

稀疏增强网络：顺序推荐鲁棒增强的对抗生成方法

摘要

顺序推荐在日常推荐系统中发挥着重要的作用，例如亚马逊和淘宝等电子商务平台。然而，即使有了大模型的出现，由于新用户的加入或新产品的推出，这些平台在个人用户的历史浏览记录中往往会面临稀疏的问题。因此，现有的序列推荐算法可能表现不佳。为了解决这个问题，基于序列的数据增强方法引起了人们的关注。现有的序列增强方法通常依赖于对现有数据的增强，采用裁剪、屏蔽预测、随机重排序和随机替换原始序列等技术。虽然这些方法已经显示出了改进，但它们往往忽略了对序列深度嵌入空间的探索。为了解决这些挑战，提出了一种稀疏增强网络 (SparseEnNet)，这是一种鲁棒的对抗生成方法。SparseEnNet 旨在充分探索序列推荐中的隐藏空间，生成更健壮的增强项。此外，采用了一种对抗生成方法，允许模型区分数据增强类别，并对序列中的下一个项目实现更好的预测性能。实验表明，在对现实世界数据集进行评估时，本文的方法比现有方法实现了 4-14% 的显著改进。

关键词：顺序推荐；推荐算法；稀疏增强网络

1 引言

顺序推荐 (SR) 的目的是根据用户的历史行为来理解他们不断变化的偏好，从而促进对他们即将到来的物品偏好的精确预测 [4] [12] [15] [22]。SR 在预测用户兴趣方面的有效性引起了人们的极大兴趣。

例如，FPMC (Rendle, Freudenthaler, and Schmidt-Thieme 2010) 协同矩阵分解和马尔可夫链方法，通过从有限的观察中为用户导出个性化的过渡矩阵来增强推荐系统，在顺序篮数据中优于其他模型。此外，ICL [4] 从未标记的行为序列中开发出用户意图分布函数，并通过整合学习到的意图来优化具有对比自监督学习的 SR 模型。尽管如此，在用户历史序列较短且稀疏的场景中，这些方法很难有效地解决下一项预测任务。

为了应对这一挑战，最近的研究引入了创新的方法。CoSeRec [15] 探索了对比自监督学习的使用，使用信息增强算子来创建精细视图，有效地解决了数据稀疏性和噪声数据等问题。此外，CL4SRec [22] 引入了三种数据增强技术 (裁剪/掩码/重新排序)，将用户交互序列投射到不同的角度，增强了对优秀用户表示的学习。然而，之前的方法主要集中在增强原始序列的表层，忽略了序列隐藏空间内的互联性。

为了说明这个问题，在图 1 中提供了一个示例，其中子图中的每个顶部行表示使用数据增强方法生成的附加序列，而相应的原始序列显示在底部行。该图演示了以下场景：(a) 从原始序列中随机选择从位置 v_2 开始的连续子序列 (b) 序列内多个项目的随机掩码；(c) 子序列

的随机洗牌。此外，还有一些替代方法，比如在序列中替换或插入各种项目。总的来说，将上述方法称为浅层数据增强。

为了进一步全面探索序列推荐中的潜在空间，整合了前面提到的数据增强技术，并引入了一种名为 SparseEnNet 的新型对抗生成方法，如图 1 (d) 所示。在这种方法中，处理项目 v3, v4, 和 v5 通过专门设计的编码器产生虚拟物品嵌入，从而增强原始序列。然而，观察到过度使用数据增强方法可能会阻碍原始模型的预测准确性，特别是对于固有包含丰富历史信息的较长序列。这种过度的增强可能会导致次优结果。

因此，为了提高序列推荐性能的鲁棒性，在提出的 SparseEnNet 中设计了四个主要组件。更具体地说，首先利用增强鉴别器来区分数据增强类别，并对序列中的下一个项目实现更好的预测性能。然后，采用稳定性鉴别器来稳定由相同增强操作生成的项目嵌入。之后，采用负样本学习方法来最大化两个正对之间的互信息，同时有效地增加与负项的距离。在不损失通用性的情况下，使用 Transformer [19] 作为序列编码器来预测下一个项目。最后，使用自训练增强学习使用序列编码器对所有原始序列进行编码，然后聚合序列表示以形成一组序列表示，以捕获相似序列之间的一致性。请注意，所有提到的编码器都是作为生成器的组部分。工作贡献可以总结如下：

(1) 提出了一种鲁棒的对抗生成方法，称为 SparseEnNet，它可以通过生成更鲁棒的增强项来充分探索序列推荐中的隐藏空间。

(2) 设计了四个主要组件：一个增强鉴别器、一个稳定性鉴别器、一个负样本学习模块和一个自训练增强学习模块来实现上述目的。在三个广泛使用的数据集上进行了大量的实验，以证明我们的方法在几个基线上的优越性。

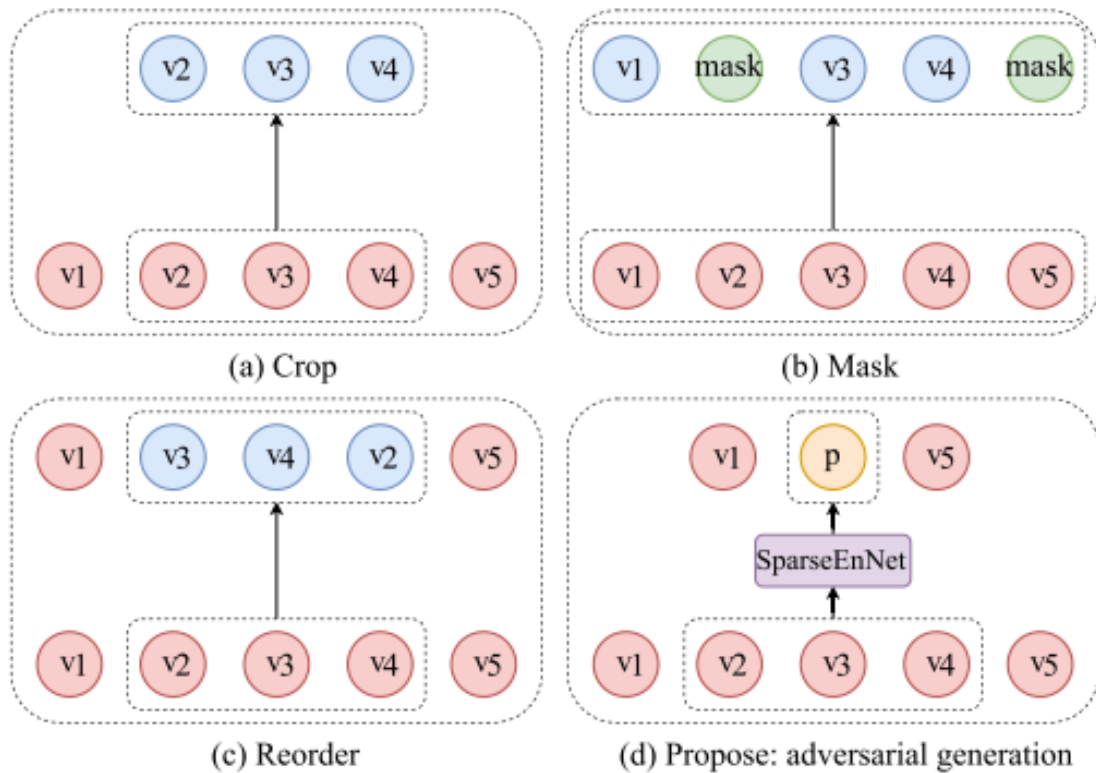


图 1. 数据增强技术的例子

2 相关工作

2.1 SR 中的对比学习

顺序推荐 (SR) 的早期研究通常依赖于马尔可夫链 (MC) 来捕获项目之间的顺序相关性 [25]。随着深度学习的发展, 各种深度学习架构被纳入到序列推荐任务中。例如, [11] 将门控循环单元 (GRU) 集成到序列推荐中。此外, [14] [13] [16] 利用注意机制提取上下文信息, 导致更有希望的结果。在此之后, 对比学习 (CL), 一种自我监督的任务, 在各个领域都引起了极大的关注, 包括计算机视觉 (CV) [3] [9] 和自然语言处理 (NLP) [6]。最近, 对比学习也在推荐领域发现了效用, 增强了顺序推荐模型的性能。例如, [23] 提出了一种基于协同过滤的推荐方法, 该方法采用基于 dnn 的对比学习来增强项目特征。此外, 一些研究已经展示了在图神经网络中使用对比学习来增强推荐性能的潜力 [20] [21]。在顺序推荐领域, S3-Rec [24] 采用对比学习进行预训练, 以提高项目表征。相反, CL4SRec [22] 将 CL 纳入多任务学习框架, 将对比学习与顺序推荐任务协同起来, 以提高性能。此外, ICLRec [4] 引入了意图对比自监督学习来捕捉用户意图。

上面提到的顺序推荐方法通常依赖于广泛的历史用户-项目交互 [2]。尽管一些早期的方法 [22] [4] 集成了数据增强技术, 但它们往往缺乏对序列底层嵌入空间的深入探索。因此, 它们可能无法有效地生成更具弹性的项目嵌入。

2.2 SR 中的对抗性学习

对抗学习最初是由生成对抗网络 (GAN) [8] 引入的, 它通过在生成器和鉴别器之间进行极大极小博弈来增强模型性能。这种方法在领域自适应 [7] 和异常检测 [1] 等领域得到了广泛的应用。在推荐领域, 对抗性方法已被用于增强基础模型的性能。例如, 对抗性个性化排名 (APR) [10] 采用对抗性训练来增强贝叶斯个性化排名 (BPR) 的鲁棒性。在序列推荐中, MFGAN [17] 采用多因素判别器来评估编码器生成的推荐, 有效地解耦了各种推荐因素。与之前的模型相比, 本文的方法在序列推荐中将对抗生成纳入数据增强中。这种集成使模型能够区分数据增强类别, 最终提高序列中后续项目的预测性能。

3 本文方法

3.1 问题定义

定义顺序推荐如下: 给定一个用户-物品交互序列 $s^u = [v_1^u, \dots, v_t^u, \dots, v_{|s^u|}^u]$, 其中 $u \in U$ 表示用户集, $v \in V$ 表示项目集。对于用户 u , 顺序推荐任务涉及预测下一步 $|s^u| + 1$ 最可能的项目交互。这可以公式化如下:

$$\arg \max_{v \in V} P(v_{|s^u|+1}^u = v | s^u) \quad (1)$$

3.2 公式化数据增强

浅层数据增强: 在不牺牲一般性的情况下, 我们采用三种数据增强方法进行序列增强, 遵循之前工作中概述的方法 [22]。增广可以表述为:

Crop (C): 从原始序列 s^u 中选择一个长度为 L_C 的随机子序列，子序列从位置 C 裁剪，这个过程可以定义为：

$$s_u^C = [v_c^u, v_{c+1}^u, \dots, v_{c+L_C-1}^u] \quad (2)$$

Mask (M): 从原始序列 s^u 中随机选择项目进行屏蔽。掩码操作可以描述如下：

$$s_u^M = [\hat{v}_1^u, \hat{v}_2^u, \dots, \hat{v}_{|s^u|}^u] \quad (3)$$

其中 \hat{v}_u^t 表示随机被屏蔽的项。

Reorder (R): 从 r 开始的给定长度 L_R 的连续子序列在原始序列 s^u 内随机洗牌。得到的重新排序的序列可以计算为：

$$s_u^R = [v_1^u, \dots, \hat{v}_r^u, \dots, \hat{v}_{r+L_R-1}^u, \dots, v_{|s^u|}^u] \quad (4)$$

深度数据增强: 已经观察到，过度使用数据增强方法可能会对原始模型的预测准确性产生负面影响，特别是对于固有地保存大量历史信息的较长序列。这种过度的增强可能会导致次优结果。为了解决这个问题，并且为了增强序列推荐性能的鲁棒性，引入了以下方法，旨在彻底探索序列推荐中的潜在空间，从而生成更鲁棒的增强项目：

池化 (P): 使用池化算子来探索项目的隐藏信息。该操作的描述如下：

$$s_u^P = [v_1^u, \dots, v_{n-1}^u, p, v_{n+L_P}^u, \dots, v_{|s^u|}^u] \quad (5)$$

其中 $p = \text{mean}(v_n^u, \dots, v_{n+L_P-1}^u)$ 为生成项， n 是池子序列的起始位置， L_P 是用于池化的项数。

3.3 稀疏增强网络

SparseEnNet 的完整结构如图 2 所示，其中每个组件的具体说明如下。

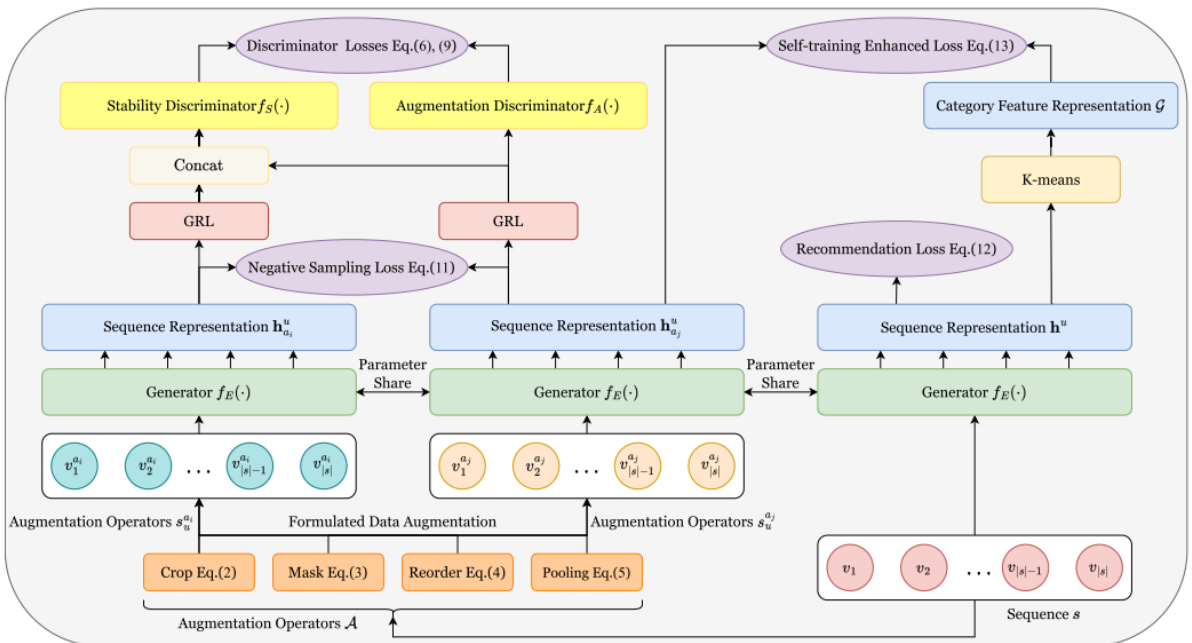


图 2. SparseEnNet 的总体结构

尽管上述数据增强方法可以缓解短序列中的数据稀疏性问题，但如果生成的项目与原始序列之间的分布存在实质性差异，则生成的项目可能会对原始序列产生不利影响。为了减少分布差异对模型性能的影响，设计了一种对抗性生成方法，允许模型区分数据增强类别，并对序列中的下一个项目实现更好的预测性能。具体内容如下。

增强鉴别器: 为了解决上述问题，开发了一种最小最大博弈的增强鉴别器。为了增强编码器在不同增广范围内捕获潜在不变性的能力，增广鉴别器的目的是区分序列表示的源增广方法。在这种情况下，将单个增广序列表示输入到鉴别器中，以确定从增广集中使用的增广操作。表示为非线性函数，增广鉴别器用 $f_{AD}(\cdot)$ 表示。该模块的损失函数使用交叉熵，定义如下：

$$\mathcal{L}_{AD} = - \sum_{u \in \mathcal{U}} \sum_{a_j \in \mathcal{A}} j \log \left(f_{AD} \left(h_{a_j}^u \right) \right) \quad (6)$$

其中 a_j 为增广操作集 \mathcal{A} 中的标签 (如“公式数据增强”一节所述), $h_{a_j}^u$ 为从生成器 $f_E(\cdot)$ 导出的序列表示。鉴别器中的参数 θ_{AD} 通过以下方式进行优化：

$$\hat{\theta}_{AD} = \arg \min_{\theta_{AD}} \mathcal{L}_{AD} (\theta_E, \theta_{AD}) \quad (7)$$

其中 θ_E 表示编码器中的参数。相应地，对其进行如下优化：

$$\hat{\theta}_E = \arg \max_{\theta_E} \mathcal{L}_{AD} (\theta_E, \theta_{AD}) \quad (8)$$

在增广操作上，Eq.(6) 的分类损失间接评估了编码器在各种增广操作上捕获潜在不变性的能力。损失 \mathcal{L}_{SD} 的较大值表明编码器在不同增广方法之间提取不变性的能力得到了提高。相反，小的损失表示鉴别器在数据增强类别之间进行了有效的区分，从而增强了对序列中后续项目的预测性能。

稳定性鉴别器: 在这一部分中，目标是稳定由相同的增强操作生成的项嵌入。将稳定性鉴别器设计为一个非线性函数 $f_{SD}(\cdot)$ ，并通过交叉熵定义鉴别器的分类损失如下：

$$\mathcal{L}_{SD} = - \sum_{u \in \mathcal{U}} \sum_{a_j, a'_j \in \mathcal{A}} \log \left(f_{SD} \left(h_{a_j}^u \| h_{a'_j}^u \right) \right) \quad (9)$$

其中 a'_j 表示与 a_j 相同的数据增强类型， $\|$ 表示连接。在同一序列上重复两次相同的增强操作，同时减少了损失函数优化过程中的鉴别器损失。该策略确保了编码器的稳定学习。类似地，在鉴别器中更新参数 θ_{SD} 如下：

$$\hat{\theta}_{SD} = \arg \min_{\theta_{SD}} \mathcal{L}_{SD} (\theta_E, \theta_{SD}) \quad (10)$$

负样本学习: 最近，自监督对比方法 [22] [15] 由于在批训练过程中增强负样本学习方面取得了显著的成功，在推荐领域受到了广泛的关注。采用这种负样本学习方法来最大化两个正对之间的相互信息，同时有效地增加与负项的距离。利用 InfoNCE 损失作为损失函数，如下所示：

$$\mathcal{L}_{NSL} = \sum_{u \in \mathcal{U}} \sum_{a_i, a_j \in \mathcal{A}} - \log \frac{\exp(\text{sim}(h_{a_i}^u, h_{a_j}^u)/\tau)}{\sum_{\bar{u} \in \text{neg}} \exp(\text{sim}(h_{a_i}^u, h_{\bar{u}}^u)/\tau)} \quad (11)$$

其中 $\text{sim}(\cdot)$ 表示点积, τ 是比例因子, $h_{a_i}^u$ 和 $h_{a_j}^u$ 表示 $s_{a_i}^u$ $s_{a_j}^u$ 的序列表示, 它们被两个不同的增广算子 a_i 和 a_j 独立增。由 $h_{a_i}^u$ 和 $h_{a_j}^u$ 组成的对被认为是正对, 而 $s_u^{a_j}$ 则是不属于用户 u 但驻留在同一训练批内的增广序列。在这种上下文中, 它被视为负对。

下一项预测: 在不损失通用性的情况下, 使用 Transformer [19] 作为序列编码器来提取序列信息以预测下一项。使用对数似然损失函数来优化时间步长为 t 的预测:

$$\mathcal{L}_{NIP}(s^u, t) = -\log(\sigma(h_t^u \cdot e_{v_{t+1}})) - \sum_{v_j \notin s^u} \log(1 - \sigma(h_t^u \cdot e_{v_j})) \quad (12)$$

其中 h_t^u 表示位置 t 的 Transformer 编码器的输出, $e_{v_{t+1}}$ 表示时间步长 t 的实际下一个项目, σ 表示 Sigmoid 函数, v_j 对应于随机选择的负项目, 该负项目不存在于序列 s^u 中, 并从批中提取 [4]。

自训练增强学习: 正如“增强判别器”一节中提到的, 将同一序列的两个不同的增强视为正值, 并将不同序列的增强视为负值, 旨在使正值更紧密地联系在一起, 将负值推得更远。然而, 即使两个不同的序列有些相似, 鉴别器仍然可能将它们视为阴性并试图将它们分开, 这可能会阻碍特征编码器捕获相似序列之间一致性的能力。受 [4] 的启发, 使用序列编码器对所有原始序列进行编码然后将序列表示聚合成一组序列表示。然后, 应用 k-means 聚类获得不同类别嵌入表示的伪聚类。聚类允许将特征相似的序列分组到同一类别中, 称之为自训练增强学习。利用每个类别的均值作为类别特征表示, 并将该类别内的序列应用于以下损失:

$$\sum_{g \in \mathcal{G}} \sum_{u \in \mathcal{U}_g} -\log \frac{\exp(\text{sim}(h_a^u, g) / \tau)}{\sum_{\bar{g} \in \mathcal{G}} \exp(\text{sim}(h_{a_j}^u, \bar{g}) / \tau)} \quad (13)$$

其中 \mathcal{U}_g 为属于集群 g 的用户集合, g 为集群特征表示。

3.4 统一训练损失

本质上, 目标是最小化下一个项目预测的推荐损失, 如 Eq.(12) 所示。此外, 采用负样本学习来加强推荐任务, 并采用自我训练来增强学习, 以减轻假阴性问题, 如 Eq.(11) 和 Eq.(13) 所示。为了增强增强序列之间的语义一致性, 需要集成对抗机制, 通过 Eq.(9) 和 Eq.(6) 实现。因此, SparseEnNet 的联合损失函数定义为:

$$\mathcal{L} = \mathcal{L}_{NIP} + \lambda_1 \mathcal{L}_{NSL} + \lambda_2 \mathcal{L}_{SEL} + \lambda_3 (\mathcal{L}_{SD} + \mathcal{L}_{AD}) \quad (14)$$

其中 λ_1 、 λ_2 和 λ_3 作为平衡权值来调节对多任务的强调。

用于对抗训练的 GRL: 为了促进对抗方法的端到端训练, 在编码器和鉴别器之间合并了一个梯度反转层 (GRL) [7]。在反向传播过程中, GRL 将梯度反转, 使得 GRL 之前的参数被优化以增加损失, 而 GRL 之后的参数的梯度方向保持不变以优化以减少损失。GRL 前后参数的优化目标是相反的, 从而达到对抗学习的目的。参数更新过程如下:

$$\theta = \theta - \mu \frac{\partial \mathcal{L}}{\partial \theta} \quad (15)$$

其中 $\theta \in \{\theta_E, \theta_I, \theta_T\}$ 表示编码器和鉴别器内的参数, μ 表示学习率。

4 复现细节

4.1 与已有开源代码对比

本文在复现原代码的基础上使用了生成对抗网络的改进版本，并且加入了多样性损失使得生成的增强数据不仅与真实数据相似，而且保持多样性。

4.2 实验环境搭建

Requirements: Python = 3.9.0 torch \geq 1.12.1 faiss-gpu scipy tqdm nni

4.3 创新点

生成对抗网络 (GANs) 虽然在许多任务中表现优异，但其训练过程中可能会面临不稳定的问题。为了提高方法的稳定性，改进损失函数：使用一些现代 GAN 的改进版本，比如 WGAN (Wasserstein GAN) 或 LSGAN (Least Squares GAN)，来减少模式崩溃或梯度消失的问题。

目前的稀疏增强网络方法只注重在增强数据上增加噪声或者扰动，但对于多样性的增强还可以进一步提升：多样性损失：加入多样性损失（如对比损失），使得生成的增强数据不仅要与真实数据相似，而且要保持多样性。这对于序列推荐系统尤为重要，因为推荐系统需要处理多种可能的用户行为模式。数据模仿：通过模仿不同用户群体的行为来生成增强数据，确保增强数据的广度和多样性。

5 实验结果分析

数据集: 在三个公开可用的数据集上进行实验，这些数据集来源于真实世界的真实数据。美容和玩具子集是从亚马逊评论数据集 [16] 中提取的，该数据集来自世界上最大的电子商务平台之一。此外，还利用了 Yelp 数据集，该数据集是推荐任务中经常使用的资源，起源于商业平台。坚持既定惯例 [14] [22] 进行数据集处理。然后，按时间顺序组织每个用户的评论，并将其转换为适合推荐模型的用户-物品交互序列。

评估指标: 采用留一策略 [14] [24] [22] 是序列推荐中广泛使用的方法，用于划分数据集。在该策略中，对于每个用户-物品交互序列，将最后一个物品作为测试数据，将紧接在它之前的物品作为验证数据。剩下的项目被用来训练模型。为了全面评估所有模型，采用命中率 (HR) 和归一化贴现累积增益 (NDCG) 作为推荐性能的评估指标。HR@top-k 量化了目标物品在 top-k 推荐物品中的比例。在比较中，NDCG@top-k 考虑了目标物品在 top-k 推荐物品中的排名。

基线方法: 选定的基线方法包括以下内容: [11] 将 GRU 架构集成到基于会话的推荐任务中，通过新颖的损失函数设计和有效的采样策略增强模型性能。SAS-Rec [14] 利用基于 transformer 的单向注意机制来捕获顺序模式并提出下一项建议。BERT4Rec [18] 将 BERT 技术 [5] 应用于序列推荐，该技术最初在 NLP 中取得了成功。它采用双向自注意机制和掩码操作来增强性能。CL4SRec [22] 在序列推荐中采用对比学习，为对比学习框架引入了裁剪、掩码

和重排序增强技术。ICLRec [4] 对用户意图进行无监督建模，并将这些意图整合到对比顺序推荐任务中。

表 1 给出了所有方法的综合性能比较。从表中可以观察到：

(1) 提出的 SparseEnNet 在所有数据集上都优于所有其他基线，强调了提出的模型的有效性。值得注意的是，SparseEnNet 表现出了显著的性能改进，在 Beauty、Toys 和 Yelp 数据集 Hit@5 方面，它比第二好的模型分别提高了 11.93%、6.91% 和 4.27%。此外，就 NDCG@5 而言，SparseEnNet 在所有数据集上都比表现第二好的模型有了实质性的改进，在 Beauty、Toys 和 Yelp 数据集上分别获得了 14.47%、7.09% 和 6.21% 的收益。类似的增强在其他评估指标上也很明显。

(2) BERT4Rec 持续优于 GRU4Rec，巩固自注意机制在提取序列特征用于序列推荐任务中的功效。此外，与 BERT4Rec 相比，SASRec 表现出更优越的性能。这种差异可能归因于这样一个事实，即稀疏数据集中的掩蔽操作会导致此类稀疏数据中更明显的上下文信息丢失。

(3) CL4SRec 和 ICLRec 为序列数据增强量身定制，优于其他基线模型。毫无疑问，CL4SRec 在所有数据集上都取得了第二好的性能，强调了数据增强技术在序列推荐领域的重要性和有效性。

表 1. top-N 推荐上各种方法的性能比较。每个指标的最佳得分用粗体表示，次优得分用下划线表示。最后一行显示了每个数据集上优于最佳基线的改进。

Datasets	Methods	Hit@5	NDCG@5	Hit@10	NDCG@10	Hit@20	NDCG@20
Beauty	GRU4Rec	0.0256	0.0164	0.0426	0.0218	0.0690	0.0285
	SASRec	0.0338	0.0222	0.0532	0.0285	0.0828	0.0359
	BERT4Rec	0.0293	0.0183	0.0477	0.0242	0.0688	0.0295
	CL4SRec	0.0427	0.0278	0.0648	0.0349	0.0957	0.0427
	ICLRec	<u>0.0461</u>	<u>0.0304</u>	<u>0.0728</u>	<u>0.0389</u>	<u>0.1054</u>	<u>0.0471</u>
	SparseEnNet	0.0516	0.0348	0.0762	0.0426	0.1103	0.0512
	Improv.	11.93%	14.47%	4.67%	9.51%	4.65%	8.70%
Toys	GRU4Rec	0.0211	0.0145	0.0337	0.0186	0.0536	0.0236
	SASRec	0.0399	0.0264	0.0584	0.0324	0.0832	0.0387
	BERT4Rec	0.0304	0.0199	0.0461	0.0248	0.0689	0.0305
	CL4SRec	0.0541	0.0449	0.0772	0.0374	0.1063	0.0522
	ICLRec	<u>0.0579</u>	<u>0.0395</u>	<u>0.0820</u>	<u>0.0472</u>	<u>0.1131</u>	<u>0.0550</u>
	SparseEnNet	0.0619	0.0423	0.0855	0.0499	0.1162	0.0576
	Improv.	6.91%	7.09%	4.27%	5.72%	2.74%	5.03%
Yelp	GRU4Rec	0.0152	0.0091	0.0248	0.0124	0.0371	0.0145
	SASRec	0.0160	0.0101	0.0260	0.0133	0.0443	0.0179
	BERT4Rec	0.0196	0.0121	0.0339	0.0167	0.0564	0.0223
	CL4SRec	0.0227	0.0143	0.0384	0.0194	0.0623	0.0254
	ICLRec	<u>0.0234</u>	<u>0.0145</u>	<u>0.0401</u>	<u>0.0199</u>	<u>0.0645</u>	<u>0.0260</u>
	SparseEnNet	0.0244	0.0154	0.0414	0.0209	0.0678	0.0275
	Improv.	4.27%	6.21%	3.24%	5.72%	5.12%	5.77%

6 总结与展望

在本文中,复现了 SparseEnNet,这是一种创新的、鲁棒的对抗生成方法,旨在通过生成更鲁棒的增强项来彻底探索序列推荐中的潜在空间。SparseEnNet 框架由四个基本部分组成:增强鉴别器、稳定性鉴别器、负样本学习模块和自训练增强学习模块。通过在三个知名数据集上进行的大量实验,建立了模型的有效性。此外,消融研究进一步验证了每个单独成分的贡献。此外,一个关注冷启动问题的案例研究证实了 SparseEnNet 在稀疏数据集中产生不同项目嵌入的能力。在实际应用中,生成对抗网络可能会面临训练不稳定的问题,未来可以探索更高效、更稳定的对抗训练方法,如 WGAN、LSGAN 等,以提高模型的收敛速度和稳定性。

参考文献

- [1] Samet Akcay, Amir Atapour-Abarghouei, and Toby P Breckon. Ganomaly: Semi-supervised anomaly detection via adversarial training. In *Computer Vision-ACCV 2018: 14th Asian Conference on Computer Vision, Perth, Australia, December 2-6, 2018, Revised Selected Papers, Part III 14*, pages 622-637. Springer, 2019.
- [2] Junyang Chen, Jialong Wang, Zhijiang Dai, Huisi Wu, Mengzhu Wang, Qin Zhang, and Huan Wang. Zero-shot micro-video classification with neural variational inference in graph prototype network. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 966-974, 2023.
- [3] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597-1607. PMLR, 2020.
- [4] 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.
- [5] Jacob Devlin. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.
- [6] Hongchao Fang, Sicheng Wang, Meng Zhou, Jiayuan Ding, and Pengtao Xie. Cert: Contrastive self-supervised learning for language understanding. *arXiv preprint arXiv:2005.12766*, 2020.
- [7] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario March, and Victor Lempitsky. Domain-adversarial training of neural networks. *Journal of machine learning research*, 17(59):1-35, 2016.

- [8] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.
- [9] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020.
- [10] Xiangnan He, Zhankui He, Xiaoyu Du, and Tat-Seng Chua. Adversarial personalized ranking for recommendation. In *The 41st International ACM SIGIR conference on research & development in information retrieval*, pages 355–364, 2018.
- [11] B Hidasi. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939*, 2015.
- [12] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670*, 2018.
- [13] Wendi Ji, Keqiang Wang, Xiaoling Wang, Tingwei Chen, and Alexandra Cristea. Sequential recommender via time-aware attentive memory network. In *Proceedings of the 29th ACM international conference on information & knowledge management*, pages 565–574, 2020.
- [14] Wang-Cheng Kang and Julian McAuley. Self-attentive sequential recommendation. In *2018 IEEE international conference on data mining (ICDM)*, pages 197–206. IEEE, 2018.
- [15] 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.
- [16] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*, pages 43–52, 2015.
- [17] Ruiyang Ren, Zhaoyang Liu, Yaliang Li, Wayne Xin Zhao, Hui Wang, Bolin Ding, and Ji-Rong Wen. Sequential recommendation with self-attentive multi-adversarial network. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*, pages 89–98, 2020.
- [18] 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.

- [19] A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.
- [20] Jiancan Wu, Xiang Wang, Fuli Feng, Xiangnan He, Liang Chen, Jianxun Lian, and Xing Xie. Self-supervised graph learning for recommendation. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*, pages 726–735, 2021.
- [21] 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.
- [22] Xu Xie, Fei Sun, Zhaoyang Liu, Shiwen Wu, Jinyang Gao, Jiandong Zhang, Bolin Ding, and Bin Cui. Contrastive learning for sequential recommendation. In *2022 IEEE 38th international conference on data engineering (ICDE)*, pages 1259–1273. IEEE, 2022.
- [23] Tiansheng Yao, Xinyang Yi, Derek Zhiyuan Cheng, Felix Yu, Ting Chen, Aditya Menon, Lichan Hong, Ed H Chi, Steve Tjoa, Jieqi Kang, et al. Self-supervised learning for large-scale item recommendations. In *Proceedings of the 30th ACM international conference on information & knowledge management*, pages 4321–4330, 2021.
- [24] 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.
- [25] Andrew Zimdars, David Maxwell Chickering, and Christopher Meek. Using temporal data for making recommendations. *arXiv preprint arXiv:1301.2320*, 2013.