arXiv:2303.01297v1 [cs.IR] 2 Mar 2023

# 利用生成对抗网络为协同过滤推荐系统创建合成数据集

**Jesús Bobadilla**
马德里理工大学计算机系统高等技术学院计算机系统系，C/ Alan Turing s/n 28031，马德里，西班牙 jesus.bobadilla@upm.es

**Abraham Gutiérrez**
马德里理工大学计算机系统高等技术学院计算机系统系，C/ Alan Turing s/n 28031，马德里，西班牙 abraham.gutierrez@upm.es

**Raciel Yera**
信息学系 哈恩大学 哈恩，西班牙 谢戈德阿维拉大学 谢戈德阿维拉，古巴
ryera@ujaen.es

**Luis Martínez**
信息学系 哈恩大学 哈恩，西班牙
martin@ujaen.es

**摘要**

机器学习研究与教育需要多样、具有代表性且开放的数据集，其中包含足够的样本，以满足必要的训练、验证和测试任务。目前，推荐系统领域涵盖众多子领域，其准确率及超越准确率的质量度量不断得到改进。为支撑这类多样化的研究，有必要且适宜通过合成数据集来补充现有数据集。本文提出一种基于生成对抗网络（GAN）的方法，以参数化方式生成协同过滤数据集，用户可选择所需的用户数量、物品数量、样本数量和随机变异性。常规 GAN 无法实现这种参数化。与基于大型稀疏输入向量的传统方法相比，我们的 GAN 模型以稠密、短小且连续的用户和物品嵌入表示作为输入，而非稀疏、大型且离散的向量，从而实现准确、快速的学习。所提出的架构包含一个 DeepMF模型，用于提取稠密的用户和物品嵌入，并通过聚类过程将 GAN 生成的稠密样本转换为离散且稀疏的样本，以创建所需的每个合成数据集。三个不同源数据集的结果表明，与源数据集相比，生成的数据集具有适当的分布以及符合预期的质量值和变化趋势。研究人员可获取合成数据集和源代码。

**关键词** 推　　荐系统 · 生成对抗网络 · 深度学习 · 协同过滤　　g

## 1 引言

推荐系统（RS）是人工智能中的一个重要领域，这得益于社交网络日益增长的普及度。广泛使用推荐系统的知名企业包括 TripAdvisor、Netflix、Spotify、YouTube Music、TikTok、YouTube 和亚马逊 [1]。这些企业利用推荐系统模型，向用户推荐与其已经消费过的物品（音乐、视频、旅行、新闻）相似的物品；而 Facebook 等其他企业则致力于收集客户活动数据，以提供个性化宣传，而非个性化产品或服务。推荐系统通常根据其过滤方法进行分类 [2]；基于内容的推荐系统通过寻找相似内容来选择推荐物品

[3]；由于大多数物品内容都是文本，因此会使用自然语言处理模型。评论 [4] 和推文 [5]是两种常见的基于内容过滤的数据。产品图像也可以经过处理以生成推荐；卷积神经网络是执行此任务时最常用的模型 [6]。社交过滤已被广泛用于改进基于社交的推荐。这种过滤方法使用标签、关注者和被关注者等数据，并利用声誉和信任的概念 [7]。GPS 坐标和兴趣点等地理信息主要用于支持上下文感知过滤 [8]。人口统计过滤（年龄、性别、国家/地区等）通常会与其他过滤类型结合，从而实现推荐集成 [9]。除上述过滤策略外，协同过滤（CF） [10] 是实现推荐系统最重要的方法，因为它能够提供更高的准确率，尤其是在与其他过滤类型结合时。有效的推荐系统研究会采用创新模型、适当的质量度量和具有代表性的数据集。推荐系统研究主要聚焦于模型和度量，而本文试图在协同过滤数据集方面取得进展。

协同过滤的发展历程始于基于记忆的模型，主要是K近邻算法[11]。由于基于模型的机器学习方法整体性能更优，基于记忆的方法逐渐被取代：它们的结果准确率更高，获取预测结果所需的时间也更短（模型训练完成后）；而且其输出还能通过事后技术进行解释 [12]。矩阵分解 (MF) [13] 一直是实现协同过滤最常用的机器学习模型；它对用户和物品进行降维，从而捕捉用户和物品与所给评分之间的主要模式。此外，通过使用非负矩阵分解 (NFM) [14], , 可以为潜在因子赋予语义含义。贝叶斯非负矩阵分解 [15] 使得同时对用户进行分组并做出预测成为可能，为面向用户群体的有效推荐以及社交聚类应用打开了大门 [16]。目前，协同过滤研究以深度学习模型为基础，其中DeepMF[17] 是现代方法的基础。本文采用的DeepMF模型通过嵌入层将用户编码到一个潜在空间中，同时通过另一个嵌入层将物品编码到不同的潜在空间中；最后，对物品嵌入和用户嵌入进行点积，以生成预测。DeepMF优于矩阵分解，这是因为神经网络天生擅长捕捉样本之间的非线性关系。神经协同过滤 (NCF) [18] 被广泛用于实现协同过滤；该模型以多层感知机 (MLP)取代DeepMF的点积层，并且在大型复杂数据集上表现优于DeepMF。除了提高准确率，深度学习模型也开始承担一些创新任务，例如提升公平性：DeepFair模型 [19] 在公平性与精确率之间取得了平衡；绿色计算 [20]；通过潜在空间可视化解释结果 [21]；以及高效的邻域识别 [22]。基于对抗网络的推荐最近被引入推荐系统领域 [23] , 我们将在“相关工作”一节重点讨论。生成对抗网络 (GAN) [24]因生成广受欢迎的伪造人脸和伪造视频而闻名，这些内容充斥着社交网络。其架构包含两个相互竞争（“对抗”）的独立神经网络，例如艺术品伪造者与艺术专家相互较量，从而确保双方都不断改进。GAN中的“伪造者”是生成器模型，它通过随机噪声向量创建伪样本；GAN中的“专家”则是判别器模型，以简单的二元分类器实现，类别为：伪造、非伪造。

质量度量是开展有效研究的关键，因为它们使人们能够将最先进的基线与所提出的算法、方法和模型进行比较。除了常见的预测和推荐质量指标（平均绝对误差、均方差、精确率、召回率、F1分数、归一化折损累计增益等）之外，近年来，其他一些指标（如新颖性和多样性） [25],也日益重要。其中，多样性受到研究人员的关注，因为社交网络中的推荐可能存在不足，例如缺乏可变性并助长先入为主的观念和行为。通过在矩阵分解模型中引入多样性增强约束，提升了推荐系统中的多样性和可靠性 [26]；此外，还提出了一种深度学习分类模型 [27] , 用于从神经网络的 softmax 输出层获取推荐可靠性值。在均衡协同过滤数据集上测试模型或方法后，即可获得质量值。文献 [28]提出了确定性策略，用于构建在用户和物品分布方面均衡的训练集和测试集。大多数推荐系统研究使用 Movielens、FilmTrust、MyAnimeList 或 CiteSeer 等热门协同过滤数据集；协同过滤数据集涵盖不同领域，例如音乐、电影、兴趣点、旅游、新闻、研究论文、带标签数据等。有些数据集包含用户给出的显式评分，而另一些则包含用户与系统之间的隐式交互。此外，还有一些数据集包含爬取的网页或学术PDF文件 [29] , 另一些则通过研究人员为文章添加的社交标签 [30]加以丰富。文中列出了一系列相关的社交协同过滤数据集 [31] , 并介绍了使用这些数据集的一些文章。最近，发布了一个教育新闻数据集 [32] , 其中包含情境信息：时间和位置。最后，文中还提供了一个推荐系统数据集，其中包含人工智能研究数据 [33] , 可用于获取分段信息、聚类结果和地理位置。值得特别注意的是，迄今尚未使用合成数据集，因此协同过滤研究未能受益于参数化在实验设计中带来的灵活性，例如可灵活设置数据集规模、用户数量和物品数量等；本文旨在填补这一空白。

本文其余部分安排如下：第 2 节介绍相关工作，重点讨论生成对抗网络模型近期在推荐系统中的应用。第 3 节阐述所提出模型及其形式化过程。第 4 节展示实验的设计、结果和讨论。最后，第 5 节总结本文的主要结论并展望未来工作。

## 2 相关工作

生成式深度学习是协同过滤推荐系统领域的一个创新方向。尽管已有一些变分自编码器方法发表 [34, 35], , 当前研究主要集中于 GAN 模型 [36]。生成对抗网络应用于协同过滤的一个子领域是攻击/防御策略 [37], , 这些模型可以增强推荐系统的安全性。然而，协同过滤生成对抗网络最广泛的应用包括：a）解决数据噪声问题；b）应对数据稀疏问题，并通过捕获真实数据的分布来实现数据增强框架。CFGAN [38] 是一种生成购买向量而非物品 ID 的模型，随后利用生成的虚假购买向量来增强真实向量。CFGAN 的 Wasserstein版本是统一生成对抗网络（UGAN） [39] , 与 CFGAN 相比，其性能有所提升。为优先处理推荐系统中的长期和短期信息（用户与物品之间快速或缓慢变化的交互）, PLASTIC [40] 模型训练了一个生成器，并将其用作强化学习智能体。循环生成对抗网络 RecGAN[41], 学习评分中的时间模式；该模型结合了生成对抗网络和循环神经网络模型。为捕获协同过滤数据集中的负采样信息，IPGAN [42] 实现了两种不同的生成模型：一种用于正样本，另一种用于负样本。IPGAN 考虑了采样得到的正评分与所选负评分之间的关系。

目前，DCGAN 模型 [43] 将 GAN 与强化学习模型相结合，以捕获推荐系统会话中的信息，而非传统的用户对物品评分历史矩阵。会话信息包括用户对当前推荐的响应。用户的即时反馈由与 GAN 相结合的强化学习模型处理。NCGAN [44] 引入神经网络来提取用户的非线性特征，并通过 GAN 指导推荐训练；生成器模型负责为用户生成推荐，而判别器模型则衡量真实分布和生成分布之间的距离。一种改进从生成器到判别器的信息流的创新方法 [45] 可减少协同过滤 GAN 中两个模型之间的差异。 [46], 采用正则化Wasserstein生成对抗网络模型，并将自编码器用作生成器；该模型应用于高维稀疏协同过滤矩阵时，准确率有所提升。 [47]使用 CGAN（条件 GAN）来改进协同过滤推荐，并且可以设定评分向量的大小，从而简化生成器和判别器的任务。此外，该方法还支持条件评分生成。对于不符合标准高斯分布的数据集， [48] 提出了一种基于 GAN 的缺失数据填补方法；结果表明，该方法提高了若干具有代表性的分类数据集的质量。 [49] 利用信任信息来生成有效推荐。该方法提出了一种 GAN，其中判别器为多层感知机模型，生成器为长短期记忆网络模型。最后，由于社交数据收集的特点（例如，年轻人多于老年人），协同过滤数据集通常存在类别不平衡问题。为了解决这一局限， [50] 在生成器中采用 Wasserstein GAN 模型，并在判别器中引入 PacGAN 概念 [51], , 以尽量减少模式崩溃问题。

一种用于多智能体推荐系统仿真的平台是基于概率的 RecSim [52], , 它会生成用户和物品的合成档案，并使用马尔可夫链和循环神经网络。Virtual-Taobao [53] 是一个多智能体强化学习系统，旨在改进社交网站淘宝上的搜索；它利用生成对抗网络来模拟内部数据分布。它采用简单的矩阵分解 [54] , 在推荐流程中实现主题多样化。DataGenCars [55] 是一个基于 Java 的推荐系统合成数据生成器；它具有统计基础，提供了灵活性，但与深度学习生成模型相比，其准确率较低。最后，SynEvaRec 框架 [56] 使用合成数据仓库（SVD）库生成合成推荐系统数据集。该库使用 Copula函数对多元分布进行建模；其 CTGAN 子库包含 GAN 模型。与以往框架相比，SynEvaRec 的主要优势在于可以将不同的推荐系统作为数据源；其主要缺点是在大多数情况下结果质量较差，且训练阶段耗时过长。

以往研究主要致力于改善不同目标，例如降噪、推荐质量、预测值、攻击防御或数据平衡。为此，人们结合了许多不同的方法和信息来源：使用 GAN、条件生成对抗网络、Wasserstein生成对抗网络等。GAN 模型与循环神经网络和长短期记忆时序网络相结合，并在基于生成对抗网络的架构中引入了强化学习。除了信任信息、会话日志（包括用户对先前推荐的反馈）以及推断出的负向投票外，研究还将长、短期数据引入所提出的模型。在这一新兴的生成对抗网络应用于协同过滤推荐系统的领域中，单纯生成合成数据集似乎并非研究目标；目前的重点是利用 GAN 模型捕捉高维稀疏协同过滤数据集中的非线性复杂模式这一固有能力，通过数据增强提升预测和推荐质量。创新之处

在于，我们的方案旨在生成具有代表性且实用的协同过滤合成数据集，而非改善现有数据集的各项质量指标。此外，该方案允许设置具有代表性的参数，并以 Movielens、FilmTrust 或 MyAnimeList 等真实数据集为来源，生成一整“系列”的合成数据集。这些参数包括用户数量、项目数量、样本数量以及所生成数据的可变性。通过改变参数值，我们可以生成同一种协同过滤模式的不同版本，例如，包含 8000 个用户和 3000 个物品的 Movielens 数据集，以及另一个包含 2000 个用户和 1000 个物品的 Movielens 数据集，等等。这样，我们就能测试数据集规模、稀疏度、用户数量和物品数量对准确率及性能的影响，例如，随着用户数量增加，平均绝对误差（MAE）如何改善。据我们所知，目前尚无已发表的方法或模型，能够通过参数化深度学习模型，从多种来源创建准确且可扩展的合成数据集。

## 3 基于生成对抗网络构建协同过滤数据集的方法

本节提出 GANRS 方法，该方法利用 GAN 网络生成合成协同过滤数据集；GAN 以真实的协同过滤数据集为输入，并学习其中的内在模式。最具创新性的贡献是向 GAN 输入用户和物品的稠密、小型嵌入表示，而非采用传统方法，将包含每位用户所投评分的大型稀疏向量输入 GAN。GANRS 方法的主要优势在于能够大幅降低 GAN 架构的复杂度、加快其收敛速度并提升其性能。

为说明 GANRS 方法，我们将使用图1和图3；这两幅图也与本节末尾按步骤列出的形式化描述相关联。图1和图3展示了从真实数据集（Movielens、Netflix 等）生成不同合成数据集所设计的七个阶段。阶段1（图 1 左上方的图）展示了 DeepMF 模型的训练过程，该模型用于设置用户和物品的嵌入层。简而言之，神经网络中的嵌入层能够将输入的稀疏表示高效地转换为输出的稠密表示。对于训练集中的每个输入样本 $< user,\ item,\ rating >$，输出点积层会结合嵌入层的值来预测评分值，并计算输出误差“（评分－预测值）”；随后，该误差将被反向传播，用于更新学习参数。步骤6至10对这些概念进行了形式化说明。DeepMF模型完成学习后，阶段2（图 1 右上方的图）展示了 DeepMF 的前馈过程：每个物品ID（从1到数据集中的物品数量）都会输入物品嵌入层，由此输出该物品ID的稠密表示；通常，协同过滤的嵌入向量大小为5至10。用户ID作为输入时也采用相同的处理方式，并生成相应的稠密表示。请注意，数据集中的物品数量与用户数量并不相同。步骤11至14说明了这一第二阶段。

第三阶段的目的是将源稀疏协同过滤数据集转换为其稠密表示。为完成这项任务，对于数据集中的每个源 $< user,\ item,\ rating >$ 样本（例如：$< 8920,\ 345,\ 4 >$），我们将用户ID（本例中为8920）替换为其对应的稠密表示；物品ID也同样处理。使用大小为 5 的嵌入，本例中的结果可以如下所示：

$$< [0.03, 0.94, 1.02, 0.87, -0.78], [-1.23, 0.99, 1.02, 0.65, -0.48], 4 >$$

图 1 的阶段3展示了一个示例。步骤 15 对具体操作进行了形式化描述。所得稠密数据集将在阶段 4 中用于训练一个能够生成虚假用户画像的生成对抗网络。我们的生成对抗网络将使用阶段 3 的稠密数据集，通过向判别器提供必要的真实样本来训练判别器。生成对抗网络生成器以高斯随机噪声为输入，并通过迭代学习生成越来越逼真的虚假用户画像，从而骗过判别器模型。生成器和判别器完成训练后，生成器便可将输入噪声向量转换为密集样本，这些样本能够模仿阶段 3 中提供的真实数据集的模式。阶段 4 在步骤 16 至 18 中进行了形式化描述。
图 1（左下图）中的最后一个阶段使用经过训练的生成对抗网络生成器模型（阶段 4）来生成任意数量的伪样本。我们向生成器依次输入服从高斯分布的随机噪声值向量，生成器则依次输出符合真实数据集（在阶段3中获得）模式的伪密集样本。高斯分布的标准差越大，生成的伪密集样本中各个值的多样性就越高。例如，随机噪声高斯分布的标准差较小时，评分“3”（范围为 1 到 5）的比例会较高；而选择较大的标准差值，则会产生更多评分为“5”和“1”的样本。步骤 19 对伪样本的生成过程进行了形式化描述。

尽管可以认为 GANRS 方法已经完成，但事实并非如此，因为我们的目标是生成由稀疏样本构成的伪数据集（例如 Movielens 或 FilmTrust）；因此，有必要将阶段 5 中得到的稠密表示转换为阶段 1 中常见的稀疏表示。这个过程并不

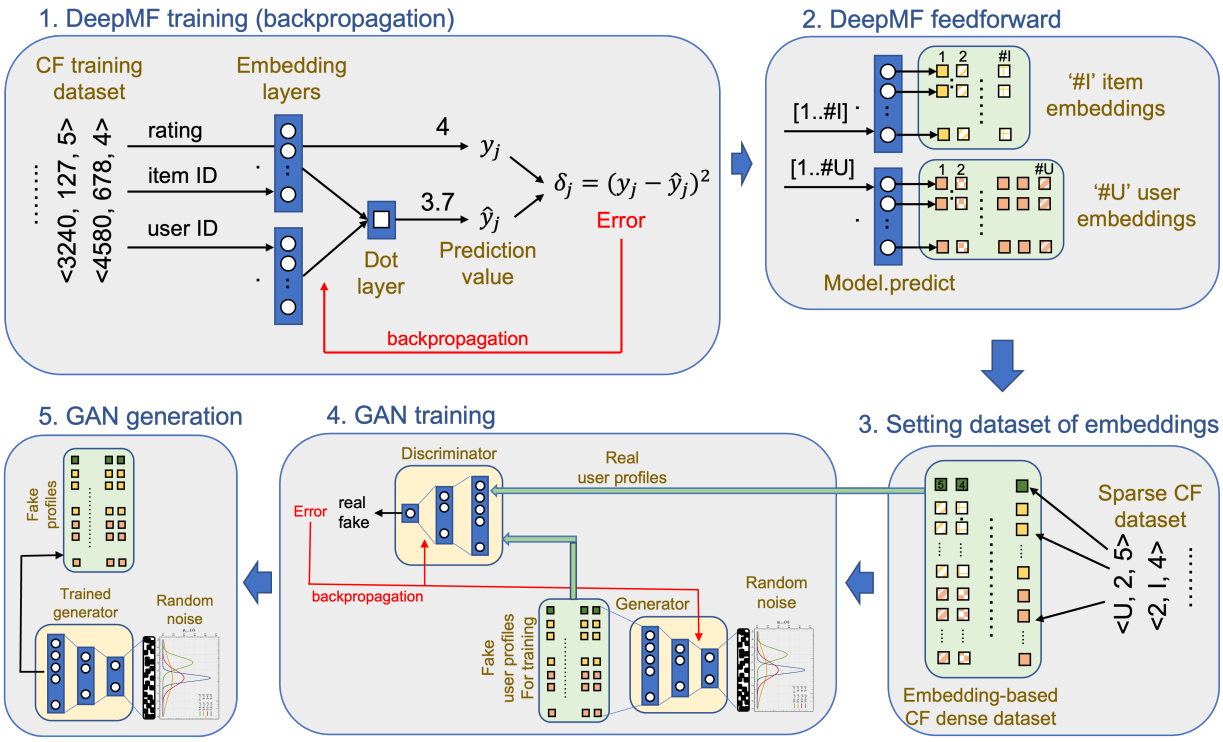



图 1：GANRS方法的主要阶段：生成稠密伪样本。

简单，因为伪样本的所有稠密表示彼此不同；表1中的示例可以更好地说明这一点：可以看到，用户 890（前两行）的稠密嵌入值非常相似，但并不完全相同，因为 GAN生成器无法根据噪声输入向量生成完全相同的值。表1中 ID 为 31 的物品也存在同样的情况。因此，GANRS 方法提供了一种将相似的稠密嵌入“归并”为唯一ID的方法；也就是说，将表1中用户的粗体稠密向量转换为一个唯一用户ID（不一定是 890），并将物品的粗体稠密向量转换为一个唯一物品ID（也不一定是 31）。

| 稀疏 | 密集型 |
|---|---|
| $<$ **890**,47, 5 $>$ | $<$ **[0.03, 0.94, 1.02, 0.87,-0.78]**, $[-1.23, 0.99, 1.02, 0.65, -0.48]$, 5 $>$ |
| $<$ **890**, **31**, 4 $>$ | $<$ **[0.02, 0.95, 0.99, 0.81,-0.69]**, **[0.45, -0.78, 0.83, -0.15, 0.09]**, 4 $>$ |
| $<$ 968,**31**, 4 $>$ | $<$ $[-1.04, 0.04, 0.66, -0.67, 0.11]$, **[0.42, -0.71, 0.80, -0.10, 0.14]**, 4 $>$ |
| $<$ 123,**31**, 2 $>$ | $<$ $[1.56, -1.12, 0.33, 1.22, -0.87]$, **[0.43, -0.75, 0.80, -0.11, 0.06]**, 2 $>$ |

表1：样本表示示例

为了将相似的稠密嵌入归入唯一ID，我们选择了 K 均值聚类 [57]。该算法有一个相关特性：必须预先选定簇的数量 K。在此情境下，这一点非常便利，因为这样我们就能确定 GANRS 合成生成数据集中用户和物品的数量。图3的阶段 6 展示了这一概念，其中 $K^*$ 被选为用户数量，$K^{**}$ 被选为物品数量。我们运行两个彼此独立的 K 均值过程：一个用于对用户嵌入分组，另一个用于对物品嵌入分组。步骤 20 至 23 对这两个聚类过程进行了形式化描述。为了更好地理解这一阶段，我们可以设想这样一个例子：已生成一百万个伪样本，我们想创建一个包含两千个虚假用户和一千个虚假物品的合成数据集。为了完成这项任务（如表1所示），我们应从一百万个用户向量中获得两千个分组（物品也同样如此，即一千个物品分组）。平均而言，每个用户分组可分配五百个用户向量（同理，每个物品分组可分配一千个物品向量），但我们知道，这取决于用户和物品向量的模式。为了恰当地完成分组任务，机器学习为我们提供了聚类算法，其中

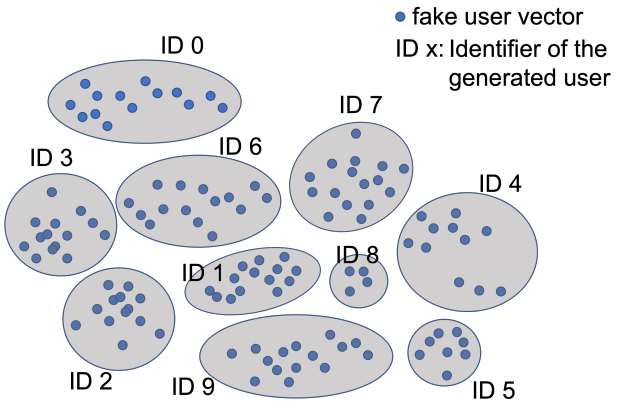



图2：将虚假用户向量（蓝色圆圈）分组，并为每个向量分配一个虚假用户ID（ID x）。灰色椭圆表示 K 均值聚类组。

K 均值使我们能够设定所需的分组数量（在本例中，用户为两千组，物品为一千组）。运行这两个聚类过程（一个针对用户，另一个针对物品）后，我们可以为每个簇中的所有虚假用户向量分配一个虚假用户ID。请注意，ID 编号可以随机分配给两千个簇中的每一个簇（物品的一千个 ID 也同样如此）。图2展示了这一概念，其中蓝色圆圈表示本例中的一百万个虚假向量，“ID 编号”文字表示所需的两千个用户。灰色椭圆表示 K 均值聚类的结果。因此：a）每组“相似”“最近”“关联更密切”的用户向量共享同一个虚假用户ID；b）所有用户向量都已分配虚假用户ID；c）每个虚假用户关联的用户向量数量可能各不相同。

图3中的第七阶段将稠密伪样本（来自阶段 5）转换为稀疏样本 $<user,\ item,\ rating>$。为完成此任务，我们将稠密表示中的每个样本的用户向量替换为其质心编号（从 1 到 $K^*$），并将其物品向量替换为其质心编号（从 1 到 $K^{**}$）；评分值保持不变。图3展示了这一操作的示例，该操作在步骤 24 中进行了形式化说明。请注意，在前述离散化过程中会出现重复样本，因为 GAN生成器可能生成非常相似的稠密样本，而这些样本会被转换为相同的离散编码。重复样本的数量受多个因素影响，例如生成样本的数量、嵌入维度、噪声向量的大小以及高斯分布的标准差，但最相关的因素是所选用户或物品的数量（$K^*$ 和 $K^{**}$）：$K$越低，重复样本的数量就越多。当用户或物品数量较少时，每个聚类中归入的样本平均数量较多。步骤 25 对移除重复离散样本的过程进行了形式化说明。最后，GANRS方法可能会生成少量样本，其中同一用户对同一物品给出不同评分；例如：$<879,\ 56,\ 4>$，$<879,\ 56,\ 5>$。这可以视为一种有益行为：编码更高范围的评分（示例中的 4.5），或表达用户意见的变化。这些情况可以保留、修改或删除。步骤 26 对其移除操作进行了形式化说明。

附录 A（表4）展示了用于设计所提出的 GANRS 方法中涉及的 DeepMF 和生成对抗网络这两个模型的主要参数和超参数值。有关详细信息，可通过分析 http://suleiman.ujaen.es:8061/gitlab-instance-981c80cc/ganrs 上提供的源代码来了解。GANRS 方法的形式化定义按照本节所述并在图1和图3中展示的七个阶段进行呈现和组织：

• 阶段0. 协同过滤定义 1 设 $U$ 为使用协同过滤推荐系统的用户集合。 2 设 $I$ 为协同过滤推荐系统中可供评分的物品集合。 3 设 $V$为允许的评分范围；通常为 $V = \{1, 2, 3, 4, 5\}$。 4 设 $S$ 为协同过滤数据集所包含的样本集合；其中 $N = |S| = thetotalnumberofcastvotes$。
5 $S = \{< u, i, v >_1, < u, i, v >_2, \ldots, < u, i, v >_N\}$；其中每个 $u \in \{1, \ldots, |U|\}$、每个 $i \in \{1, \ldots, |I|\}$以及每个 $v \in \{1, \ldots, |V|\}$.

• 阶段 1. DeepMF 训练 6 设 E 为用于将分别属于 U 和 I 的每个用户和每个物品向量化的两个神经网络层嵌入的大小。

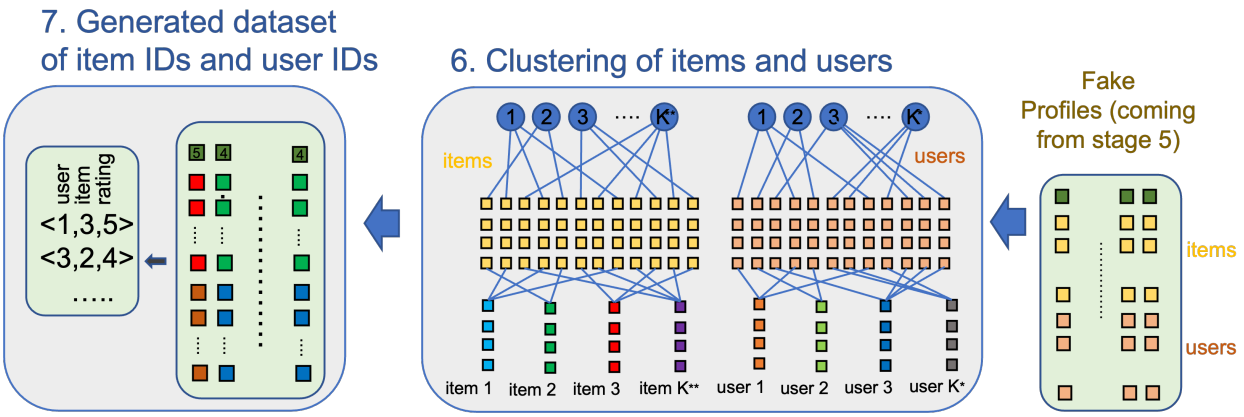



图3：所提出的GANRS方法的最后阶段：将密集样本转换为稀疏样本。

7 设 $f^{eu}(u) = [e_0^u, e_1^u, \ldots, e_E^u]$，其中 $f^{eu}$ 是用户的嵌入层输出，其中 $u \in \{1, \ldots, |U|\}$。8 设 $f^{ei}(i) = [e_0^i, e_1^i, \ldots, e_E^i]$，其中 $f^{ei}$ 是物品的嵌入层输出，其中 $i \in \{1, \ldots, |I|\}$。通过结合用户嵌入和物品嵌入的两个稠密向量：($[e_0^u, e_1^u, \ldots, e_E^u]$ 和$[e_0^i, e_1^i, \ldots, e_E^i]$)，我们可以在 DeepMF 训练阶段进行评分预测。每个 $< u, i, v >_j \in S$ 中的用户嵌入与物品嵌入的点积给出其评分预测：9 ˆ $y_j = f^{eu}(u) \cdot f^{ei}(i) = [e_0^u, e_1^u, \ldots, e_E^u] \cdot [e_0^i, e_1^i, \ldots, e_E^i]$10 1 2$(y_j - \hat{}y_j)^2$ 是 DeepMF神经网络中用于启动反向传播算法的输出误差，其中神经网络权重会根据 $\delta_j$ 值进行迭代改进：$\triangle w_{ji} = \alpha y_j f'(Net_i) \sum_k^- w_{ik}\delta_k$，当 $k$ 是隐藏层时；若 $k$ 是输出层，则为 $\triangle w_{ji} = \alpha y_i f'(Net_i)$ 1 2$(y_k - \hat{}y_k)^2$,。i、j 和 k 是依次相邻的层。

–

- 阶段2。DeepFM前馈传播
  DeepMF完成学习后，我们可以收集协同过滤推荐系统中每个用户和每个物品的嵌入表示。

  11 设 $E^* = \{< u, [e_0^u, e_1^u, \ldots, e_E^u] >, \forall u \in U\}$为所有推荐系统用户的嵌入集合。12 设 $E^*(u) = [e_0^u, e_1^u, \ldots, e_E^u]$13 设$E^{**} = \{< i, [e_0^i, e_1^i, \ldots, e_E^i] >, \forall i \in I\}$为所有推荐系统物品的嵌入集合。14 设 $E^{**}(i) = [e_0^i, e_1^i, \ldots, e_E^i]$

- 阶段3。设置嵌入数据集
  15 令 $R = [< E^*(u)\ E^{**}(i)\ v >], \forall < u, i, v >_j \in S$ 为真实样本的基于嵌入的数据集。
- 阶段 4. GAN 训练
  16 令 $f^D$ 为属于 GAN 模型的判别器 D 模型。
  17 令 $f^G$ 为属于 GAN 模型的生成器 G 模型。
  18 令 $f^{GD}$ 为 GAN 模型的优化函数；$f^{GD} = Min_G Max_D f(D, G) = E_R[log(D(R))] + E_z[log(1 - D(G(z)))]$, 其中，$E_R$ 为真实样本的期望值，$z$ 为输入生成器 $G$ 的随机噪声，$E_z$ 为生成的虚假用户画像的期望值$G(z)$。注意，$R$ 指的是 [15]。
- 阶段 5. GAN 生成
  19 令 $F = f^G(z)$ 表示由不同随机噪声向量生成的伪样本数据集 $z$.
- 阶段 6。物品和用户的聚类。
  20 令 $K^*$ 表示用于对用户嵌入进行分组的簇数量。
  21 令 $K^{**}$ 为用于对物品嵌入进行分组的簇数。
  22 令 $h^*(u) = c | c \in \{1, \ldots, K^*\}$ 表示将质心分配给每个用户的聚类操作。
  23 令 $h^{**}(i) = c | c \in \{1, \ldots, K^{**}\}$ 表示将质心分配给每个物品的聚类操作。

• 第7阶段。设置物品ID和用户ID数据集

24 令 H 表示从伪样本的基于嵌入的数据集 F 中获得的物品ID和用户ID离散数据集。 $H = \{< h^*(u)\ h^{**}(i)\ v > |\forall < E^*(u)\ E^{**}(i)\ v >\in F\}$25 令 $S = \{H\}$ 表示 H 的合成生成数据集版本，其中已移除重复样本。26 令 $G' = \{< h^*(u)\ h^{**}(i)\ v >\in H|\nexists < h^*(u')\ h^{**}(i')\ v' >\in H$，其中 $h^*(u) = h^*(u') \wedge h^*(i) = h^{**}(i) \wedge v \neq v'\}$

## 4 实验与结果

此处评估了所提出的、专注于构建合成数据集的流程的适用性。首先介绍将作为当前流程起点的传统数据集，并描述将要开展的实验。随后展示并讨论所得结果。

### 4.1 实验

为了测试所提出的 GANRS方法的行为，我们将在协同过滤领域使用三个具有代表性的开放数据集：Movielens、Netflix 和 MyAnimeList。我们选择了 Movielens 100K 版本，以及完整 Netflix 数据集的一个缩减版本：Netflix*，该版本可通过 [58] 获取。表2 展示了这些数据集的主要参数值。我们使用 Netflix* 开展了一整套实验，而对于 Movielens 和 MyAnimeList，仅展示其中一部分实验结果，以缩减论文篇幅。Movielens 和 MyAnimeList 的测试结果将在本节末尾进行总结。三个源数据集中的每一个都用于生成其对应的合成版本：设置不同数量的用户、物品和样本，并改变高斯随机噪声的标准差。

| 数据集 | #用户 | #物品 | #评分 | 分数 | 稀疏度 |
|---|---|---|---|---|---|
| Movielens 100K | 943 | 1682 | 99,831 | 1至5 | 93.71 |
| Netflix* | 23,012 | 1,750 | 535,421 | 1至5 | 98.68 |
| MyAnimeList | 19,179 | 2,692 | 548,967 | 1至10 | 98.94 |

表2：所测试数据集的主要参数值 集

表3展示了使用 Netflix* 作为源数据、用于测试所提出的 GANRS 方法的 GAN 生成合成数据集。灰色单元格显示生成的数据集数量；“std”表示随机噪声高斯分布所使用的标准差；#users 和 #items 分别表示生成每个数据集所选取的用户总数和物品总数；#samples 表示 GAN生成器创建的伪样本数量。请注意，由于移除了重复样本，每个数据集实际包含的样本数量少于 #samples。表3中的案例1至15用于测试改变标准差和用户数量所产生的影响。案例16和17用于测试增加物品数量所带来的影响。最后，案例18、19和20用于测试合成数据集在样本数量不同时的表现。所有生成的数据集以及所提出的 GANRS 方法的源代码均可通过 http://suleiman.ujaen.es:8061/gitlab-instance-981c80cc/ganrs 完整获取。此外，附录B（图16）展示了每个合成数据集所得分布图的示例。通过开放访问链接 http://suleiman.ujaen.es:8061/gitlab-instance-981c80cc/ganrs，可以在各生成数据集对应的专属目录中找到该数据集及其分布图，并且目录中还提供了一个 readme.txt 文件。

| # | std | #用户 | #物品 | # | std | #用户 | #物品 | # | std | #用户 | #物品 | #样本 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 2.0 | 100 | 4000 | 6 | 2.5 | 100 | 4000 | 11 | 3.0 | 100 | 4000 | 1.5M |
| 2 | 2.0 | 1000 | 4000 | 7 | 2.5 | 1000 | 4000 | 12 | 3.0 | 1000 | 4000 | |
| 3 | 2.0 | 2000 | 4000 | 8 | 2.5 | 2000 | 4000 | 13 | 3.0 | 2000 | 4000 | |
| 4 | 2.0 | 4000 | 4000 | 9 | 2.5 | 4000 | 4000 | 14 | 3.0 | 4000 | 4000 | |
| 5 | 2.0 | 8000 | 4000 | 10 | 2.5 | 8000 | 4000 | 15 | 3.0 | 8000 | 4000 | |
| 16 | 1.5 | 4000 | 2000 | 17 | 1.5 | 4000 | 8000 | | | | | |
| 18 | 1.2 | 2000 | 4000 | | | | | | | | | 150K |
| 19 | 1.2 | 2000 | 4000 | | | | | | | | | 500K |
| 20 | 1.2 | 2000 | 4000 | | | | | | | | | 1M |
| 21 | 1.2 | 2000 | 4000 | | | | | | | | | 3M |

表3：生成对抗网络生成的合成数据集的参数值。来源：Netflix*。

使用表3中的参数值，我们开展了多种实验。实验分类如下：

1. 用户数量
   (a) 用户与评分的分布 (b) 用户评分的分布 (c) 重复样本数量 (d) 用户和项目相同的样本比例 (e) 数据集的平均绝对误差和准确率 (f) 用户的精确率和召回率

2. 项目数量 (a) 数据集的平均绝对误差和准确率 (b) 物品的精确率和召回率 3. 样本数量 (a) 生成的样本数量 (b) 精确率和召回率

### 4.2 结果

本小节展示了运行设计好的实验（见上一小节）后得到的图表。我们使用表3中描述的合成数据集来获取结果，以便：1）比较源数据集中用户、物品和评分的分布与使用合成数据集得到的分布；2）测量聚类阶段返回的重复样本数量；3）测试运行所提出的 RSGAN 方法后得到的预测和推荐质量及趋势，并将其与源数据集所呈现的结果进行比较。

#### 4.2.1 实验 1a。用户数量：用户数量与评分数的分布

图4展示了投出不同数量评分（x 轴）的用户密度（y 轴）。（所选数据集：表3中的 8 和 10。）正如预期的那样，在数据集中的评分总数固定时，可以观察到，用户数量越多，评分数量越少。如果数据集中的固定样本数量分布给大量用户，那么聚类阶段中每个用户质心接收到的样本数量就越少。请注意，Netflix* 包含 23000 名用户。

#### 4.2.2 实验 1b。用户评分的分布

图5展示了应用不同随机噪声高斯分布时，各评分值 1,...,5（x 轴）所占的比例。（所选数据集：3、8 和 13。）可以观察到，与相邻的 2 和 2.5 标准分布相比，标准差为 2.5 时生成的评分分布与原始 Netflix* 数据集中的分布更为相似。图5还展示了高斯标准差对生成对抗网络生成样本中各个值布局的影响。

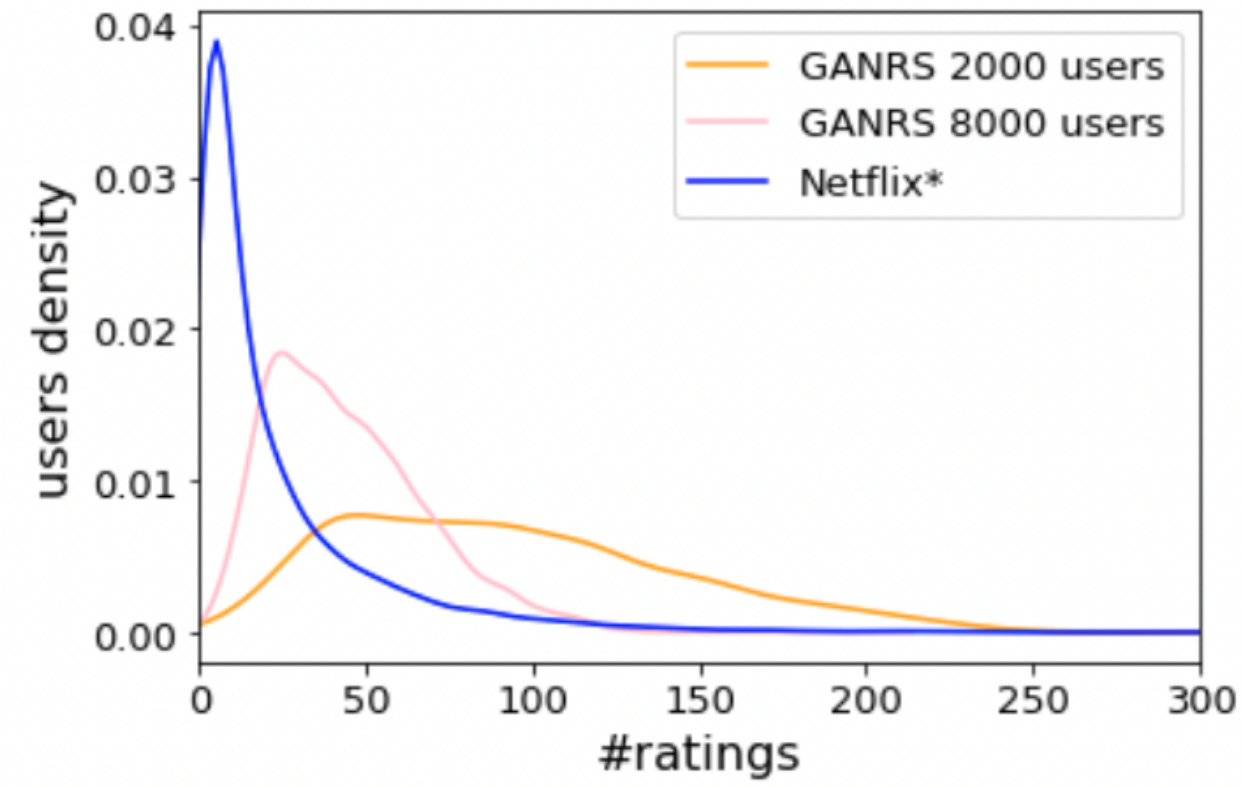
#### 4.2.3 实验1c。重复样本数量

**量**

正如“方法”部分所述，经过训练的生成器可根据随机噪声向量预测出任意数量的稠密样本；随后，所有这些样本都会从连续稠密值转换为离散稀疏值。在离散化过程中，会出现重复样本，必须将其移除（表1给出了一个示例）。图6展示了移除过程后数据集中剩余的样本数量。用户数量越少，分配给每个用户的样本数量就越多（在聚类过程中分配到其质心），因此出现重复离散样本的概率也越高。总体而言，用户数量越少，剩余的样本数量就越少。选定的数据集：表3中的6至10。

#### 4.2.4 实验1d。具有相同用户和物品的样本比例

GANRS生成的数据集具有一个源数据集（Movielens等）不存在的属性：其中有一定比例的样本显示，同一用户对同一物品给出了不同的评分；例如：$< 348, 90, 5 >,< 348, 90, 4 >$，正如“方法”部分所述。这可以被视为一种允许中间评分的机制（4.5




Fig图4：用户数量与评分的分布。项目数量：4000。表3中的数据集8和10 .

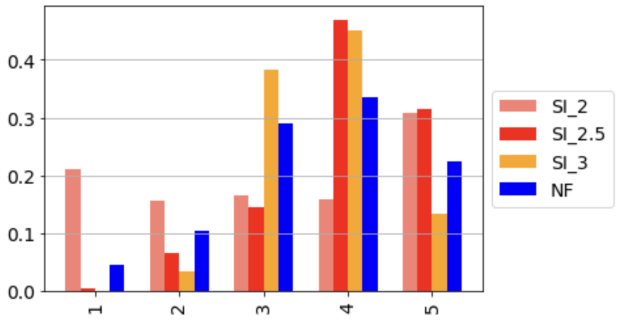



图5：用户评分的分布。用户数量：2000；项目数量：4000。表3中的数据集3、8和13。

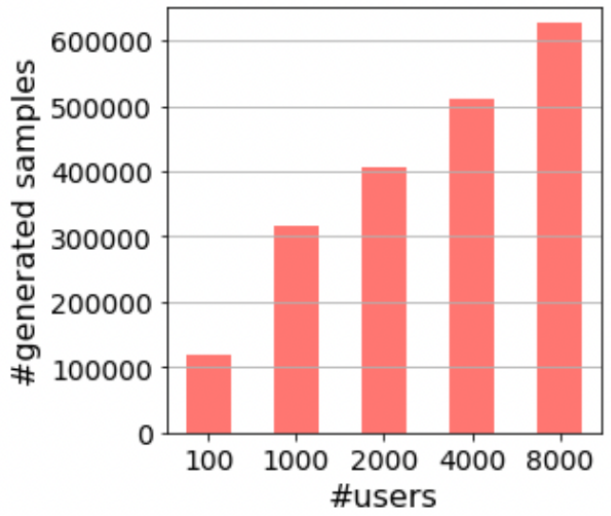



图6：移除重复样本后剩余的样本数量。物品：4000。表3中的数据集6至10。

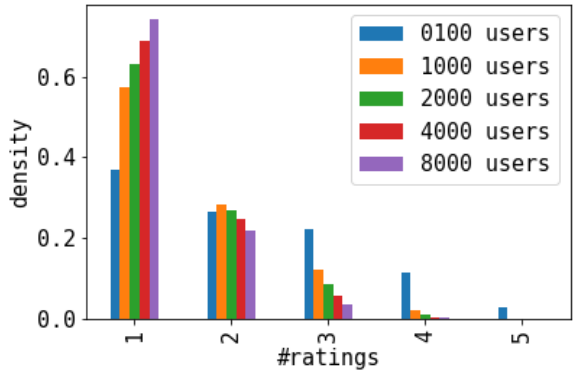
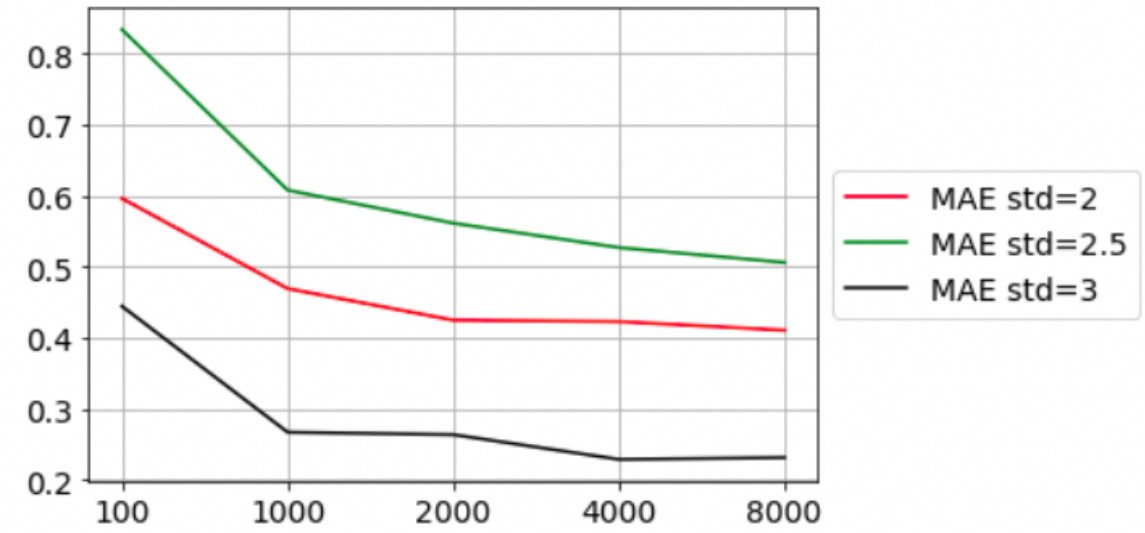



图7：同一用户对同一物品给出不同评分的样本比例。物品：4000。表3中的数据集6至10。

（如示例所示），或者允许用户改变主意。这是合理的，因为重复评分的次数为两次或三次。我们应当更好地移除少数出现的四次或五次重复评分，就像我们在所有生成的数据集中所做的那样。图7显示，对于常规的协同过滤推荐系统（用户数量达到1000或更多），重复四次或五次的比例并不显著；随着用户数量增加，重复评分的比例会迅速下降。

#### 4.2.5 实验1e：数据集的平均绝对误差和准确率

前面的实验分析了合成数据集的内部构成和分布，而本实验及下一个实验则测试生成的数据集在预测和推荐任务中的表现。图8展示了表3中数据集1至15的每组独立样本所得到的预测质量（平均绝对误差）和推荐准确率。请注意，这些指标并非通过分析并对用户结果取平均而得。图8中的图表显示，随着用户数量增加，准确率有所提高（相应的平均绝对误差也随之降低）。这符合协同过滤推荐系统中的预期情况：用户数量越多，预测效果越好；这也表明生成对抗网络生成的样本符合协同过滤的适用模式。图8上方图表中的平均绝对误差值与标准差为2.0、2.5和3.0时的评分分布密切相关。平均绝对误差/准确率结果可用于选择最合适的标准差；本例中为：标准差 = 2.5。



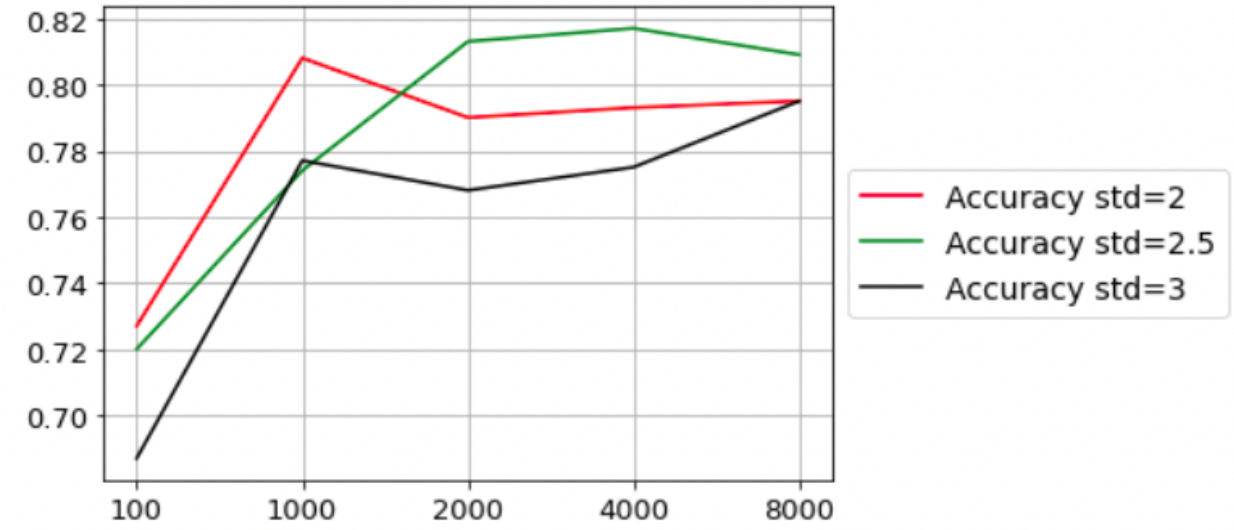
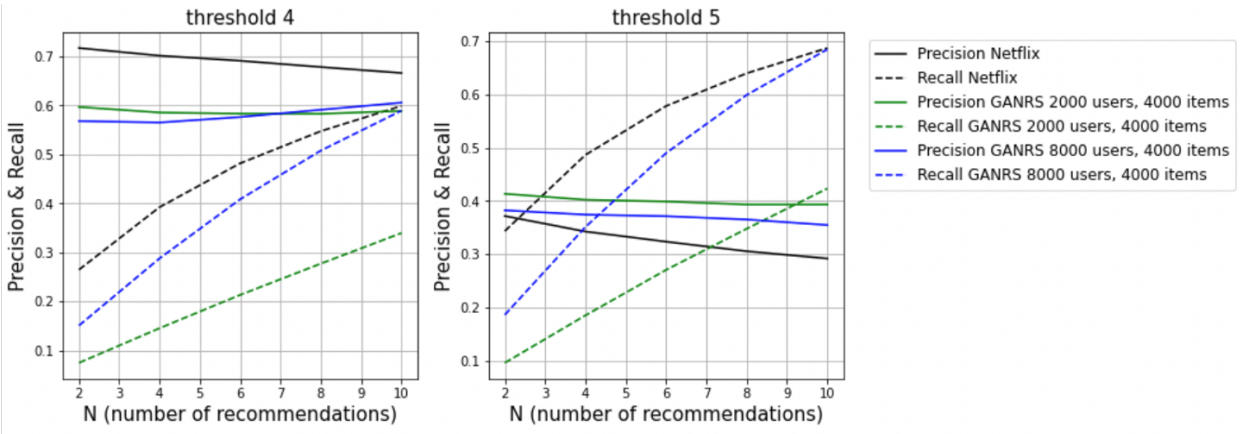




图8：平均绝对误差和准确率。用户数量：2000；项目数量：4000；“标准差”是高斯分布随机噪声分布。表3中的数据集1至15。



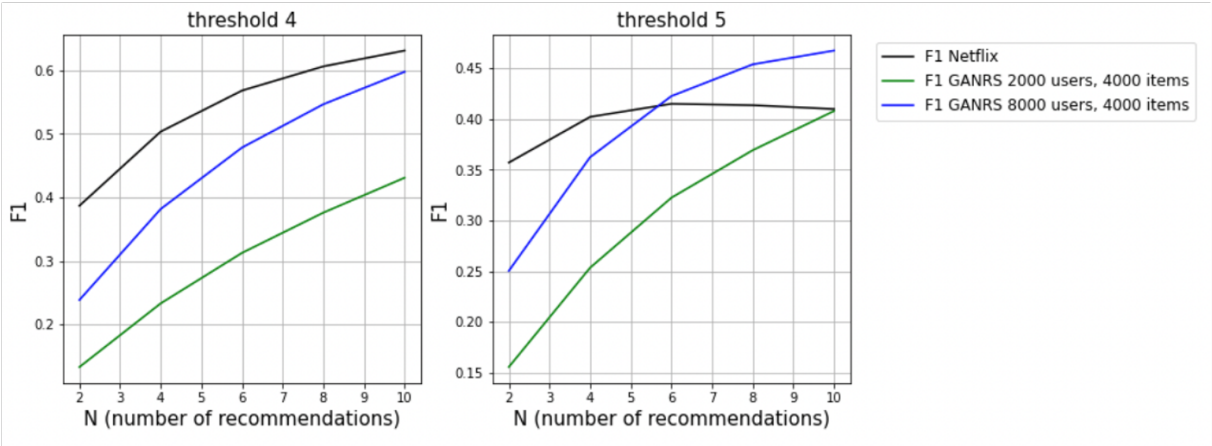




图9：精确率、召回率和F1分数。随机噪声高斯分布的标准差：2.5。推荐数量 N = [2,4,6,8,10]。表3中的数据集6至10。

### 4.2.6 实验1f。用户的精确率和召回率

本实验提供了测试生成的数据集的最重要结果：我们绘制了两个具有代表性的推荐质量指标——精确率和召回率的数值及变化趋势。图9上方的图表展示了测试不同推荐数量 N 时获得的质量数值： [2,4,6,8,10] （横轴）、两个不同的相关性阈值 $\theta$： [4,5], , 以及两种用户数量：2000（绿色线条）和8000（蓝色线条）。高斯随机噪声的标准差设为2.5。所选数据集：表3中的数据集6至10。合成数据集所得的数值及变化趋势与源数据集Netflix*（黑色线条）相符。此外，如预期所示，8000用户生成数据集的总体结果优于2000用户生成数据集，并且与Netflix*参照数据更加相似（请注意，Netflix*包含23000名用户）。图9下方的两幅图表示精确率和召回率的F1组合；它们清楚地展示了生成的数据集与源数据集在行为上的相似性。

### 4.2.7 实验2a：项目数量变化时的平均绝对误差和准确率。

实验1e在用户数量不同的数据集上测试了平均绝对误差和准确率这两项质量度量。现在，我们在物品数量不同的数据集上测试这两项质量度量： [100, lK、2K、4K、8K]。图10中的结果显示，平均绝对误差和准确率的数值都令人满意，且变化趋势一致：随着物品数量（x 轴）增加，准确率上升，平均绝对误差下降。因此，物品数量越多，准确率越高：这表明 GAN生成器能够丰富数据。Netflix*源数据集包含 1,750 个物品，我们可以从图10中观察到，性能提升在物品数量接近这一数值时（x 轴）开始放缓。选用的数据集：表3中的 16 和 17，以及未列入表3的 100、1000、4000 用户版本。

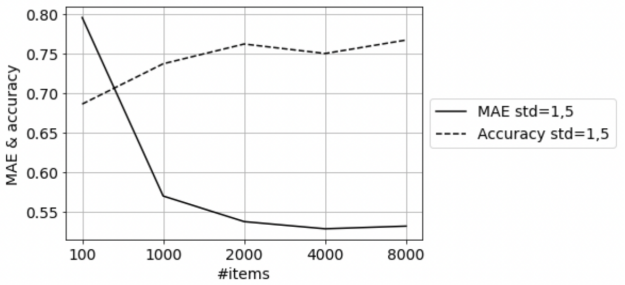



图10：物品数量变化时，数据集样本的平均绝对误差和准确率。用户数量：4000。高斯随机噪声的标准差：1.5。表3中的数据集16和17。

#### 4.2.8 实验2b：物品的精确率和召回率

实验2b与实验1f类似；现在我们测试包含不同物品数量的数据集的表现（而非包含不同用户数量的数据集）。图11展示了 Netflix*（1750 个物品）的表现（以黑线表示），并将其与包含 2000 个物品的数据集（绿线）和包含 8000 个物品的数据集（蓝线）进行比较。我们可以看到，其变化趋势和数值与源数据集（黑线）一致；此外，2K 和 4K 物品版本的表现都不错：前者恰当地捕捉了 Neflix*的物品模式，因为两者的物品数量相近。第二个生成数据集（8K 物品）能够丰富数据，且准确率高于 2K 物品版本。选用的数据集：表3中的 16 和 17，以及未列入表 3 的 100 、 1000 、 4000 用户版本。

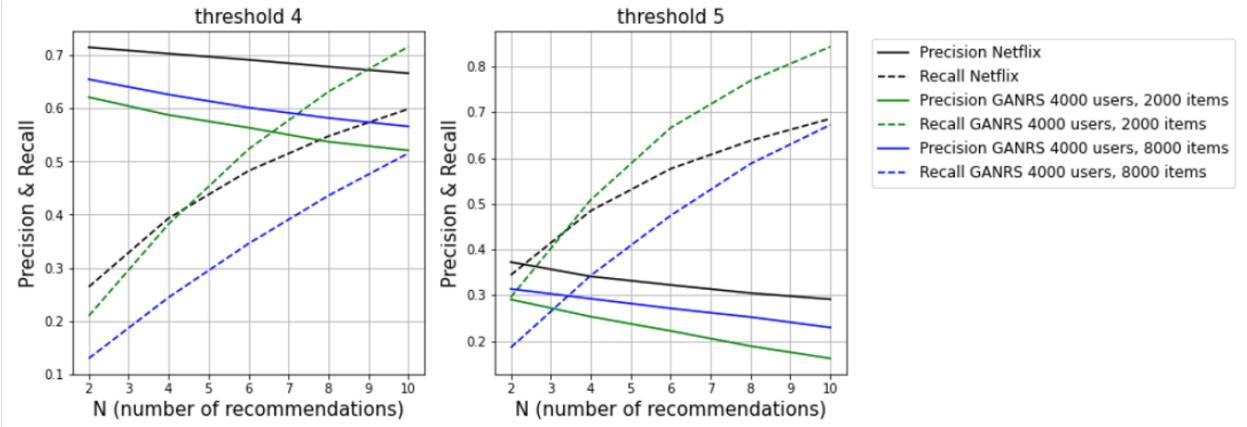
#### 4.2.9 实验3a：不同规模数据集中生成的样本数量

在此，我们测试当设定不同的生成样本数量和不同的用户数量时，GANRS方法获得的样本数量。为此，我们在GAN生成过程中设定了四种不同的样本数量：150K、500K、1M和3M（表3中的数据集18至21，以及对应用户数量为100、1000、4000和8000的数据集）。所有实验中的物品数量固定为4K。图12显示，用户数量越少，生成样本数量也越少；这是因为用户数量越少，分配给每个用户（聚类阶段中的每个质心）的样本数量就越多，因此产生重复样本并将其移除的概率也越高。例如，图12显示，用户数量为8K的数据集保留了约1M个GAN生成的样本（3M版本），而在1M版本中则保留了600K个样本。

#### 4.2.10 实验3b：不同规模数据集上的精确率和召回率

本实验展示了在参数固定的数据集上增加样本数量所产生的影响。本例中，参数为：2000名用户、4000个物品，标准差为1.2（表3；数据集18、19和21）。需要注意的是，我们使用同一个源数据集 Netflix* 生成图13所示的三种情况：15万样本（黄色线条）、50万样本（洋红色线条）和300万样本（红色线条）。请注意，15万、50万和300万样本指的是密集连续生成样本，即尚未经过移除阶段将其转换为稀疏离散版本之前的样本。图12展示了2000名用户的数据中各数据集最终的规模（横轴）。
图13比较了 Netflix* 数据集（黑色线条）与生成数据集所获得的精确率和召回率。总体而言：1）精确率上升，召回率下降；2）生成数据集越大，精确率越高；3）数据集越大，召回率越低。使用大型数据集时，精确率会提高，因为可供选择的相关样本更多，因此更容易在固定数量 N 的推荐预测中取得成功。另一方面，使用大型数据集时，召回率会变差，因为样本的可变性更大，尤其是在随机噪声高斯分布采用较大标准差时。与分母为常数 N（推荐数量）的精确率不同，召回率这一质量指标取决于变量“每位受测用户的测试项目集合中的相关评分数量”。随着样本数量增加，每位用户的评分数量也会增加（其中相关评分的数量也会增加）；这就是图13中300万样本的合成数据集召回率较低，而15万样本版本召回率较高的原因。

图 14 和图 15 分别展示了从 MyAnimeList 和 Movielens 100K 测试数据集中获得的结果。图表“a)”比较了每个源数据集的评分分布（蓝色）与通过设置不同高斯随机噪声标准差值所生成的评分分布。我们为 MyAnimeList 选择了 1.2 的标准差值，为 Movielens 100K 选择了 2.5 的标准差值，因为它们获得了



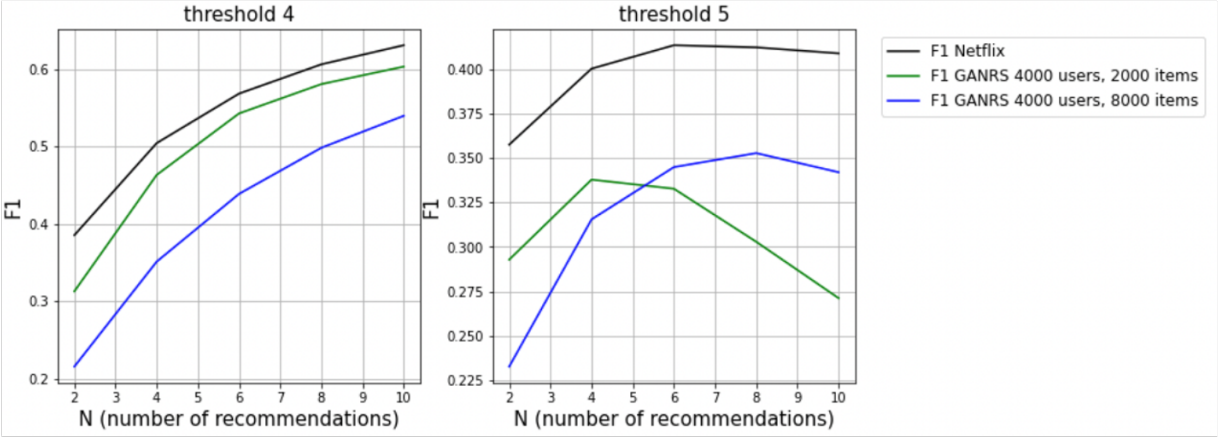




图 11：项目数量变化时的精确率、召回率和 F1分数。随机噪声高斯分布的标准差：1.5。推荐数量 N = [2,4,6,8,10]。表3中的数据集 16 和 17。

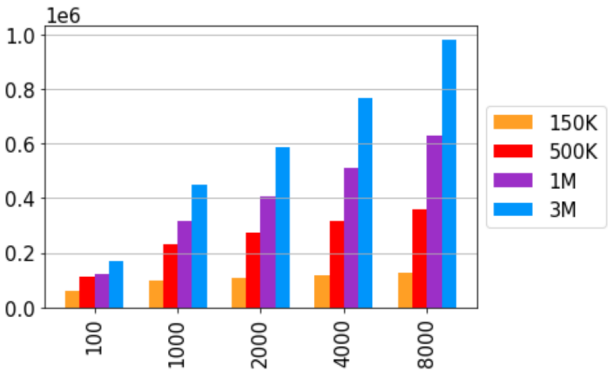



图 12：使用不同用户数量（横轴）和不同生成对抗网络生成样本数量（图例）时的生成样本数量。随机噪声高斯分布的标准差：1.2。项目数量：4000。表3中的数据集 18 至 21。

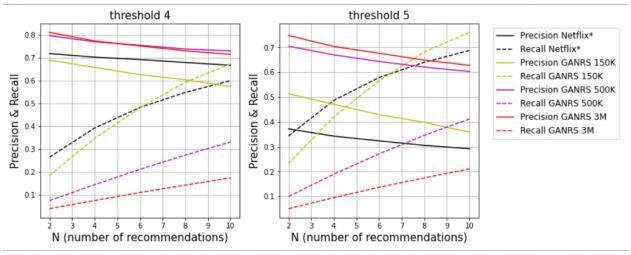



图13：使用不同数量的推荐（横轴）和不同数量的GAN生成样本（图例）时的精确率和召回率。随机噪声高斯分布的标准差：1.2。表3中的数据集18、19和21。

评分分布与各自基线最为接近。结果“b)”、“c)”和“e)”使用选定的标准差值获得。图“b)”显示了根据给出的评分数量（横轴）划分的用户分布。正如预期，它们遵循与Netflix*相同的模式。为比较结果，请注意，MyAnimeList数据集包含19179个用户，而Movielens 100K包含943个用户。图“c)”显示了移除重复实例后剩余的样本数量。用户数量越多，生成包含相同用户ID、物品ID和评分的样本的概率越低。在MyAnimeList的情况下，我们从150万个生成样本开始，而对于Movielens，我们选取了100万个生成样本。图“d)”涉及处理每个数据集所包含的单个样本后得到的平均绝对误差和准确率值。与CF语境中的通常情况一样，用户数量越多，误差越低，准确率越高。最后，图“e)”测试了处理每个数据集中的用户后得到的推荐。与Netflix*的情况一样，与基线相比，精确率提高，而召回率变差。

### 4.3 讨论

我们生成了大量合成数据集，用于测试所提出的 GANRS方法的性能。这些数据集通过为该方法的主要参数设置不同取值而创建，包括用户数量、物品数量、高斯随机噪声的变化以及生成样本数量。为使本文结论具有普适性，我们选取了三个开放且具有代表性的协同过滤数据集作为生成过程的来源。最后，我们对生成的数据集进行了多种质量度量测试，其中精确率和召回率最为重要。一个关键问题是，我们无法像检验常见的伪造人脸那样，通过目视检查来评估生成样本的质量；事实上，在协同过滤领域，我们只能通过将生成数据集的协同过滤质量结果与真实协同过滤数据集通常取得的结果进行比较，才能恰当地检验其质量。因此，我们将设计的实验重点放在测试精确率和召回率这两项质量度量上：使用包含不同用户数量、不同物品数量以及不同样本数量（规模）的数据集。在所有情况下，相较而言，我们都取得了出色的精确率结果和中等的召回率。总体而言，在协同过滤领域，这可以视为积极结果，因为精确率误差后果严重，而召回率误差则不那么重要：推荐一趟你不会喜欢的旅行（抱歉，概不退款！），比没有推荐一趟你可能会喜欢的旅行更糟。请注意，这与用于检测恶性肿瘤的深度学习模型恰好相反：精确率出错（未能及早发现肿瘤）比召回率出错（误判为存在肿瘤）后果更严重。

此外，实验表明，标准差对结果质量有着显著影响。GAN网络的学习基于一个包含噪声值的向量，这些噪声值作为种子，用于在合成数据集中生成不同的样本。每个“伪造”样本都由“噪声”向量中的随机值列表生成。按照 GAN 领域的惯例，随机值取自均值为 0、标准差为 1 的高斯分布。每个生成的样本都包含稠密物品表示、稠密用户表示，以及一个用于编码用户对物品评分的独立值。GAN 完成学习后，其生成模型便可通过前馈过程，从每个不同的随机噪声向量出发，生成任意数量的样本。实验结果表明，使用标准差为 1 的高斯分布会使评分集中在评分范围的中间值：在 1 到 5 分的范围内（Movielens 和 Netflix），评分多为 3 分及其相近分值；在 1 到 10 分的范围内（MyAnimeList），评分多为 5 分及其相近分值。本节中的多项实验表明，我们可以调节随机噪声高斯分布的标准差，以通过前馈过程生成更广泛的评分。正如预期，标准差越大，评分范围也会相应扩大。

最后，所展示的实验揭示了为生成对抗网络生成的伪样本数量与最终数据集中所含样本数量之间的关系。正如“方法”一节所述，

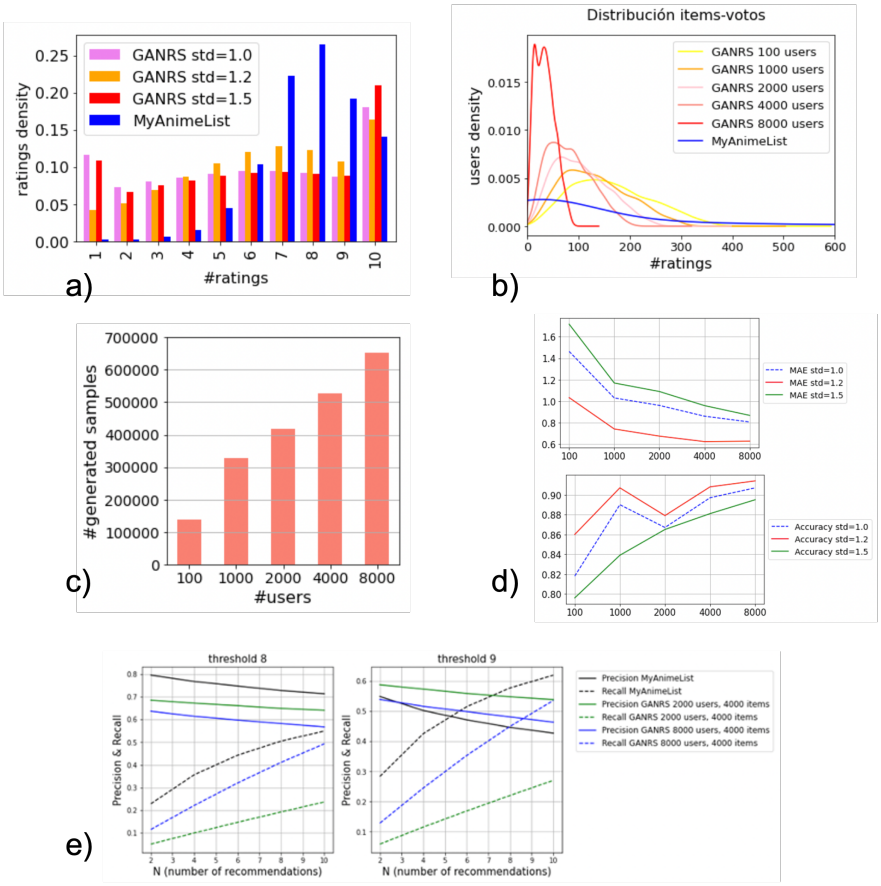



图14：MyAnimeList。生成了150万个样本，a）MyAnimeList评分（1至10分）的分布，b）用户按其给出的评分数量划分的分布，c）移除重复样本后剩余的样本数量，d）处理数据集样本所得的误差和准确率，e）协同过滤精确率和召回率（通过测试数据集中的用户计算）。GANRS std=1.2 值已用于测试实验 b）至 e）。

从稠密连续值转换为稀疏离散值会导致样本重复的概率。结果表明，正如预期，合成数据集中的用户和物品数量越多，重复样本的数量就越少。研究还表明，对于常见的用户数量，例如 4000 个或更多，用户对同一物品给出两个以上不同评分的概率可以忽略不计。

## 5 结论

本文提出了一种创新方法，可从真实数据集中生成参数化的协同过滤合成数据集。通过选择不同的用户数量、物品数量、样本数量和分布可变性，可以生成合成数据集。这意味着可以基于一整“系列”生成数据集设计对比实验，例如，在用户数量增加时测试一种新矩阵分解模型的准确率。本文使用生成对抗网络从真实样本中获取“伪”样本，利用了 GAN 网络捕捉源数据集中复杂模式的内在能力。GAN 从物品和用户的稠密连续嵌入表示中学习，而非从协同过滤数据集的稀疏离散表示中学习。其效果是实现快速且准确的学习过程。

所提出的 GANRS方法包含一个聚类阶段，用于将密集生成的“虚假”样本转换为填充生成数据集所需的稀疏离散值。该聚类阶段使用 k均值算法对物品进行分组，并使用另一种 k均值算法对用户进行分组。自然地，两个参数“k”分别设定了数据集中的用户数量和物品数量。离散化过程的一个缺点是会生成重复样本，而我们的方法会将其删除。我们使用三个具有代表性的源数据集进行了一系列完整的实验。我们测试了结果的分布值和变化情况，以及预测和推荐质量。

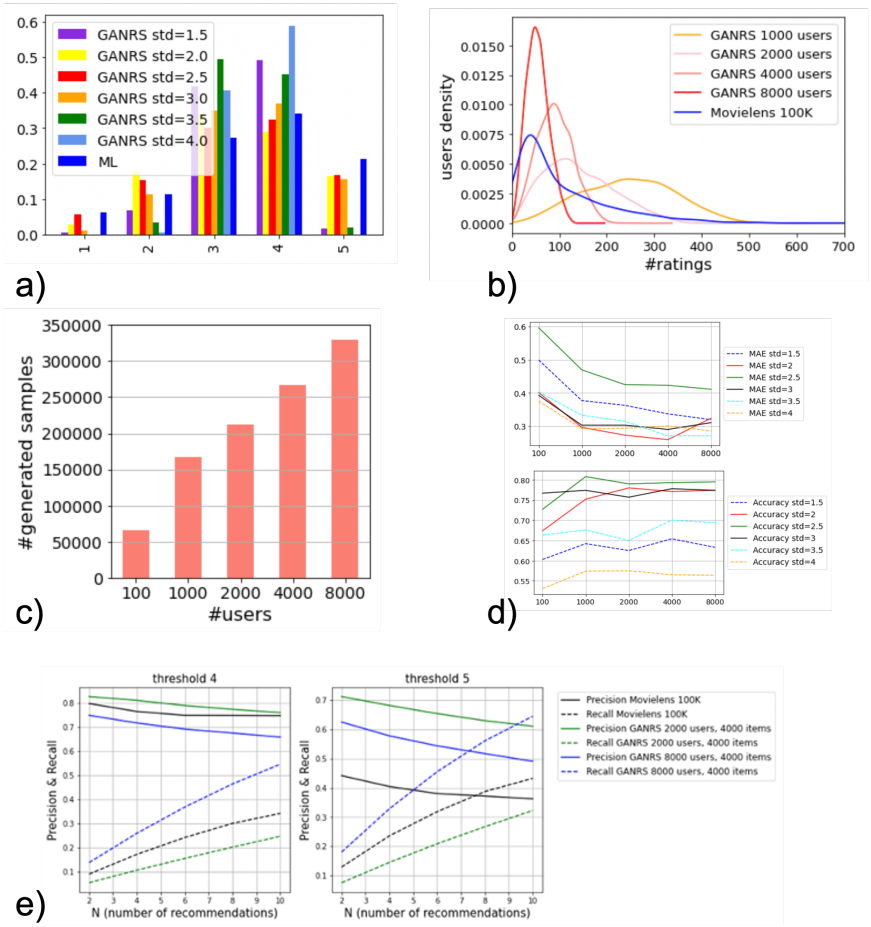



图 15：Movielens 100K 结果。生成 100 万个样本，a) Movielens 100K 评分 1 到 5 的分布，b) 根据用户给出的评分数量划分的用户分布，c) 移除重复样本后剩余的样本数量，d) 处理数据集样本时的误差和准确率，e) 协同过滤精确率和召回率（通过测试数据集用户）。GANRS std=2.5 值已设置用于测试实验 b) 至 e)。

尽管精确率往往会提高，而召回率往往会下降，但总体准确率仍可认为是理想的，因为在推荐系统场景中，精确率比召回率更重要。结果表明，生成的数据集能够很好地模拟源数据集的行为。Movielens、MyAnimeList 等。

为确保实验的可复现性，已提供所提出的 GANRS方法的源代码。同时，也提供了一整套生成的数据集供研究使用。本文及其相关文档为开展一些未来工作提供了可能，例如设计聚类阶段的替代方案、在 GAN判别器中实现 PacGAN 概念、使用完整范围的机器学习和深度学习协同过滤模型测试生成的数据集、将 GAN 模型替换为条件生成对抗网络（CGAN）模型、生成在人口统计学方面保持平衡的数据集，以及深入研究随机噪声向量的变化对生成样本集的影响。

致谢：本研究部分经费由西班牙科学与创新部通过项目 PID2019-106493RB-I00 (DL-CEMG) 提供；马德里自治区通过与马德里理工大学签订的多年期协议，在大学教师卓越计划框架下提供支持；此外，安达卢西亚研究、发展与创新计划（PAIDI 2020）通过项目 PROYEXCEL_00257 提供支持。

## 参考文献

[1] Zhiyuan Fang、Lingqi Zhang 和 Kun Chen。基于行为挖掘的混合推荐系统。载于《2016 IEEE 大数据分析国际会议（ICBDA）论文集》，第 1–5 页。IEEE，2016。[2] Raciel Yera 和 Luis Martínez。推荐系统中的模糊工具：综述。《International Journal of Computational Intelligence Systems》，10(1)：776–803，2017。[3] Raciel Yera、Ahmad A Alzahrani 和 Luis Martínez。具有聚合函数动态选择机制的模糊内容型群组推荐系统。《International Journal of Approximate Reasoning》，150：273–296，2022。[4] Lei Zheng、Vahid Noroozi 和 Philip S Yu。利用评论对用户和物品进行联合深度建模以实现推荐。载于《第十届 ACM 网络搜索与数据挖掘国际会议论文集》，第 425–434 页，2017。[5] Yuyun Gong 和 Qi Zhang。利用基于注意力的卷积神经网络进行话题标签推荐。载于 IJCAI，第 2782–2788 页，2016。[6] Hira Kanwal、Muhammad Assam、Abdul Jabbar、Salman Khan 等。基于卷积神经网络和主题建模的混合推荐系统。《International Journal of Advanced Computer Science and Applications》，11(7)，2020。[7] Kevin McNally、Michael P O'Mahony 和 Barry Smyth。社交推荐系统中基于协作的声誉模型比较研究。《User Modeling and User-Adapted Interaction》，24(3)：219–260，2014。[8] Norha M Villegas、Cristian Sánchez、Javier Díaz-Cely 和 Gabriel Tamura。情境感知推荐系统的特征分析：系统性文献综述。《Knowledge-Based Systems》，140：173–200，2018。[9] M Moradi 和 J Hamidzadeh。考虑不完整数据的基于集成的 top-k 推荐系统。《Journal of AI and Data Mining》，7(3)：393–402，2019。[10] Mahdi Jalili、Sajad Ahmadian、Maliheh Izadi、Parham Moradi 和 Mostafa Salehi。协同过滤推荐算法评估：综述。《IEEE Access》，6：74003–74024，2018。[11] Bo Zhu、Remigio Hurtado、Jesus Bobadilla 和 Fernando Ortega。一种基于评分的数值相关性和非数值结构的高效推荐系统方法。《IEEE Access》，6：49935–49954，2018。[12] Raciel Yera、Ahmad A Alzahrani 和 Luis Martínez。探索用于可解释烹饪食谱推荐的事后无关模型。《Knowledge-Based Systems》，251：109216，2022。[13] Edoardo D'Amico、Giovanni Gabbolini、Cesare Bernardis 和 Paolo Cremonesi。分析并提升推荐系统矩阵分解的稳定性。《Journal of Intelligent Information Systems》，58(2)：255–285，2022。[14] Mehdi Hosseinzadeh Aghdam。一种基于用户与物品成对关系、用于推荐系统的新型约束非负矩阵分解方法。《Expert Systems with Applications》，195：116593，2022。[15] Gonul Ayci、Abdullatif Köksal、M Melih Mutlu、Burak Suyunu 和 A Taylan Cemgil。利用贝叶斯非负矩阵分解进行推荐系统主动学习。载于《2019年第27届信号处理与通信应用会议（SIU）论文集》，第 1–4 页。IEEE，2019。[16] Jesús Bobadilla、Rodolfo Bojorque、Antonio Hernando Esteban 和 Remigio Hurtado。利用贝叶斯非负矩阵分解进行推荐系统聚类。《IEEE Access》，6：3549–3564，2017。[17] Hong-Jian Xue、Xinyu Dai、Jianbing Zhang、Shujian Huang 和 Jiajun Chen。面向推荐系统的深度矩阵分解模型。载于 IJCAI，第 17 卷，第 3203–3209 页。澳大利亚墨尔本，2017。[18] Xiangnan He、Lizi Liao、Hanwang Zhang、Liqiang Nie、Xia Hu 和 Tat-Seng Chua。神经协同过滤。载于《第26届万维网国际会议论文集》，第 173–182 页，2017。[19] Jesus Bobadilla、Raul Lara-Cabrera、Angel Gonzalez-Prieto 和 Fernando Ortega。DeepFair：利用深度学习提升推荐系统的公平性。《International Journal of Interactive Multimedia and Artificial Intelligence》，6(6)：86–95，2021。[20] Yassine Himeur、Abdullah Alsalemi、Ayman Al-Kababji、Faycal Bensaali、Abbes Amira、Christos Sardianos、George Dimitrakopoulos 和 Iraklis Varlamis。建筑能效推荐系统综述：原理、挑战与前景。《Information Fusion》，72：1–21，2021。[21] Jesús Bobadilla、Jorge Dueñas、Abraham Gutiérrez 和 Fernando Ortega。神经协同过滤推荐系统中的深度变分嵌入表示。《Applied Sciences》，12(9)：4168，2022。

[22] Jesús Bobadilla、Ángel González-Prieto、Fernando Ortega 和 Raúl Lara-Cabrera。利用深度学习方法获取协同过滤邻域。Neural Computing and Applications，34(4):2939–2951，2022。[23] Shuai Zhang、Lina Yao、Aixin Sun 和 Yi Tay。基于深度学习的推荐系统：综述与新视角。ACM Computing Surveys (CSUR)，52(1):1–38，2019。[24] Ian Goodfellow、Jean Pouget-Abadie、Mehdi Mirza、Bing Xu、David Warde-Farley、Sherjil Ozair、Aaron Courville 和 Yoshua Bengio。生成对抗网络。Communications of the ACM，63(11):139–144，2020。[25] Dimitris Sacharidis。基于社交的协同过滤中的多样性与新颖性。载于第27届ACM用户建模、自适应与个性化会议论文集，第139–143页，2019。[26] Anupriya Gogna 和 Angshul Majumdar。Diablo：一种基于优化的设计方法，用于提升推荐系统的多样性。Information Sciences，378:59–74，2017。[27] Jesus Bobadilla、A Gutierrez、S Alonso 和 Ángel González-Prieto。用于获取预测可靠性的神经协同过滤分类模型。International Journal of Interactive Multimedia and Artificial Intelligence，7(4):18–26，2022。[28] Francisco Pajuelo-Holguera、Juan A Gómez-Pulido 和 Fernando Ortega。评估推荐系统测试数据集的选择策略。载于混合人工智能系统国际会议论文集，第243–253页。Springer，2019。[29] Kurt D Bollacker、Steve Lawrence 和 C Lee Giles。Citeseer：用于自动检索和识别有趣出版物的自主网络代理。载于第二届自主代理国际会议论文集，第116–123页，1998。[30] Worasit Choochaiwattana。利用标签标注进行研究论文推荐。载于2010年第三届先进计算机理论与工程国际会议（ICACTE），第2卷，第V2–439页。IEEE，2010。[31] Jyoti Shokeen 和 Chhavi Rana。社交推荐系统：技术、领域、指标、数据集与未来展望。Journal of Intelligent Information Systems，54(3):633–667，2020。[32] Yujie Xing、Itishree Mohallick、Jon Atle Gulla、Özlem Özgöbek 和 Lemei Zhang。面向推荐系统的教育新闻数据集。载于欧洲机器学习与数据库知识发现联合会议论文集，第562–570页。Springer，2020。[33] Fernando Ortega、Jesús Bobadilla、Abraham Gutiérrez、Remigio Hurtado 和 Xin Li。用于推荐系统的人工智能科学文献数据集。IEEE Access，6:48543–48555，2018。[34] Dawen Liang、Rahul G Krishnan、Matthew D Hoffman 和 Tony Jebara。用于协同过滤的变分自编码器。载于2018年万维网大会论文集，第689–698页，2018。[35] Siamak Zamany、Dingcheng Li、Hongliang Fei 和 Ping Li。深入理解用于二元协同过滤的变分自编码器。载于2022年ACM SIGIR信息检索理论国际会议论文集，第254–263页，2022。[36] Min Gao、Junwei Zhang、Junliang Yu、Jundong Li、Junhao Wen 和 Qingyu Xiong。基于生成对抗网络的推荐系统：问题驱动视角。Information Sciences，546:1166–1185，2021。[37] Yashar Deldjoo、Tommaso Di Noia 和 Felice Antonio Merra。对抗式推荐系统综述：从攻击/防御策略到生成对抗网络。ACM Computing Surveys (CSUR)，54(2):1–38，2021。[38] Dong-Kyu Chae、Jin-Soo Kang、Sang-Wook Kim 和 Jung-Tae Lee。CFGAN：一种基于生成对抗网络的通用协同过滤框架。载于第27届ACM信息与知识管理国际会议论文集，第137–146页，2018。[39] Zongwei Wang、Min Gao、Xinyi Wang、Junliang Yu、Junhao Wen 和 Qingyu Xiong。面向推荐的生成式与判别式样本模型的极小极大博弈。载于亚太知识发现与数据挖掘会议论文集，第420–431页。Springer，2019。[40] Wei Zhao、Benyou Wang、Jianbo Ye、Yongqiang Gao、Min Yang 和 Xiaojun Chen。PLASTIC：在Top-N推荐中通过对抗训练优先利用长短期信息。载于IJCAI，第3676–3682页，2018。[41] Homanga Bharadhwaj、Homin Park 和 Brian Y Lim。RecGAN：用于推荐系统的循环生成对抗网络。载于第12届ACM推荐系统会议论文集，第372–376页，2018。[42] Guibing Guo、Huan Zhou、Bowei Chen、Zhirong Liu、Xiao Xu、Xu Chen、Zhenhua Dong 和 Xiuqiang He。IPGAN：通过对抗采样生成信息丰富的物品对。IEEE Transactions on Neural Networks and Learning S

$y$词干，2020。

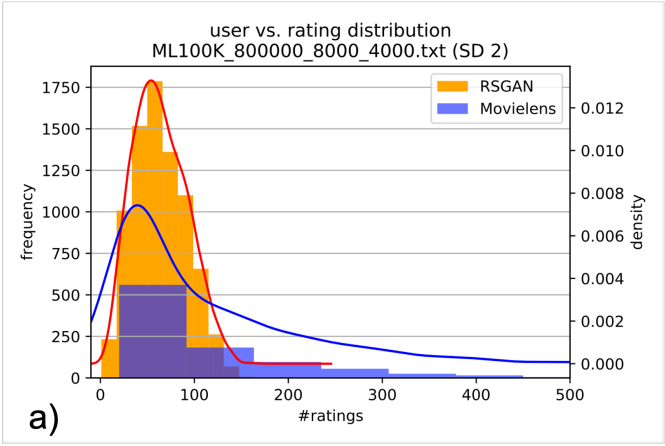
[43] Jianli Zhao、Hao Li、Lijun Qu、Qinzhi Zhang、Qiuxia Sun、Huan Huo 和 Maoguo Gong。Dcfgan：一种采用改进负采样的基于会话的推荐系统对抗性深度强化学习框架。*InformationSciences*，596:222–235，2022。[44] Jinyang Sun、Baisong Liu、Hao Ren 和 Weiming Huang。Ncgan:: 一种用于推荐系统的神经对抗协同过滤。*Journal of Intelligent & Fuzzy Systems*，42(4):2915–2923，2022。[45] Yuan Lin、Zhang Xie、Bo Xu、Kan Xu 和 Hongfei Lin。面向推荐系统的信息流增强生成对抗网络。载于第*44*届国际*ACM*信息检索研究与发展会议论文集，第1703–1707页，2021。[46] Qingxian Wang、Qing Huang、Kangkang Ma 和 Xuerui Zhang。基于模型正则化Wasserstein生成对抗网络的推荐系统。载于*2021*年*IEEE*国际系统、人类与控制论会议（*SMC*），第2043–2048页。IEEE，2021。[47] Jing Wen、Xi-Ran Zhu、Chang-Dong Wang 和 Zhihong Tian。采用条件生成对抗网络实现个性化推荐的框架。*KnowledgeandInformationSystems*，64(10):2637–2660，2022。[48] Grace Deng、Cuize Han 和 David S Matteson。利用生成对抗网络扩展面向排序应用的缺失数据填补。*Data Mining and Knowledge Discovery*，第1–23页，2022。[49] Honglong Chen、Shuai Wang、Nan Jiang、Zhe Li、Na Yan 和 Leyi Shi。结合循环神经网络的信任感知生成对抗网络推荐系统。*InternationalJournalof IntelligentSystems*，36(2):778–795，2021。[50] Wafa Shafqat 和 Yung-Cheol Byun。解决推荐系统数据不平衡问题的混合式生成对抗网络方法。*IEEE Access*，10:11036–11047，2022。[51] Zinan Lin、Ashish Khetan、Giulia Fanti 和 Sewoong Oh。Pacgan：生成对抗网络中两个样本的威力。载于第*32*届国际神经信息处理系统会议论文集，第1505–1514页，2018。[52] Martin Mladenov、Chih-wei Hsu、Vihan Jain、Eugene Ie、Christopher Colby、Nicolas Mayoraz、Hubert Pham、Dustin Tran、Ivan Vendrov 和 Craig Boutilier。通过RecSim NG展示推荐系统生态中的原理性不确定性建模。载于第十四届*ACM*推荐系统会议，第591–593页，2020。[53] Jing-Cheng Shi、Yang Yu、Qing Da、Shi-Yong Chen 和 An-Xiang Zeng。Virtual-Taobao：为强化学习构建真实世界在线零售环境的虚拟化平台。载于*AAAI*人工智能会议论文集，第33-01卷，第4902–4909页，2019。[54] Cai-Nicolas Ziegler、Sean M McNee、Joseph A Konstan 和 Georg Lausen。通过主题多样化改进推荐列表。载于第*14*届国际万维网会议论文集，第22–32页，2005。[55] María del Carmen Rodríguez-Hernández、Sergio Ilarri、Ramón Hermoso 和 Raquel Trillo-Lado。DataGenCars：用于评估情境感知推荐系统的合成数据生成器。*Pervasiveand MobileComputing*，38:516–541，2017。[56] Vladimir Provalov、Elizaveta Stavinova 和 Petr Chunaev。SynEvaRec：用于在合成数据类别上评估推荐系统的框架。载于*2021*年国际数据挖掘研讨会会议（*ICDMW*），第55–64页。IEEE，2021。[57] Mohiuddin Ahmed、Raihan Seraj 和 Syed Mohammed Shamsul Islam。k均值算法：全面综述与性能评估。*Electronics*，9(8):1295，2020。[58] Fernando Ortega、Bo Zhu、Jesús Bobadilla 和 Antonio Hernando。Cf4j：面向Java的协同过滤。*Knowledge-BasedSystems*，152:94–99，2018。

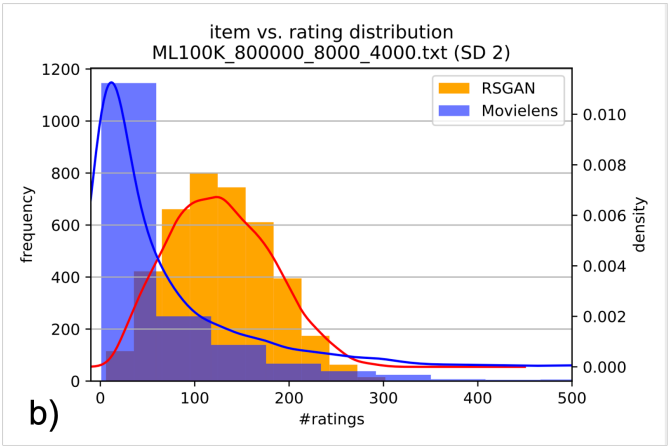
附录 A

| DeepMF | 值 |
| --- | --- |
| 嵌入维度（用户和物品均适用） | 5 |
| 优化器 | Adam |
| 损失函数 | 均方误差 |
| 训练轮数 | 20 |
| **GAN生成器** | |
| 输入形状，噪声向量维度 | 100 |
| 块1全连接层 #神经元 | 10 |
| 块1激活函数 | LeakyReLU，阿尔法 0.2 |
| 第1块归一化 | 批归一化 ，动量 0.8 |
| 第2块全连接层 #神经元 | 20 |
| 第2块激活函数 | LeakyReLU，阿尔法 0.2 |
| 第2块正则化 | 随机失活 0.2 |
| 第3块全连接层 #神经元 | $2 * embedding\ size + 1$ |
| 第3块激活函数 | 线性 |
| **GAN判别器** | |
| 输入：形状 | $2 * embedding\ size + 1$ |
| 块1全连接层 #神经元 | 6 |
| 块1激活函数 | LeakyReLU，阿尔法 0.2 |
| 第2块全连接层 | 1 |
| 第2块激活函数 | Sigmoid |
| **生成对抗网络训练** | |
| 训练轮数 | 20 |
| 批次大小 | 64 |
| 随机噪声 | 高斯分布 (0,1) |
| 损失函数 | $(real\ samples\ loss + fake\ samples\ loss \quad 2$ |

)/表4：主要参 d.
数以及 RSGAN 方法中所涉及神经模型的超参数值

附录B



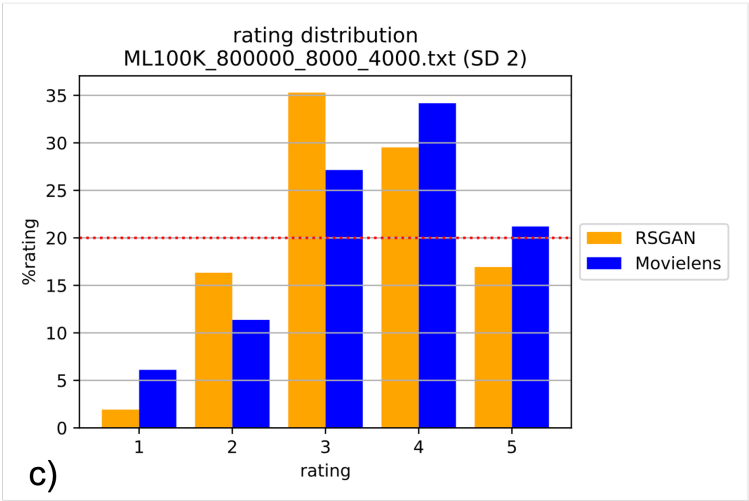




图16：与源数据集中的数据分布相比，由 Movielens 100K 生成的合成数据集中的主要数据分布。用户数量：8000，物品数量：4000，初始样本数量：800000，高斯噪声的标准差：2.5。图 a) 显示虚假用户的分布（y 轴）与每个用户所对应的评分数量（x 轴）的关系。图 b) 显示虚假物品的分布（y 轴）与每个物品所对应的评分数量（x 轴）的关系。图 c) 显示数据集中各个可用投票值 1、2、3、4、5（x 轴）对应的评分百分比（y 轴）。