

1. 背景介绍

在当今信息化时代，推荐系统已经渗透到各类互联网平台中，成为连接用户与内容的重要桥梁。例如，在电子商务平台（如亚马逊和淘宝），推荐系统帮助用户快速找到感兴趣的商品；在流媒体服务中（如Netflix和Spotify），系统为用户推送契合其喜好的视频或音乐；在社交媒体平台（如抖音和Instagram），推荐机制驱动了用户与内容之间的互动，提升了用户留存率。

随着数据规模的增长和用户需求的多样化，推荐系统逐渐从简单的规则方法演化为高度复杂的机器学习系统。然而，现代推荐技术仍然面临以下挑战：

- 1. **数据稀疏性**：大多数用户只与少量物品交互，交互矩阵极度稀疏，导致推荐模型难以有效学习。
- 2. **噪声问题**：用户行为数据中往往存在噪声，例如误点击和虚假的交互记录。
- 3. **动态偏好**：用户兴趣会随时间变化，推荐模型需要适应这种变化。

常见的推荐系统技术包括以下几种：

1、基于内容的方法：

早期的推荐系统多采用基于内容的推荐方法，通过分析物品的特征为用户推荐相似物品。例如，在电影推荐场景中，用户喜欢某部电影后，系统会推荐具有相同导演或相似题材的电影。然而，这种方法有两个主要问题：依赖于高质量的物品特征描述，获取这些特征可能代价高昂；容易导致推荐的多样性不足，用户探索性体验较差。

2、协同过滤：

协同过滤方法通过挖掘用户行为之间的相似性实现推荐。分为两种主要类型：1、通过找到相似用户群体，推荐这些用户喜欢的物品。2、通过分析物品的相似性，将用户对某物品的喜好推及至相似物品。协同过滤的优势在于无需显式依赖物品特征，但它在以下情况下表现较差：数据稀疏性高时，无法有效建立相似性；面对新用户或新物品时，即“冷启动问题”，缺乏足够的历史数据支持。

3、矩阵分解方法

矩阵分解技术通过将用户-物品交互矩阵分解为低维潜在因子表示（Latent Factors），有效解决了数据稀疏性问题。典型的矩阵分解算法包括SVD（奇异值分解）和ALS（交替最小二乘法）。然而，矩阵分解通常假设用户的偏好是静态的，难以捕捉时间变化特性。

尽管这些方法显著提高了推荐性能，但在应对数据噪声和动态变化时，仍有较大改进空间。

生成模型为推荐任务带来了新的研究视角，它们通过建模用户与物品交互的生成过程，可以更加准确地预测用户的未来行为。

1、GAN（生成对抗网络）

GAN通过生成器和判别器的对抗博弈，实现高质量的交互预测。生成器负责生成用户潜在的交互数据，而判别器则判别生成数据与真实数据的差异。然而，GAN在推荐任务中表现有限，主要因为：

- 1. **训练不稳定性**：生成器与判别器之间的动态博弈使得模型训练复杂，难以达到平衡。
- 2. **数据稀疏问题**：在稀疏交互矩阵中，GAN可能生成大量常见模式，忽视长尾物品。

2、VAE (变分自动编码器)

VAE是一种基于概率图模型的生成方法，能够将用户交互数据映射到潜在空间，并通过解码器预测未来交互行为。VAE的优点在于：

1. 能够自然建模数据中的噪声。
2. 适合高维稀疏数据的分布建模。

然而，VAE在表达能力上仍然存在局限，特别是在需要高精度建模复杂用户行为的场景。

扩散模型的潜力

扩散模型 (Diffusion Models) 最初用于图像生成，通过逐步添加噪声 (正向扩散) 和逐步去噪 (逆向生成) ，实现对数据分布的高效建模。其关键特性包括：

1. **逐步建模**：扩散模型通过迭代方式生成数据，能够捕获细粒度的分布特性。
2. **鲁棒性**：由于其核心机制是逐步去噪，扩散模型对噪声数据具有较强的适应能力。

扩散模型的这些特性表明其在推荐任务中具有很大潜力，特别是应对数据稀疏性和噪声问题。

本文贡献

本文提出了一种基于扩散模型的推荐框架 (Diffusion Recommender Model, DiffRec) ，其创新点包括：

1. 提出扩散推荐模型 (DiffRec) ，通过逐步去噪生成用户交互分布。
2. 针对推荐任务的特点，设计了两个扩展模块：L-DiffRec (降维扩散) 和T-DiffRec (时间建模) 。
3. 在三个真实数据集上进行实验，验证了DiffRec在干净数据、含噪数据和时间序列数据中的优越性。

2. 扩散推荐模型 (DiffRec)

2.1 建模过程

扩散模型由两个核心过程组成：正向扩散和逆向生成。

正向扩散过程

正向扩散通过逐步添加噪声，将原始数据分布扰乱为高斯分布：

$$q(\mathbf{x}_t | \mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t} \mathbf{x}_{t-1}, \beta_t \mathbf{I})$$

其中， β_t 为每一步的噪声强度参数。经过多次迭代，数据逐渐被转换为接近标准正态分布的形式。

逆向生成过程

逆向生成从完全扰乱的数据开始，通过神经网络逐步去噪恢复原始数据：

$$p_{\theta}(\mathbf{x}_{t-1} | \mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \mu_{\theta}(\mathbf{x}_t, t), \Sigma_{\theta}(\mathbf{x}_t, t))$$

其中， μ_{θ} 和 Σ_{θ} 由参数化神经网络生成。

优化目标：ELBO

扩散模型的优化通过最大化证据下界（ELBO）实现：

$$\mathcal{L}_{\text{ELBO}} = \mathbb{E}_q[\log p(\theta(\mathbf{x}_0 | \mathbf{x}_1))] - \sum_{t=2}^T \mathbb{E}_q[D_{\text{KL}}(q(\mathbf{x}_{t-1} | \mathbf{x}_t, \mathbf{x}_0) || p(\theta(\mathbf{x}_{t-1} | \mathbf{x}_t)))]$$

2.2 DiffRec的创新点

扩散推荐模型的核心在于使用扩散模型预测用户未来的交互概率，其主要设计包括以下几个模块：

1. **正向扩散**：通过添加小规模的噪声逐步扰乱用户交互历史，同时避免完全丢失个性化信息。
2. **逆向生成**：通过参数化神经网络逐步去噪，恢复用户可能的交互分布。

L-DiffRec和T-DiffRec扩展

面对推荐场景下的应用与挑战，扩散模型同样面临以下挑战：

1. **高维稀疏性**：用户与物品交互矩阵通常是稀疏的，直接扩散可能导致数据丢失。
2. **资源开销**：生成模型需要对所有物品的交互概率进行预测，计算成本高昂。
3. **动态偏好**：用户偏好随时间变化，需要模型具备时间敏感性。

为解决推荐任务中的资源和动态问题，DiffRec提出了两种扩展：

1. **L-DiffRec（降维扩散）**：通过物品聚类和潜在空间扩散，降低计算复杂度。
2. **T-DiffRec（时间建模）**：采用时间权重策略，捕捉用户偏好的时间变化。

3. 实验与结果

实验在三个真实数据集上进行：

Amazon-book：包含丰富的书籍交互数据。**Yelp**：用户对餐馆的评价数据。**ML-1M**：电影评分数据集。

数据集分为干净数据（去除低评分）、含噪数据（加入低评分或随机噪声）和时间序列数据三种设置。采用 Recall@K 和 NDCG@K 评估推荐效果。

3.1 DiffRec在不同设置下的性能表现

Model	MIND						ML-1M					
	HR@5	NDCG@5	HR@10	NDCG@10	HR@20	NDCG@20	HR@5	NDCG@5	HR@10	NDCG@10	HR@20	NDCG@20
FM	0.1204	0.0745	0.1923	0.0977	0.2663	0.1164	0.3214	0.2113	0.4870	0.2646	0.6836	0.3142
Wide&Deep	0.0949	0.0623	0.1339	0.0749	0.1809	0.0868	0.3641	0.2462	0.5289	0.2991	0.7248	0.3486
DeepFM	0.1183	0.0780	0.1942	0.1022	0.3018	0.1295	0.3264	0.2180	0.4866	0.2694	0.6853	0.3195
AFM	0.1199	0.0762	0.1650	0.0908	0.2325	0.1077	0.3845	0.2600	0.5538	0.3146	0.7551	0.3656
DCN	0.1146	0.0771	0.1725	0.0957	0.2660	0.1191	0.3695	0.2503	0.5370	0.3042	0.7288	0.3527
xDeepFM	0.1143	0.0712	0.1787	0.0919	0.2792	0.1171	0.3320	0.2234	0.4930	0.2751	0.6847	0.3235
AutoInt	0.1384	0.0927	0.2151	0.1172	0.3594	0.1533	0.3615	0.2439	0.5205	0.2951	0.7150	0.3443
DCNv2	0.1472	0.0978	0.2222	0.1218	0.3666	0.1577	0.3740	0.2541	0.5473	0.3100	0.7403	0.3588
FinalMLP	0.1759	0.1161	0.2558	0.1419	0.3576	0.1676	0.3834	0.2627	0.5464	0.3152	0.7404	0.3643
SAM	0.0935	0.0579	0.1566	0.0782	0.2442	0.1002	0.3774	0.2564	0.5451	0.3104	0.7434	0.3605
DIN	0.2536	0.1686	0.3627	0.2038	0.4030	0.2390	0.4939	0.3615	0.6448	0.4101	0.7921	0.4475
DIEN	0.2351	0.1628	0.2946	0.1822	0.3543	0.1970	0.5227	0.3881	0.6705	0.4360	0.8163	0.4729
CAN	0.1952	0.1203	0.3041	0.1622	0.4946	0.1999	0.4989	0.3624	0.6441	0.4110	0.7958	0.4517

Methods	R@10	R@20	N@10	N@20	R@10	R@20	N@10	N@20	R@10	R@20	N@10	N@20
MF	0.0437	0.0689	0.0264	0.0339	0.0341	0.0560	0.0210	0.0276	0.0876	0.1503	0.0749	0.0966
LightGCN	0.0534	0.0822	0.0325	0.0411	0.0540	0.0904	0.0325	0.0436	0.0987	0.1707	0.0833	0.1083
CDAE	0.0538	0.0737	0.0361	0.0422	0.0444	0.0703	0.0280	0.0360	0.0991	0.1705	0.0829	0.1078
MultiDAE	0.0571	0.0855	0.0357	0.0442	0.0522	0.0864	0.0316	0.0419	0.0995	0.1753	0.0803	0.1067
MultiDAE++	0.0580	0.0864	0.0363	0.0448	0.0544	0.0909	0.0328	0.0438	0.1009	0.1771	0.0815	0.1079
MultiVAE	0.0628	0.0935	0.0393	0.0485	0.0567	0.0945	0.0344	0.0458	0.1007	0.1726	0.0825	0.1076
CODIGEM	0.0300	0.0478	0.0192	0.0245	0.0470	0.0775	0.0292	0.0385	0.0972	0.1699	0.0837	0.1087
DiffRec	0.0695*	0.1010*	0.0451*	0.0547*	0.0581*	0.0960*	0.0363*	0.0478*	0.1058*	0.1787*	0.0901*	0.1148*
% Improve.	10.67%	8.02%	14.76%	12.78%	2.47%	1.59%	5.52%	4.37%	4.86%	0.90%	9.21%	6.69%

	Amazon-book				Yelp				ML-1M			
	R@10	R@20	N@10	N@20	R@10	R@20	N@10	N@20	R@10	R@20	N@10	N@20
LightGCN	0.0400	0.0659	0.0231	0.0308	0.0466	0.0803	0.0278	0.0379	0.0648	0.1226	0.0470	0.0679
MultiVAE	0.0536	0.0820	0.0316	0.0401	0.0494	0.0834	0.0293	0.0396	0.0653	0.1247	0.0469	0.0680
DiffRec	0.0546	0.0822	0.0335	0.0419	0.0507	0.0853	0.0309	0.0414	0.0658	0.1236	0.0488	0.0703

	Amazon-book				Yelp				ML-1M			
	R@10	R@20	N@10	N@20	R@10	R@20	N@10	N@20	R@10	R@20	N@10	N@20
DiffRec	0.0546	0.0822	0.0335	0.0419	0.0507	0.0853	0.0309	0.0414	0.0658	0.1236	0.0488	0.0703
L-DiffRec	0.0586 ^{+7.3%}	0.0876 ^{+6.6%}	0.0347 ^{+3.6%}	0.0434 ^{+3.6%}	0.0521 ^{+2.8%}	0.0876 ^{+2.7%}	0.0311 ^{+0.7%}	0.0419 ^{+1.2%}	0.0665 ^{+1.1%}	0.1272 ^{+2.9%}	0.0493 ^{+1.0%}	0.0710 ^{+1.0%}

3.2 实验结论

1. 干净数据集上的表现 DiffRec在三个数据集上的推荐精度显著高于基线模型（如MultiVAE和LightGCN）。这表明扩散模型对用户复杂偏好的建模能力优于传统生成模型。
2. 含噪数据上的鲁棒性分析 在含自然噪声和随机噪声的数据集上，DiffRec表现出较强的抗噪能力。特别是当噪声比例增加时，DiffRec仍能保持较高的推荐精度。
3. 重要性采样的作用：通过在高损失步骤上增加权重，显著提高了模型的优化效果。
4. 噪声规模的影响：过大的噪声会削弱个性化信息，适当的噪声规模能够提升性能。
5. L-DiffRec对资源消耗的优化效果：与DiffRec相比，L-DiffRec在性能接近的情况下显著降低了内存和参数开销。
6. T-DiffRec对时间序列建模的提升：在时间序列数据集上，T-DiffRec优于现有的序列推荐模型（如ACVAE）。