$$\sigma^+ = \frac{1}{|\sigma^+|+|\sigma^-|} \sum_{i=1}^{|\sigma^+|} s(\tilde{\mathbf{z}}^i, \tilde{\mathbf{z}}^j)$$

$$\sigma^- = \frac{1}{|\sigma^+|+|\sigma^-|} \sum_{i=1}^{|\sigma^-|} s(\tilde{\mathbf{z}}^i, \tilde{\mathbf{z}}^j)$$

$$\sigma^+, \sigma^- = contrast(\mathbf{z}^1, \mathbf{z}^2)$$

$$o_{min} = \min(\{\min(\sigma^+), \max(\sigma^-)\}),$$
$$o_{max} = \max(\{\min(\sigma^+), \max(\sigma^-)\}),$$

$$\mathcal{R} = \frac{1}{|\sigma^+|} \sum([\sigma^+ - o_{min}]_+) + \frac{1}{|\sigma^-|} \sum([o_{max} - \sigma^-]_+),$$

$|\sigma^+|$和$|\sigma^-|$分别表示正样本和负样本的数量。$[\cdot]_+$是零阈值函数，定义为 $[a]_+ = \max(a, 0)$。

该式可重写为：

$$\frac{|\sigma^+|}{|\sigma^+|+|\sigma^-|} = \frac{[|X-\mu|]_+}{|X-\mu|+[1-|X-\mu|]_+}$$

$$\mathcal{L}_0 = \mathcal{L}_{rec} + \lambda\mathcal{L}_{cl1} + \beta\mathcal{L}_{cl2} + \gamma\mathcal{R}.$$

将方程重写为：

$$\mathcal{L}_1 = \mathcal{L}_{cl2} + \gamma\mathcal{R},$$

γ是一个权重，用于平衡对比性正则化与其他损失。训练过程由算法详述。

Algorithm 1 The MCLRec Algorithm

Input: Training dataset $\{S_u\}_{u=1}^{|U|}$, learning rate l and l', hyper-parameters λ, β, γ;
Initialize: θ for encoder $f_\theta(\cdot)$, $\phi 1$ for augmenter $w_{\phi 1}(\cdot)$ and $\phi 2$ for augmenter $w_{\phi 2}(\cdot)$;

repeat
  for t-th training iteration **do**
    $\mathcal{L}_0 = \mathcal{L}_{rec} + \lambda\mathcal{L}_{cl1} + \beta\mathcal{L}_{cl2} + \gamma \cdot \mathcal{R}$
    $\theta \leftarrow \theta - l \triangle_\theta \mathcal{L}_0$
    Update encoder $f_\theta(\cdot)$ by minimizing $\mathcal{L}_0$
  end for
  for t-th training iteration **do**
    $\mathcal{L}_1 = \mathcal{L}_{cl2} + \gamma \cdot \mathcal{R}$
    $\phi 1 \leftarrow \phi 1 - l' \triangle_{\phi 1} \mathcal{L}_1$
    $\phi 2 \leftarrow \phi 2 - l' \triangle_{\phi 2} \mathcal{L}_1$
    Update $w_{\phi 1}(\cdot)$ and $w_{\phi 2}(\cdot)$ by minimizing $\mathcal{L}_1$
  end for
until $\theta, \phi 1, \phi 2$ converge

Discussion

Connections with Contrastive SSL in SR

知乎

Time Complexity Analysis of MCLRec

我们模型的复杂性源自于训练和测试阶段的多任务学习优化θ,ϕ_1,ϕ_2。在训练过程中，我们使用了四个优化目标来优化网络f_θ，导致总的时间复杂度*为

$$\mathcal{O}(|\mathcal{U}|^2 d + |\mathcal{U}|d^2)$$

对于第二阶段，我们有两个优化目标来优化增强器，导致总的时间复杂度为$\mathcal{O}(|\mathcal{U}|d^2)$。总体复杂度主要由$\mathcal{O}(|\mathcal{U}|^2 d)$主导，其中$|\mathcal{U}|$代表用户数量。在测试阶段，我们不需要使用增强器和对比性目标，因此模型的计算效率与最先进的对比性SR模型相同，即$(\mathcal{O}(d|\mathcal{I}|))$，其中$|\mathcal{I}|$表示物品的数量。

Experimental Settings

Datasets.

验证方法有效性，使用3个真实世界数据集评估模型。

Datasets	#users	#items	#actions	avg.length	sparsity
Sports	35598	18357	296337	8.3	99.95%
Beauty	22363	12101	198502	8.8	99.93%
Yelp	30431	20033	316354	10.4	99.95%

Baseline Methods.

- 本文比较了BERT4Rec、S^3-Rec、S^3-Rec$_{MIP}$、CL4SRec、CoSeRec、LMA4Rec、ICLRec、DuoRec和SRMA五种推荐系统模型。
- BERT4Rec使用深度双向自注意力；
- S^3-Rec使用自我监督学习；
- S^3-Rec$_{MIP}$只使用了掩码项预测任务进行训练；
- CL4SRec首次使用了数据增强和对比学习；
- CoSeRec提出了两种数据增强方法；
- LMA4Rec通过引入可学习伯努利丢弃改进CoSeRec；
- ICLRec通过聚类学习用户行为并将其集成到模型中；
- DuoRec提出一种采样策略和dropout进行模型级增强；
- SRMA引入三种模型增强方法并将它们与数据增强相结合，构造视对。

Evaluation Metrics.

将数据集按时间戳划分为训练、验证和测试集，最后一部分用于测试，倒数第二到倒数一部分用于验证，剩余部分用于训练。未使用负采样在整个项目集上排名。使用HR\@k和NDCG\@k两种常用的评估指标，其中k=5,10,20。HR指标关注事实真相是否出现在前五名，NDCG指标考虑的是排名位置的敏感性。

Implementation Details.

我们实现了Caser、S^3-Rec、BERT4Rec、CoSeRec、LMA4Rec、ICLRec、DuoRec和SRMA。BPR、GRU4Rec、SASRec和CL4SRec则是基于公开资源实现的。我们在验证数据上的性能选择最佳参数设置，如使用的模型、参数数量等。MCLRec的编码器采用transformer，自注意力块数量和注意力头数量分别为2。增强器是一个3层全连接MLP网络。我们设置d为64，n为50，学习率为0.001，l'为0.001，批大小为256。我们从
$\{0.01, 0.02, 0.03, 0.04, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5\}$中选择$\lambda$和$\beta$，$\gamma$等于$\beta$的0.1倍。整个模型使用Adam优化器进行优化。我们使用早停策略根据验证数据的表现训练模型。所有实验在NVIDIA GeForce RTX 2080 Ti GPU上进行。

Overall Performances (RQ1)

式训练，相比BERT4Rec和S^3-Rec_{MIP}整体表现更好。这表明对比学习方法通过最大化互信息生成更多表达性用户和项目嵌入，从而得到更表达性的嵌入。

- 引入模型增强能提升性能，DuorRec在所有数据集表现优秀，比其他基线好。DuorRec使用SSL数据增强和随机模型增强，显著提高性能。这鼓励我们在CL训练中结合这两种增强方法。
- (1) 我们的模块能学习适应性强的增强表示。(2) 元学习有效提高推荐准确度。结果支持对比式推荐框架使模型学习更丰富信息。

Dataset	Metric	BPR	GRU4Rec	Caser	SASRec	BERT4Rec	S^3-Rec_{MIP}	CL4SRec	CoSeRec	LMA4Rec	ICLRec	DuoRec	SRMA	MCLRec	Improv.
Sports	HR@5	0.0123	0.0162	0.0154	0.0214	0.0217	0.0121	0.0231	0.0290	0.0297	0.0290	0.0312	0.0299	**0.0328**	5.13%
	HR@10	0.0215	0.0258	0.0261	0.0333	0.0359	0.0205	0.0369	0.0439	0.0439	0.0437	0.0466	0.0447	**0.0501**	7.51%
	HR@20	0.0369	0.0421	0.0399	0.0500	0.0604	0.0344	0.0557	0.0636	0.0634	0.0646	0.0696	0.0649	**0.0734**	5.46%
	NDCG@5	0.0076	0.0103	0.0114	0.0144	0.0143	0.0084	0.0146	0.0196	0.0197	0.0191	0.0192	0.0199	**0.0204**	2.51%
	NDCG@10	0.0105	0.0142	0.0135	0.0177	0.0190	0.0111	0.0191	0.0244	0.0245	0.0238	0.0244	0.0246	**0.0260**	5.69%
	NDCG@20	0.0144	0.0186	0.0178	0.0224	0.0251	0.0146	0.0238	0.0293	0.0293	0.0291	0.0302	0.0297	**0.0319**	5.63%
Beauty	HR@5	0.0178	0.0180	0.0251	0.0377	0.0360	0.0189	0.0401	0.0504	0.0511	0.0500	0.0559	0.0503	**0.0581**	3.94%
	HR@10	0.0296	0.0284	0.0342	0.0624	0.0601	0.0307	0.0642	0.0725	0.0735	0.0744	0.0825	0.0724	**0.0871**	5.58%
	HR@20	0.0474	0.0427	0.0643	0.0894	0.0984	0.0487	0.0974	0.1034	0.1047	0.1058	0.1193	0.1025	**0.1243**	4.19%
	NDCG@5	0.0109	0.0116	0.0145	0.0241	0.0216	0.0115	0.0268	0.0339	0.0342	0.0326	0.0340	0.0318	**0.0352**	2.92%
	NDCG@10	0.0147	0.0150	0.0226	0.0342	0.0300	0.0153	0.0345	0.0410	0.0414	0.0403	0.0425	0.0398	**0.0446**	4.94%
	NDCG@20	0.0192	0.0186	0.0298	0.0386	0.0391	0.0198	0.0428	0.0487	0.0493	0.0483	0.0518	0.0474	**0.0539**	4.05%
Yelp	HR@5	0.0127	0.0152	0.0142	0.0160	0.0196	0.0101	0.0227	0.0241	0.0233	0.0239	0.0429	0.0243	**0.0454**	5.83%
	HR@10	0.0216	0.0248	0.0254	0.0260	0.0339	0.0176	0.0384	0.0395	0.0387	0.0409	0.0614	0.0395	**0.0647**	5.37%
	HR@20	0.0346	0.0371	0.0406	0.0443	0.0564	0.0314	0.0623	0.0649	0.0636	0.0659	0.0868	0.0646	**0.0941**	8.41%
	NDCG@5	0.0082	0.0091	0.0080	0.0101	0.0121	0.0068	0.0143	0.0151	0.0147	0.0152	0.0254	0.0154	**0.0332**	2.47%
	NDCG@10	0.0111	0.0124	0.0113	0.0133	0.0167	0.0092	0.0194	0.0205	0.0196	0.0207	0.0383	0.0207	**0.0334**	2.87%
	NDCG@20	0.0143	0.0145	0.0156	0.0179	0.0223	0.0127	0.0254	0.0263	0.0258	0.0270	0.0447	0.0266	**0.0467**	4.47%

Ablation Study (RQ2)

Model	Dataset					
	Sports		Beauty		Yelp	
	HR	NDCG	HR	NDCG	HR	NDCG
(A) MCLRec	**0.0734**	**0.0319**	**0.1243**	**0.0539**	**0.0941**	**0.0467**
(B) w/o $\mathcal{L}_{cl1}$	0.0705	0.0299	**0.1243**	**0.0539**	0.0918	0.0462
(C) w/o $\mathcal{L}_{cl2}$	0.0557	0.0238	0.1056	0.0394	0.0623	0.0254
(D) w/o $\mathcal{R}$	0.0691	0.0291	0.1236	0.0529	0.0873	0.0445
(E) share	0.0707	0.0299	0.1231	0.0532	0.0923	0.0456

为了评估模型各部分的有效性，我们在 Table 中分别针对MCLRec 的 HR\@20 和 NDCG\@20 性能进行了几种删除实验。具体来说，w/o 表示没有；

(A) 是 MCLRec的模型；

(B) 是通过将 λ 设置为0 来删除 $\mathcal{L}_{cl1}$ 的模型；

(C) 是与 CL4SRec 等效的 $\mathcal{L}_{cl2}$ 模型；

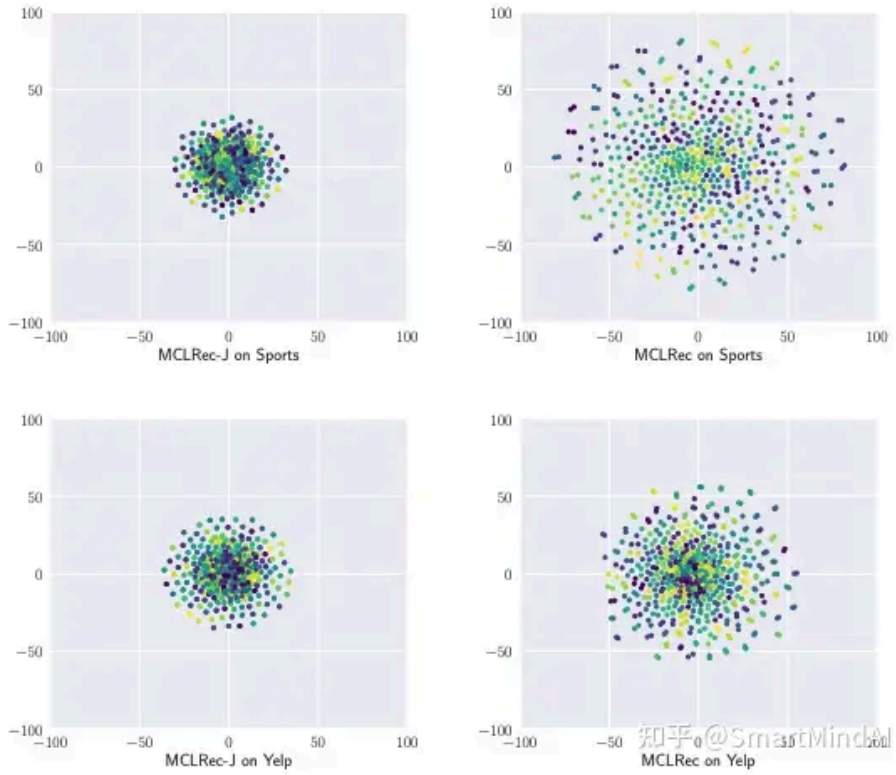
(D) 是对比正则化组件的模型；

(E) 是参数共享的两个增强器的模型。 从这个表中我们可以看出，MCLRec 在所有数据集上都表现得最好，这说明我们的框架的所有组成部分都是有效的，而且元优化的对比学习增强了模型学习更具表达性的表示的能力。 通过比较 (A) 和 (C) 及 (D)，我们可以发现可学习的模型增强和对比正则化可以显著提高模型精度，这与我们的陈述相一致。 通过比较 (B) 和 (C)，我们可以观察到可学习的数据增强比随机数据增强更有效。 通过比较 (A) 和 (B)，可以看出数据和模型级别增强的组合可以进一步提高模型性能。 通过比较 (A) 和 (E)，我们可以发现共享参数的增强器会降低结果。这可能是由于使用相同的增强器可能会进一步导致学习增强观点的高度相似性，从而导致性能下降的事实。 最后，根据 Table 显示，在删除正则项$^+$后，我们模型在三个数据集上的性能均有所下降，这表明了正则项的有效性。

为了分析正规项对模型的影响，我们通过TSNE可视化学习得到的增强视图

$\tilde{\mathbf{z}}^1$和$\tilde{\mathbf{z}}^2$。

简化后，我们不带$\mathcal{R}$表示为w/o，带有$\mathcal{R}$表示为w。我们用w/o$\mathcal{R}$和w$\mathcal{R}$训练模型各进行300个epoch$^+$的端到端训练，然后利用TSNE将增强嵌入到二维空间中。因为篇幅限制，Sports和Yelp的结果在图中展示。 我们发现在w/o$\mathcal{R}$情况下，增强器能学会压缩的视图表示，即正面和负面之间的表达分散；而在w$\mathcal{R}$情况下，增强器能学会更具有鉴别性的特征，即正面与负面之间足够"接近"，但负面相对"远离"。这进一步证实了正规项的有效性。此外，我们发现仅在其他模型（如



Effectiveness of Meta Optimization (RQ3)

我们通过对比MCLRec与其他方法（如MCLRec-J），以及可视化增强视图$\mathbf{h}^1$和$\mathbf{h}^2$，验证了元优化在MCLRec中的有效性。结果显示，元学习使得MCLRec-J在大多数情况下都能取得更好的效果，而MCLRec则得益于元学习带来的更丰富的判别增强视图。

我们使用联合学习和元学习策略训练模型，然后使用TSNE将增强的嵌入向二维空间减少。体育和Yelp的结果在图中给出。我们发现MCLRec生成的负对表示更分散且正对表示更接近，表明元学习策略有助于避免结果崩溃并产生更多有意义的表示。这是因为编码器和增强器需要更新的参数不同以及目标对象不同，导致两个对象之间的可能差距，直接联合学习可能会导致两个模块性能下降，结果不佳。

Further Analysis (RQ4)

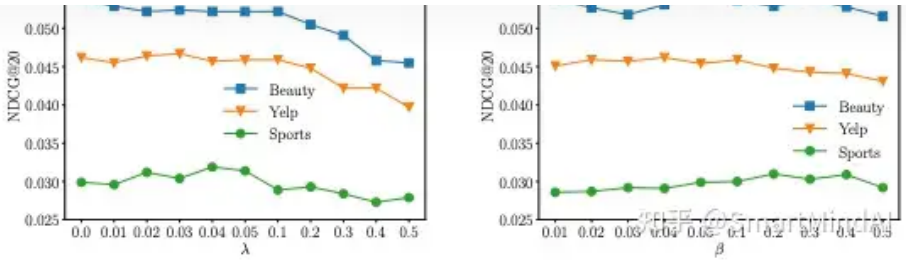
实验验证MCLRec的鲁棒性[+]。仅改变一个变量，保持其他最优超参数，使用Sports和Yelp数据集进行实验。

Impact of Batch Size.

图示显示减小批大小会导致所有模型性能降低。添加一个自监督辅助任务能显著提升模型在不同批大小下的性能。尤其值得注意的是，MCLRec在64批大小下的性能优于256批大小的所有其他模型，并在体育和Yelp上表现最佳。这表明，相较于CI4SRec，我们的方法能在不使用大批次的情况下达到较好的效果。这是因为引入可学习的模型增强使得对比学习能够在包含$\mathbf{h}^1$和$\mathbf{h}^2$以及$\mathbf{z}^1$和$\mathbf{z}^2$等更具信息性的增强视图进行训练。

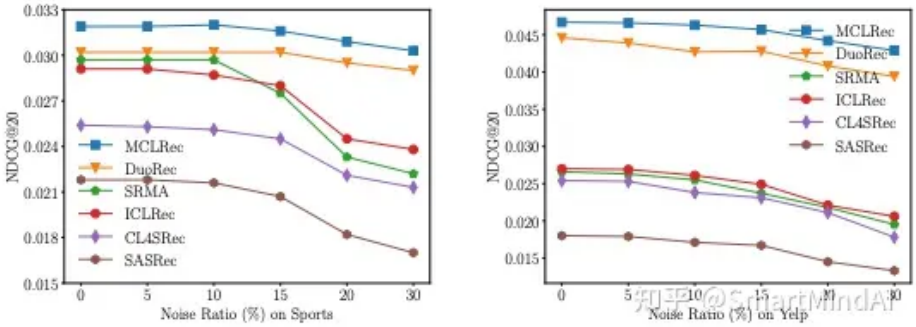
Hyper-parameters Analysis.

最后的MCLRec损失函数在方程中是一个多任务学习损失。图显示了为β和λ分配不同的权重对模型的影响。我们观察到MCLRec在不同的β和λ处性能达到峰值，这表明提出的框架的有效性，并表明引入合适的权重可以提高推荐系统的效果。从这些图中，对于Sports，$\beta = 0.4$和$\lambda = 0.04$；对于Beauty，$\beta = 0.05$和$\lambda = 0$；对于Yelp，$\beta = 0.1$和$\lambda = 0.03$通常是MCLRec的合适设置。$\mathcal{L}_{cl2}$，即β的权重通常大于λ，这表明可学习的模型增强通常比随机数据增强更重要。



Robustness to Noise Data

利用元训练方法和正则化项，我们的模型能有效地从随机增强的视角学习对比学习所需的有效表示。



Robustness w.r.t. User Interaction Frequency.

对于MCLRec在稀疏数据下的鲁棒性，我们将其分为三个组并保持总行为序列数不变。在训练和评估阶段，我们将所有的模型都独立地应用于每个用户组。根据图表，我们可以看到降低交互频率会导致所有模型的性能下降。 通过比较MCLRec与SASRec和CL4SRec，我们可以发现MCLRec在所有用户组中的表现均优于SASRec和CL4SRec。这意味着MCLRec能够通过引入更多的对比学习增强特征来更好地处理数据稀疏问题，并持续提高嵌入表示的学习效果，即使历史交互数量有限也一样。

DataSets	Sports			Yelp		
#length	=5	6-8	>8	=5	6-8	>8
#users	11416	14209	9973	8076	11109	11246
#items	18357	18357	18357	20032	20030	20033
#actions	57080	95564	143693	40380	75082	200892
sparsity	99.97%	99.96%	99.92%	99.98%	99.97%	99.91%

Conclusion

本文提出了一种名为MCLRec的新颖对比学习模型，用于序列推荐。该模型利用对比学习数据和可学习模型增强以获取更具有信息性和区分性的特征。通过应用元学习，模型可以更新其参数以反映编码器表现。广泛的实验结果显示，我们的方法优于现有对比学习序列推荐模型。此外，由于我们框架的一般化，未来MCLRec可用于多种其他推荐模型，从而进一步提升性能。

论文原文《Meta-optimized Contrastive Learning for Sequential Recommendation》

编辑于 2024-01-26 10:12 · IP 属地北京

序列推荐 meta-learning 对比学习