arXiv:2303.07001v1 [cs.IR] 13 Mar 2023

基于概率语义聚合的神经群体推荐

Jorge Due˜nas-Ler´ın[1,3], Ra´ul Lara-Cabrera[2,3], Fernando Ortega[2,3],和 Jes´ús Bobadilla[2,3]

[1]马德里理工大学，马德里，西班牙[2]马德里理工大学计算机系统系，马德里，西班牙[3]马德里理工大学 KNODIS 研究组，马德里，西班牙

摘要

面向用户群组的推荐是推荐系统中一个颇具挑战性的子领域。其核心问题在于，如何以及在何处将每组用户信息聚合为一个独立实体，例如排序推荐列表、虚拟用户或用于编码的多热输入向量。本文提出一种创新策略，将聚合应用于输入神经网络模型的多热向量。该聚合赋予向量概率语义，由此得到的输入向量馈入一个模型，使其能够根据个体预测便捷地泛化出群组推荐。此外，借助所提出的架构，只需将个体评分输入预训练模型进行前馈计算，即可获得群体推荐；也就是说，无需获取包含用户群组信息的数据集，也无需分别进行个体和群组两次训练。这种方法还避免了为支持个体学习和群组学习而维护两个不同模型。我们使用三个具有代表性的协同过滤数据集和一系列基线方法对所提出的架构进行了实验；结果表明，与当前最先进水平相比，该架构取得了令人满意的准确率提升。

1 引言

个性化是人工智能（AI）领域之一，对个人生活的影响尤为深远。我们可以找到大量服务，为我们提供个性化的新闻、视频、歌曲、餐厅、服装、旅行等选择。最具影响力的科技公司广泛使用个性化服务：亚马逊、网飞、Spotify、TripAdvisor、谷歌、TikTok 等。这些公司会生成

个性化推荐，并借助推荐系统（RS） [4] 应用为用户提供服务。推荐系统通过筛选与用户消费物品的日志、消费时间和地点，以及用户、其社交网络和物品内容（文本、图片、视频等）相关的最重要信息，为用户提供个性化产品或服务（物品）。根据过滤策略，我们可以将推荐系统分为人口统计型 [5], 基于内容型 [10],上下文感知型 [20], 社交型 [25], 协同过滤（CF） [6, 9] 以及集成过滤 [12, 8]。目前，矩阵分解（MF） [24]机器学习模型被用于根据输入数据（投票）生成准确、快速的推荐。MF 将稀疏且庞大的离散投票矩阵（从用户到物品）转换为两个稠密且相对较小的实数值矩阵。其中一个矩阵包含一组短小而稠密的向量，用来表示用户；第二个矩阵中的向量则表示物品。每个向量元素（实数值）称为“$hiddenfactor\ value$”，因为它表示输入数据（投票）之间某种复杂且未知的关系。

尽管矩阵分解机器学习模型速度快且准确，但也存在一个显著缺点：它们无法从隐藏因子中检测出输入数据之间复杂的非线性关系。神经网络（NN）可以通过其非线性激活函数解决这一问题。基于神经网络的推荐系统 [7, 5] 通过在嵌入和隐藏层中编码评分矩阵的模式来压缩信息 [18]。这些嵌入发挥着矩阵分解隐藏因子的作用，并通过纳入非线性关系丰富结果。最广为人知的基于神经网络的推荐系统方法是广义矩阵分解（GMF）和多层感知机（MLP） [16]。

群体推荐（GR） [21, 9] 是推荐系统领域的一个子领域，旨在向用户群体而非个人用户提供推荐（例如：向三位好友组成的群组推荐一部电影）。与常规推荐系统一样，目标是向群组提供准确的推荐。在这种情况下，可以采用多种策略；其中最常见的是：a）最小化平均准确性误差：推荐平均而言最受所有群组成员喜爱的物品；b）最小化最大准确性误差：推荐不会令任何一位群组成员过分反感的物品。需要指出的是，目前没有包含群体信息、可供群体推荐模型使用的开放数据集；因此，研究模型通常使用随机生成的群组进行训练和测试。

无论采用何种机器学习方法来实现群组推荐系统，最值得关注的设计概念都是确定聚合阶段的位置，以便将个体信息转化为群体信息。一般规律是：聚合阶段越早，群组推荐系统的性能越好。群体信息可以在三个不同的位置聚合为统一群组实体：a) 模型之前，b) 模型中，c) 模型之后。最直观的方法是将个体推荐合并为统一的群组推荐（选项 c）。这种方法称为个体偏好聚合（IPA），需要先处理多个个体推荐，再进行排名聚合。然而，这一过程

速度较慢，准确性也不高。另一方面，为了在推荐中考虑整个群组，我们应在模型之前或模型内部进行处理（选项 a 或 b）。这些方法称为群体偏好聚合（GPA）。在模型之前聚合群体信息，需要在更高维空间中处理用户-项目交互矩阵，这可能导致错误信息问题。要在模型中聚合群体信息，我们需要在低维空间中处理用户的隐向量。

将多个个人用户的隐向量聚合为一个统一的虚拟用户隐向量 [22]，可避免多次计算模型预测，也使排名聚合阶段变得不再必要。此外，这种方法利用了信息经矩阵分解压缩后得到的精简信息：只需对群组用户具有代表性的短而稠密的向量取平均值，即可得到虚拟用户；这种方法既高效又准确。一个有趣的问题是：神经网络能否像矩阵分解那样运作，以获得虚拟用户并生成推荐？首先要注意，许多基于神经网络的推荐系统模型会将用户嵌入和项目嵌入压缩到不同的潜在空间中，这可能会造成问题；其次，神经网络的非线性集成表示比矩阵分解的隐因子表示更复杂；因此，简单地对群组中用户的集成表示取平均值，并不能自动确保得到具有代表性的虚拟用户嵌入。此外，基于模型的聚合（上一段中的选项“b”）依赖于具体模型，因此需要针对不同的基于神经网络的推荐系统模型设计并测试不同的解决方案。虽然神经网络的潜在空间是当前捕捉用户与物品关系的最先进方法，但也有研究设计了其他机器学习方法，例如使用带重启的随机游走方法 [11]，提供一个关联用户、物品和群组的框架，并利用物品内容和用户档案。文献提出了一种三阶段方法 [1]，依次处理二元矩阵分解、图和动态共识模型，以提高群体推荐的精确性和公平性。一些当前相关的群体推荐研究旨在利用成员偏好（影响力或专业知识）的概念，该概念基于相似性和信任。其核心思想是识别群体领导者，即比其他成员更受信任、影响力也更大的群组成员。文献 [3]，将模糊聚类与隐式信任度量相结合，以寻找邻域。基于用户偏好差异的平均策略 [26] 已与可信社交网络相结合，用于修正推荐。群体推荐的一种聚合方法借鉴众包概念来估算群组成员的专业知识水平 [19]；该方法通过灵敏度和特异度参数实现。文献 [14]，使用专业知识因子、人格影响、偏好相似性和人际关系，评估社会因素对群体推荐计算模型的影响。

本文介绍了如何通过使用原始协同过滤数据（即个人用户对物品的评分，不含任何额外信息）训练模型，利用基于神经网络的推荐系统生成群组推荐。我们使用基于神经网络的推荐系统最流行的两种实现——广义矩阵分解和多层感知机——对群组推荐进行了测试。如前所述，要向

群组提供基于神经网络的推荐系统推荐，必须聚合群组中个人用户的信息。我们选择的信息聚合设计，是将群组中的用户合并到输入向量中，再将该向量输入神经网络的用户嵌入层。这种聚合设计并非新颖，因为 [23] 曾将其应用于多层感知机架构。不过，与当前最先进水平相比，我们的方法结合了若干创新之处。一方面，群组中用户的聚合采用概率函数，而非简单的多热编码 [23]；这能更好地体现输入向量中各用户的相对重要性。此外，这种聚合方法可作为任何神经网络群组推荐模型的前端。另一方面，我们提议使用简单的推荐系统神经网络模型（广义矩阵分解），而非更深层的多层感知机模型 [23]；我们的假设是，复杂模型会对群组推荐场景过拟合，因为这类模型旨在准确预测个人用户的评分，而群组推荐必须满足群组用户品味的平均值。也就是说，群组推荐应着眼于泛化群组中的个体品味。此外，所提出的架构只需训练一次，即可提供个体推荐和群组推荐；具体而言，模型仅使用个体推荐进行训练（与常规推荐系统相同）。模型经过训练、能够返回个体预测后，我们便可通过聚合群组中的所有用户来填充输入向量，再将其输入训练好的模型进行前向传播，最终获得面向该用户群组的推荐。无论如何，这些创新之处的影响可在第3节中评估，我们将在该节通过实证比较所提出的聚合设计与主要基线方法 [23]

总之，群体推荐领域当前最先进的方法存在以下缺点：a）部分研究依赖协同过滤评分之外的额外数据，例如信任或声誉信息，而大多数数据集并不包含这些信息；b）不同方法选择在模型之前（IPA）或之后（排序）聚合个人用户，因此无法利用机器学习模型的内部表示（GPA）；c）所提出的群体推荐神经网络模型解决方案往往采用为个体推荐而设计的架构，而非为群组推荐设计的架构，这会导致模型过拟合，并且随着群组用户数量增加，可扩展性较差。为弥补这一不足，我们提出的模型：a）仅使用协同过滤评分；b）在模型内部进行用户聚合；c）其模型深度和设计能够实现适当的学习泛化。此外，我们通过实验，针对不同的聚合策略测试所提出的模型，以确定学习阶段使用的群体标签。另一方面，我们的架构和实验有一个明显的局限：缺乏在一些特别具有挑战性的场景下进行测试，例如群组用户冷启动、极度稀疏的数据集、热门物品偏差的影响，以及恐惧群体推荐。

本文其余部分结构如下：第2节介绍所测试的模型和聚合函数；第3节说明实验设计、所选的质量度量和采用的数据集，并展示所得结果；第4节对结果进行解释；第5节则概述文章的主要结论及建议的未来研究方向。

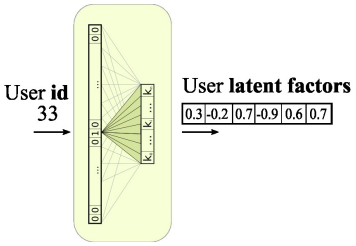

图 1：嵌入层示意图。

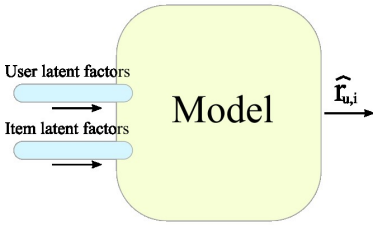

图 2：基于协同过滤的神经网络模型。

2 提出的模型

在协同过滤中，用户与物品之间的交互（购买、浏览、评分等）存储在稀疏矩阵中，因为用户通常只与可用物品中的一小部分进行交互；同样，现有用户中也只有一小部分会与这些物品进行交互。如表2所示，该矩阵的稀疏度约为95%至98%。为了处理这种稀疏性，当前基于神经网络 [16] 的协同过滤模型会使用嵌入层，将用户和物品映射到低维潜在空间中。嵌入层是神经网络中非常常见的一种层，它以任意实体为输入，并返回一个向量，用低维形式表示该实体在潜在空间中的表征。这些向量通常称为 $latent\ factors$。为了将实体转换为低维表示，嵌入层首先将实体转换为独热编码表示（通常使用哈希函数）。图1概述了这一过程。

在协同过滤的背景下，需要两个嵌入层：一个用于用户，另一个用于物品。随后，两个嵌入层通过神经网络架构进行组合（见图 2）。例如，上述模型广义矩阵分解和多层感知机分别采用点积层和连接层，并在其后接入若干全连接稠密层作为架构。

形式化地，我们定义一个神经网络模型 Φ，用于预测用户 u 对物品 i（$\hat{r}_{u,i}$）的评分，该模型结合了由嵌入层 (Emb_L) 提供的用户 u ($\vec{l_u}$) 和物品 i ($\vec{l_i}$) 的潜在因子：

$$\begin{aligned} Emb_L(u) &= \vec{l_u} \\ Emb_L(i) &= \vec{l_i}, \\ \Phi(\vec{l_u}, \vec{l_i}) &= \hat{r}_{u,i} \end{aligned} \tag{1}$$

如第 1 节所述，在群体推荐中，一种直接的策略是 IPA [17]。该策略会对群组中的每位成员进行预测，然后再进行聚合。此策略并未将群组视为一个整体。假设有一个用户群组 $G = \{u_1, u_2, ..., u_n\}$，则该群组 G 对某个项目 i （$\hat{r}_{G,\ i}$）的评分预测，是通过对个体预测取平均值计算得出的：

$$\hat{r}_{G,i} = \frac{1}{\|G\|} \sum_{u \in G} \hat{r}_{u,i} = \frac{1}{\|G\|} \sum_{u \in G} \Phi(\vec{l_u}, \vec{l_i}) \tag{2}$$

另一方面，GPA 策略会将群组视为一个整体。需要注意的是，群组内用户的顺序和群组长度都不应影响聚合；因此，聚合应满足以下约束：置换不变和固定结果长度 [23]。采用 GPA 策略的目标是，通过一次前向传播获得预测 $\hat{r}_{G,\ i}$，并将群组视为一个整体实体。我们可以通过聚合群组中每位用户的潜在因子，获得群组的潜在因子 $\vec{l}_G$，从而实现这一目标。聚合群组的潜在因子后，即可使用模型 Φ 计算预测：

$$\begin{aligned} Emb_L(G) &= \vec{l}_G \\ \hat{r}_{Gi} &= \Phi(\vec{l}_G, \vec{l}_i) \end{aligned} \tag{3}$$

通过修改神经网络的输入，可以在嵌入层中聚合群组的潜在因子。如前所述，嵌入层的输入是实体的独热表示。这种方法适用于进行个体预测；然而，对于群组推荐，我们需要对用户嵌入层应用多热表示，即通过将用户嵌入层的多个输入（与群组成员相关的输入）设为大于 0 的值来编码群组。这种编码使我们能够同时考虑群组中的所有用户，以提取群组的潜在因子 $\vec{l}_G$。

最简单的聚合方式（DeepGroup 模型 [23],所采用的方法）是将一个与群组规模成正比的常数值作为嵌入层的输入。我们将用户 u 的用户嵌入层输入定义为

$$EmbeddingInput_{Average}(u) = \begin{cases} \frac{1}{\|G\|} & if u \in G \\ 0 & if u \notin G \end{cases} \tag{4}$$

我们将这种聚合称为“$Average$”，因为嵌入层会生成一个群组潜在因子，其值等于群组中所有用户潜在因子的平均值。

推荐系统掌握的用户信息越多，预测效果就越好。因此，为了利用这一事实，我们测试了“$Expertise$”聚合方法，根据用户在系统中提交的投票数量为其赋予相应权重。设 $\|R_u\|$为用户 u的评分数量，则用户 u 嵌入层的输入定义为

$$EmbeddingInput_{Expertise}(u) = \begin{cases} \frac{\|R_u\|}{\sum_{g \in G} \|R_g\|} & if u \in G \\ 0 & if u \notin G \end{cases} \tag{5}$$

除了“$Expertise$”聚合外，我们还提出了“$Soft\text{-}\max$”聚合作为“$Expertise$”聚合的平滑版本。在

在这种情况下，用户 u 的用户嵌入层输入定义为

$$EmbeddingInput_{Softmax}(u) = \begin{cases} \frac{e^{\|Ru\|}}{\sum_{g \in G} e^{\|Rg\|}} & if u \in G \\ 0 & if u \notin G \end{cases} \quad (6)$$

在图3中，我们可以看到这些公式在群体推荐流程中的对应位置。第一步是通过所述的某种聚合方法生成多热向量（公式（4）至（6））。该向量（群组的多热表示）被输入嵌入层，以获得群组的潜在因子向量 $\vec{l}_G$ （公式（3））。获得群组和物品的潜在因子后，将它们输入模型 Φ （广义矩阵分解或多层感知机），以预测群组对物品 i 的评分G （公式（2））。

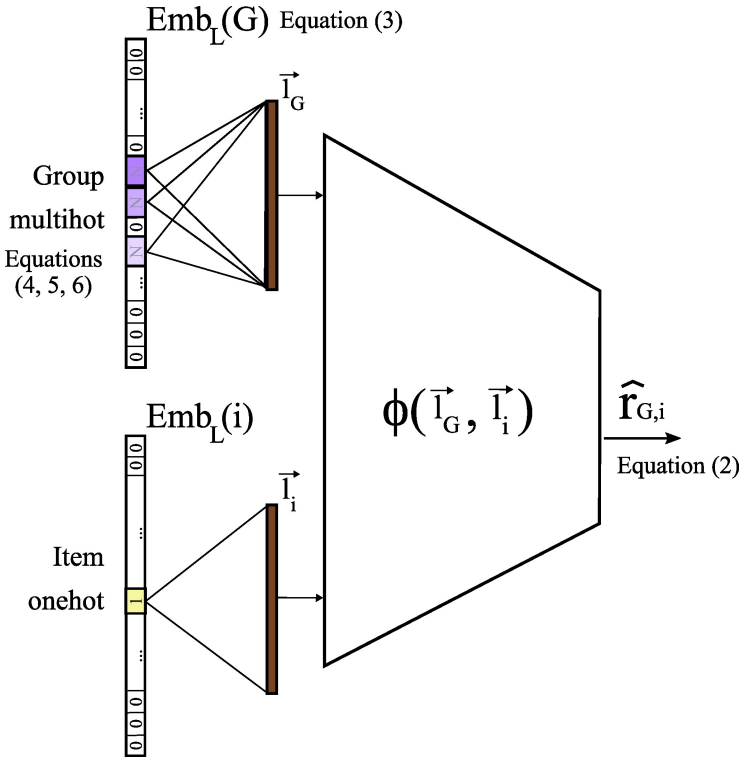

图3：所提出模型的图形表示。

在表1中，我们可以看到一个示例，其中包含若干用户（13、24、30和42），以及他们不同的评分数量（表1a）、以多热形式输入用户嵌入层的数值（表1b）、各自的潜在因子（表1c），以及采用不同聚合方式得到的最终群组潜在因子（表1d）。

User	...	u_{13}	...	u_{24}	...	u_{30}	...	u_{42}	...
#评分		2		5		6		3	

（a）评分数量。

策略\用户	...	u_{13}	...	u_{24}	...	u_{30}	...	u_{42}	...
平均值		0,25		0,25		0,25		0,25	
专业知识		0,13		0,31		0,38		0,19	
Softmax		0,01		0,26		0,70		0,03	

（b）用户嵌入层的输入值。

用户\因子	l_1	l_2	l_3
u_{13}	0,1	0,6	0,3
u_{24}	0,7	0,2	0,9
u_{30}	0,8	0,4	0,1
u_{42}	0,5	0,7	0,8

（c）假设潜在空间大小为 3 时用户的潜在因子。

聚合\因子	l_{G_1}	l_{G_2}	l_{G_3}
平均值	0,525	0,475	0,525
专业知识	0,629	0,425	0,508
Softmax	0,758	0,359	0,331

(d) 使用不同聚合方法的群组潜在因子。

表1：完整的聚合示例。

3 实验评估

本节将介绍为验证本文提出的聚合方法而开展的实验。如前所述，这些实验采用了最流行的基于神经网络的推荐系统架构：广义矩阵分解和多层感知机。我们选择这两种架构，是因为它们最为知名，并且能够为个体预测提供最佳结果。然而，所提出的聚合策略可以应用于任何基于嵌入层的神经网络架构。

选择数据集时考虑了以下因素：a）目前没有包含群体投票信息的开放数据集；b）广义矩阵分解和多层感知机模型应使用个体投票进行训练，因为所提出的聚合方法可以基于已训练的模型计算群组预测。基于这些原因，我们选择了推荐系统领域中以下具有代表性的数据集：MovieLens1M [15], , 推荐系统研究中最受欢迎的数据集；FilmTrust [13], ，一个规模小于MovieLens1M的数据集，可用于衡量聚合方法在用户、物品和评分数量较少的数据集上的表现；以及MyAnimeList，一个投票范围大于MovieLens1M的数据集。由于训练和测试模型所需的计算时间较长，因此未选择Netflix Prize或MovieLens10M等其他热门数据集。所选数据集的主要参数见表2。

数据集	用户数	项目数	评分数	分数	稀疏度
MovieLens1M	6,040	3,706	911,031	1 到 5	95.94
FilmTrust	1,508	2,071	35,497	0 到 5	87.98
MyAnimeList	19,179	2,692	548,967	1 到 10	98.94

表2：实验中使用的数据集的主要参数。

合成群组的生成方式确保了所有群组在测试集中至少对5件物品进行过投票。这样，便可以评估预测质量以及面向群组的推荐质量，具体如下文所述。生成的群组大小各不相同（从2名到10名用户）。每种群组大小均生成了10000个合成群组。群组按照以下算法生成：

1. 定义群组大小 S。2. 随机选择至少有 S 用户在测试集中评分过的 5 个物品。

3. 找出所有给第 2 步中选出的 5 个物品评分过的用户。

4. 如果找到的随机选择 S 个用户则返回第 2 步群组。

为了衡量群组预测的质量，我们计算了平均绝对误差

$$MAE = \frac{1}{\#groups} \sum_{G} \frac{1}{\|G\| \cdot \|I_G\|} \sum_{u \in G} \sum_{i \in I_G} \mid \hat{r}_{G,i} - r_{u,i} \mid, \tag{7}$$

均方误差

$$MSE = \frac{1}{\#groups} \sum_{G} \frac{1}{\|G\|} \frac{1}{\|G\| \cdot \|I_G\|} \sum_{u \in G} \sum_{i \in I_G} \left(\hat{r}_{G,i} - r_{u,i}\right)^2, \tag{8}$$

以及平均最大群组误差

$$MAX = \frac{1}{\#groups} \sum_{G} \max_{u \in G} \max_{i \in I_G} \mid \hat{r}_{G,i} - r_{u,i} \mid, \tag{9}$$

其中， I_G 表示群组评分的物品 G

为了衡量面向群组的推荐质量，我们计算了归一化折损累计增益（NDCG）得分

$$NDCG@N = \frac{1}{\#groups} \sum_{G} \frac{DCG_G@N}{IDCG_G@N}, \tag{10}$$

$$DCG_G@N = \sum_{i \in X_G^N} \frac{\bar{r}_{G,i}}{log_2(pos_G(i) + 1)}, \tag{11}$$

$$IDCG_G@N = \sum_{i \in T_G^N} \frac{\bar{r}_{G,i}}{log_2(ipos_G(i) + 1)}, \tag{12}$$

其中， N 是推荐给群组的物品数量（在我们的实验中， $N = 5$ 根据合成群组的生成方式确定）， X_G^N是推荐给群组 G的 N 个物品的集合， $pos_G(i)$是物品 i 在该群组 G 推荐列表中的位置， T_G^N 是该群组 G的前 N 个物品的集合， $ipos_G(i)$是物品 i 对群组 G而言的理想排名， $\bar{r}_{G,\ i}$ 是属于群组 G 的用户对物品 i的平均评分。

这些分数对应的实验结果见表 3（平均绝对误差）、表 4（均方误差）、表 5（最大值）和表 6（归一化折损累计增益）。最佳结果所在的单元格已高亮显示，每项指标的标准差则列于括号中。所有结果均在第 4 节中分析。

所有实验均使用配备 48 GB GDDR6 显存、4,608 个 NVIDIA 张量核心、性能达 16.3 万亿次浮点运算每秒的 NVIDIA Quadro RTX 8000 图形处理器运行。我们致力于推动可复现科学，因此，所有实验的源代码、所用参数值及随机种子均已发布在 GitHub[1] 上。

[1] https://github.com/KNODIS-Research-Group/neural-cf-for-groups

表 3：平均绝对误差

模型 \组大小	2	3	4	5	6	7	8	9	10
GMF IPA GMF 平均值	0.74205 (0.409)	0.76075 (0.341)	0.76893 (0.299)	0.77009 (0.271)	0.77745 (0.249)	0.77659 (0.234)	0.77681 (0.221)	0.77599 (0.212)	0.77558 (0.201)
GMF 专业知识	0.74393 (0.41)	0.76207 (0.341)	0.77018 (0.299)	0.77155 (0.27)	0.779 (0.249)	0.77782 (0.234)	0.77834 (0.221)	0.77729 (0.211)	0.77685 (0.201)
GMF Softmax	0.74246 (0.409)	0.7608 (0.341)	0.76891 (0.299)	0.77012 (0.27)	0.77751 (0.249)	0.7766 (0.234)	0.77687 (0.221)	0.77602 (0.212)	0.77562 (0.201)
多层感知机 IPA	0.74956 (0.444)	0.77342 (0.361)	0.78055 (0.313)	0.78211 (0.28)	0.78853 (0.258)	0.78633 (0.241)	0.78678 (0.228)	0.78591 (0.219)	0.78509 (0.207)
MLP 平均值 DeepGroup	**0.7236** **(0.486)**	**0.74289** **(0.404)**	**0.74916** **(0.355)**	**0.75018** **(0.32)**	**0.75656** **(0.295)**	0.75537 (0.275)	**0.75551** **(0.261)**	**0.75388** **(0.25)**	**0.75355** **(0.238)**
MLP 专业知识	0.72596 (0.486)	0.74432 (0.405)	0.75031 (0.355)	0.75132 (0.32)	0.75787 (0.295)	0.75607 (0.276)	0.75709 (0.261)	0.75481 (0.25)	0.755 (0.238)
MLP Softmax	0.72407 (0.485)	0.74297 (0.404)	0.74925 (0.355)	0.75019 (0.32)	0.75669 (0.295)	**0.75531** **(0.275)**	0.7556 (0.261)	0.75389 (0.25)	0.75361 (0.238)

(a) MovieLens1M

模型 \组大小	2	3	4	5	6	7	8	9	10
GMF IPA GMF 平均值	0.61552 (0.451)	0.71033 (0.32)	0.73011 (0.28)	0.73419 (0.252)	0.73606 (0.232)	0.73832 (0.217)	0.74045 (0.203)	0.74298 (0.193)	0.74212 (0.185)
GMF 专业知识	0.6149 (0.444)	0.71239 (0.319)	0.73144 (0.281)	0.73583 (0.252)	0.73742 (0.232)	0.7396 (0.217)	0.74166 (0.203)	0.74418 (0.193)	0.74336 (0.184)
GMF Softmax	0.61512 (0.448)	0.71088 (0.319)	0.73035 (0.28)	0.73445 (0.252)	0.73624 (0.232)	0.73847 (0.217)	0.74057 (0.203)	0.74309 (0.193)	0.74222 (0.185)
多层感知机 IPA	0.58165 (0.442)	0.70368 (0.325)	0.72062 (0.281)	0.7252 (0.252)	0.72788 (0.232)	0.73073 (0.217)	0.73353 (0.203)	0.73623 (0.193)	0.73525 (0.185)
MLP 平均值 DeepGroup	0.58199 (0.447)	**0.70129** **(0.319)**	**0.71693** **(0.275)**	**0.7212** **(0.247)**	**0.72421** **(0.228)**	**0.727** **(0.214)**	**0.72976** **(0.2)**	**0.7328** **(0.191)**	**0.73188** **(0.183)**
MLP 专业知识	**0.57903** **(0.453)**	0.70449 (0.321)	0.71891 (0.276)	0.72315 (0.247)	0.72581 (0.228)	0.72865 (0.213)	0.73127 (0.2)	0.73429 (0.191)	0.73336 (0.183)
MLP Softmax	0.58063 (0.45)	0.70226 (0.319)	0.71734 (0.275)	0.72153 (0.247)	0.72443 (0.228)	0.7272 (0.214)	0.72993 (0.2)	0.73295 (0.191)	0.73202 (0.183)

(b) FilmTrust

模型 \组大小	2	3	4	5	6	7	8	9	10
GMF IPA GMF 平均值	**0.93068** **(0.567)**	0.95819 (0.477)	0.97595 (0.417)	0.99149 (0.38)	1.0017 (0.346)	1.01404 (0.329)	1.01942 (0.312)	1.022 (0.297)	1.02445 (0.282)
GMF 专业知识	0.93675 (0.572)	0.96424 (0.48)	0.98092 (0.419)	0.99603 (0.382)	1.00649 (0.349)	1.01805 (0.331)	1.02352 (0.314)	1.0258 (0.3)	1.0284 (0.284)
GMF Softmax	0.93219 (0.568)	0.95848 (0.477)	0.97559 (0.417)	0.99104 (0.38)	1.00133 (0.346)	1.01363 (0.329)	1.0191 (0.312)	1.02173 (0.297)	1.02422 (0.282)
多层感知机 IPA	0.95479 (0.585)	0.98155 (0.484)	0.99709 (0.42)	1.00977 (0.381)	1.01745 (0.347)	1.02794 (0.329)	1.03188 (0.311)	1.03335 (0.298)	1.03451 (0.282)
MLP 平均值 DeepGroup	0.93161 (0.618)	0.95609 (0.515)	**0.96961** **(0.45)**	0.98456 (0.408)	0.99331 (0.371)	1.0052 (0.351)	1.00857 (0.332)	1.01039 (0.317)	1.01233 (0.3)
MLP 专业知识	0.93803 (0.622)	0.96132 (0.517)	0.97587 (0.453)	0.98923 (0.409)	0.99851 (0.373)	1.00879 (0.353)	1.01258 (0.334)	1.01493 (0.319)	1.01599 (0.302)
MLP Softmax	0.93351 (0.619)	**0.95594** **(0.515)**	0.97007 (0.451)	**0.9843** **(0.408)**	**0.99329** **(0.371)**	**1.00486** **(0.351)**	**1.00844** **(0.332)**	**1.0101** **(0.317)**	**1.01211** **(0.3)**

（c）MyAnimeList

表 4：均方误差

模型 \组大小	2	3	4	5	6	7	8	9	10
GMF IPA GMF 平均值	**0.87504** **(0.914)**	**0.90947** **(0.767)**	0.92437 (0.676)	0.92648 (0.609)	0.94191 (0.567)	0.93896 (0.53)	0.9376 (0.5)	0.93832 (0.48)	0.93571 (0.454)
GMF 专业知识	0.88044 (0.918)	0.91329 (0.77)	0.92754 (0.678)	0.92979 (0.61)	0.94504 (0.568)	0.94135 (0.531)	0.93994 (0.501)	0.94037 (0.481)	0.93737 (0.454)
GMF Softmax	0.87614 (0.914)	0.90952 (0.767)	**0.92424** **(0.676)**	**0.92645** **(0.609)**	**0.94188** **(0.567)**	**0.93885** **(0.53)**	**0.93752** **(0.5)**	**0.93823** **(0.48)**	**0.93561** **(0.454)**
多层感知机 IPA	0.94301 (0.982)	0.96557 (0.814)	0.96809 (0.712)	0.96499 (0.635)	0.97727 (0.591)	0.96889 (0.55)	0.96634 (0.518)	0.96648 (0.498)	0.96187 (0.47)
MLP 平均值 DeepGroup	0.99415 (1.037)	1.03182 (0.875)	1.04278 (0.779)	1.04242 (0.695)	1.05597 (0.651)	1.05056 (0.605)	1.04927 (0.574)	1.04836 (0.552)	1.04669 (0.524)
MLP 专业知识	0.9986 (1.038)	1.03522 (0.88)	1.04511 (0.78)	1.04435 (0.696)	1.05806 (0.651)	1.05121 (0.604)	1.05084 (0.575)	1.0491 (0.553)	1.04787 (0.523)
MLP Softmax	0.99487 (1.035)	1.03145 (0.874)	1.04258 (0.779)	1.04235 (0.694)	1.05605 (0.651)	1.05034 (0.604)	1.04925 (0.574)	1.04822 (0.552)	1.04659 (0.524)

(a) MovieLens1M

模型 \组大小	2	3	4	5	6	7	8	9	10
GMF IPA GMF 平均值	0.67941 (1.077)	0.7889 (0.688)	0.82014 (0.608)	0.82302 (0.544)	0.82574 (0.502)	0.83038 (0.469)	0.8324 (0.44)	0.8375 (0.42)	0.83544 (0.401)
GMF 专业知识	0.67314 (1.05)	0.79105 (0.686)	0.82229 (0.608)	0.82546 (0.543)	0.82758 (0.501)	0.83205 (0.468)	0.83382 (0.439)	0.83875 (0.418)	0.83673 (0.399)
GMF Softmax	0.67596 (1.063)	0.7892 (0.687)	0.82043 (0.608)	0.82333 (0.544)	0.82592 (0.502)	0.83053 (0.469)	0.83251 (0.44)	0.83758 (0.42)	0.83552 (0.401)
多层感知机 IPA	**0.61169** **(1.002)**	0.77197 (0.683)	0.79514 (0.592)	0.7988 (0.528)	0.80315 (0.487)	0.80807 (0.454)	0.81084 (0.427)	0.81611 (0.406)	0.81408 (0.388)
MLP 平均值 DeepGroup	0.61859 (1.009)	**0.7636** **(0.674)**	**0.78665** **(0.585)**	**0.79085** **(0.523)**	**0.79606** **(0.484)**	**0.80126** **(0.452)**	**0.80444** **(0.425)**	**0.81023** **(0.405)**	**0.80841** **(0.387)**
MLP 专业知识	0.62484 (0.979)	0.7691 (0.68)	0.78956 (0.586)	0.79332 (0.522)	0.79799 (0.483)	0.80305 (0.45)	0.80595 (0.423)	0.81166 (0.403)	0.80977 (0.385)
MLP Softmax	0.62106 (0.992)	0.7649 (0.675)	0.78709 (0.585)	0.79116 (0.523)	0.79625 (0.484)	0.80142 (0.452)	0.80456 (0.424)	0.81033 (0.404)	0.8085 (0.387)

(b) FilmTrust

模型 \组大小	2	3	4	5	6	7	8	9	10
GMF IPA GMF 平均值	**1.47037** **(1.922)**	**1.53981** **(1.648)**	**1.58373** **(1.43)**	**1.63441** **(1.324)**	**1.66233** **(1.213)**	**1.70615** **(1.178)**	**1.72188** **(1.111)**	**1.73177** **(1.065)**	**1.74048** **(1.015)**
GMF 专业知识	1.49874 (1.963)	1.57277 (1.689)	1.61625 (1.463)	1.66444 (1.351)	1.69199 (1.239)	1.73293 (1.202)	1.74812 (1.133)	1.75649 (1.087)	1.76487 (1.035)
GMF Softmax	1.47707 (1.932)	1.54383 (1.654)	1.58617 (1.433)	1.63575 (1.326)	1.66342 (1.214)	1.70677 (1.179)	1.72239 (1.112)	1.73223 (1.066)	1.74092 (1.016)
多层感知机 IPA	1.56623 (1.959)	1.61737 (1.655)	1.64909 (1.431)	1.69132 (1.32)	1.71029 (1.209)	1.74683 (1.169)	1.75711 (1.105)	1.76368 (1.058)	1.76877 (1.008)
MLP 平均值 DeepGroup	1.61669 (2.032)	1.68103 (1.718)	1.71103 (1.492)	1.75681 (1.368)	1.77828 (1.254)	1.81759 (1.217)	1.82764 (1.148)	1.83403 (1.098)	1.84177 (1.049)
MLP 专业知识	1.64412 (2.059)	1.70965 (1.747)	1.74467 (1.517)	1.78446 (1.386)	1.80671 (1.276)	1.84221 (1.232)	1.85095 (1.163)	1.85803 (1.116)	1.86351 (1.063)
MLP Softmax	1.62458 (2.038)	1.68445 (1.723)	1.7153 (1.496)	1.75838 (1.368)	1.78024 (1.256)	1.81848 (1.218)	1.82855 (1.149)	1.83446 (1.099)	1.84249 (1.049)

(c) MyAnimeList

表 5：平均最大误差

模型 \组大小	2	3	4	5	6	7	8	9	10
GMF IPA GMF 平均值	**1.02112** **(0.6)**	**1.2213** **(0.598)**	1.35658 (0.588)	**1.45195** **(0.583)**	**1.54115** **(0.578)**	1.59908 (0.577)	1.64979 (0.573)	1.69807 (0.576)	1.73598 (0.571)
GMF 专业知识	1.02474 (0.602)	1.22441 (0.599)	1.35928 (0.59)	1.4551 (0.584)	1.54414 (0.579)	1.60112 (0.577)	1.65118 (0.574)	1.69926 (0.576)	1.7366 (0.571)
GMF Softmax	1.02191 (0.6)	1.22143 (0.598)	**1.35653** **(0.588)**	1.45202 (0.583)	1.54118 (0.578)	**1.59897** **(0.577)**	**1.64965** **(0.573)**	**1.69792** **(0.576)**	**1.7358** **(0.571)**
多层感知机 IPA	1.04098 (0.642)	1.25715 (0.614)	1.38214 (0.602)	1.47972 (0.591)	1.56516 (0.586)	1.62067 (0.581)	1.66905 (0.576)	1.71687 (0.58)	1.7528 (0.573)
MLP 平均值 DeepGroup	1.07144 (0.666)	1.28743 (0.643)	1.41937 (0.635)	1.51234 (0.633)	1.59812 (0.638)	1.65421 (0.642)	1.70673 (0.641)	1.75552 (0.647)	1.79703 (0.645)
MLP 专业知识	1.07476 (0.667)	1.28932 (0.644)	1.42078 (0.635)	1.51325 (0.634)	1.59883 (0.638)	1.65371 (0.64)	1.70637 (0.64)	1.75428 (0.645)	1.79566 (0.643)
MLP Softmax	1.07226 (0.666)	1.28716 (0.643)	1.41906 (0.635)	1.51227 (0.633)	1.59788 (0.638)	1.65391 (0.641)	1.7066 (0.641)	1.75515 (0.646)	1.79669 (0.645)

(a) MovieLens1M

模型 \组大小	2	3	4	5	6	7	8	9	10
GMF IPA GMF 平均值	0.85192 (0.555)	1.13757 (0.573)	1.27696 (0.573)	1.36783 (0.57)	1.44244 (0.571)	1.51262 (0.575)	1.56818 (0.573)	1.62204 (0.571)	1.6636 (0.567)
GMF 专业知识	0.85171 (0.549)	1.13845 (0.571)	1.27815 (0.571)	1.36895 (0.567)	1.4425 (0.569)	1.51199 (0.573)	1.56717 (0.57)	1.62058 (0.569)	1.66146 (0.565)
GMF Softmax	0.85147 (0.552)	1.13761 (0.572)	1.27708 (0.572)	1.3679 (0.569)	1.44234 (0.571)	1.51245 (0.575)	1.56799 (0.572)	1.62184 (0.571)	1.66335 (0.567)
多层感知机 IPA	**0.78405** **(0.555)**	1.11519 (0.564)	1.24935 (0.56)	1.33899 (0.555)	1.41348 (0.556)	1.48001 (0.557)	1.5341 (0.554)	1.58622 (0.552)	1.62791 (0.549)
MLP 平均值 DeepGroup	0.79205 (0.553)	**1.11222** **(0.557)**	**1.24695** **(0.557)**	1.33662 (0.555)	1.4118 (0.557)	1.47902 (0.559)	1.53438 (0.556)	1.58697 (0.555)	1.62916 (0.55)
MLP 专业知识	0.79191 (0.56)	1.11441 (0.556)	1.24814 (0.555)	1.33715 (0.551)	**1.41142** **(0.554)**	**1.47765** **(0.556)**	**1.53235** **(0.553)**	**1.58432** **(0.551)**	**1.62581** **(0.547)**
MLP Softmax	0.79235 (0.555)	1.11261 (0.556)	1.24698 (0.556)	**1.33654** **(0.554)**	1.41161 (0.556)	1.47872 (0.558)	1.53406 (0.555)	1.58662 (0.554)	1.62879 (0.55)

(b) FilmTrust

模型 \组大小	2	3	4	5	6	7	8	9	10
GMF IPA GMF 平均值	**1.30264** **(0.845)**	**1.58715** **(0.866)**	**1.79755** **(0.864)**	**1.9702** **(0.877)**	**2.10648** **(0.876)**	**2.22836** **(0.896)**	**2.31758** **(0.895)**	**2.39541** **(0.91)**	**2.47078** **(0.919)**
GMF 专业知识	1.31784 (0.858)	1.60964 (0.883)	1.8266 (0.882)	1.99902 (0.894)	2.13631 (0.891)	2.25748 (0.91)	2.34706 (0.908)	2.42357 (0.922)	2.49875 (0.932)
GMF Softmax	1.30626 (0.847)	1.59061 (0.869)	1.80139 (0.867)	1.97309 (0.879)	2.10896 (0.878)	2.23046 (0.898)	2.31946 (0.896)	2.39701 (0.911)	2.47222 (0.92)
多层感知机 IPA	1.34151 (0.874)	1.63805 (0.873)	1.84317 (0.862)	2.01188 (0.868)	2.1406 (0.863)	2.25649 (0.876)	2.33763 (0.875)	2.4112 (0.889)	2.4836 (0.896)
MLP 平均值 DeepGroup	1.36617 (0.893)	1.66385 (0.902)	1.8675 (0.896)	2.03818 (0.907)	2.17095 (0.906)	2.28936 (0.924)	2.37588 (0.921)	2.4495 (0.934)	2.52558 (0.943)
MLP 专业知识	1.38038 (0.9)	1.68165 (0.913)	1.89237 (0.908)	2.0611 (0.918)	2.1951 (0.914)	2.31311 (0.929)	2.39796 (0.923)	2.47081 (0.935)	2.54585 (0.943)
MLP Softmax	1.37076 (0.894)	1.66676 (0.904)	1.87147 (0.898)	2.04076 (0.908)	2.17377 (0.907)	2.29111 (0.924)	2.37734 (0.921)	2.45073 (0.934)	2.52696 (0.943)

（c）MyAnimeList

表 6：折扣累计增益

模型 \组大小	2	3	4	5	6	7	8	9	10
GMF IPA GMF 平均值	**0.98021 (0.024)**	**0.98543 (0.018)**	0.9886 (0.015)	**0.99067 (0.012)**	0.99178 (0.011)	0.99286 (0.01)	0.99358 (0.009)	**0.99415 (0.008)**	**0.99457 (0.008)**
GMF 专业知识	0.97995 (0.024)	0.98528 (0.019)	0.98853 (0.015)	0.99058 (0.012)	0.99171 (0.011)	0.99282 (0.01)	0.99351 (0.009)	0.99411 (0.008)	0.99454 (0.008)
GMF Softmax	0.98006 (0.024)	0.98536 (0.019)	**0.9886 (0.015)**	0.99064 (0.012)	**0.99178 (0.011)**	**0.99288 (0.01)**	**0.9936 (0.009)**	0.99415 (0.008)	0.99456 (0.008)
多层感知机 IPA	0.97854 (0.026)	0.98342 (0.02)	0.98689 (0.016)	0.98906 (0.014)	0.99046 (0.012)	0.99178 (0.011)	0.99251 (0.01)	0.99303 (0.009)	0.99358 (0.009)
MLP 平均值 DeepGroup	0.97778 (0.026)	0.98247 (0.021)	0.98556 (0.017)	0.98762 (0.015)	0.98894 (0.014)	0.98999 (0.013)	0.99064 (0.012)	0.99109 (0.011)	0.99152 (0.011)
MLP 专业知识	0.97777 (0.026)	0.98251 (0.021)	0.9855 (0.017)	0.98763 (0.015)	0.98894 (0.014)	0.99004 (0.013)	0.99071 (0.012)	0.99116 (0.011)	0.9916 (0.011)
MLP Softmax	0.97784 (0.026)	0.98248 (0.021)	0.98554 (0.017)	0.98762 (0.015)	0.98895 (0.014)	0.98998 (0.013)	0.99069 (0.012)	0.9911 (0.011)	0.99153 (0.011)

(a) MovieLens1M

模型 \组大小	2	3	4	5	6	7	8	9	10
GMF IPA GMF 平均值	0.96788 (0.028)	0.97795 (0.022)	0.9815 (0.019)	0.98393 (0.016)	0.98596 (0.014)	0.9876 (0.013)	0.98869 (0.011)	0.98967 (0.011)	0.99038 (0.01)
GMF 专业知识	0.96795 (0.028)	0.97794 (0.022)	0.9815 (0.019)	0.98392 (0.016)	0.98593 (0.014)	0.9876 (0.013)	0.98868 (0.012)	0.98965 (0.011)	0.99038 (0.01)
GMF Softmax	0.96795 (0.028)	0.97796 (0.022)	0.9815 (0.019)	0.98394 (0.016)	0.98596 (0.014)	0.9876 (0.013)	0.98868 (0.012)	0.98967 (0.011)	0.99038 (0.01)
多层感知机 IPA	**0.97057 (0.027)**	0.9788 (0.023)	0.98249 (0.019)	0.98523 (0.016)	0.98704 (0.015)	**0.98896 (0.012)**	0.98994 (0.011)	**0.9911 (0.01)**	0.99165 (0.01)
MLP 平均值 DeepGroup	0.96897 (0.027)	0.9789 (0.022)	0.98275 (0.018)	0.98527 (0.016)	**0.98729 (0.014)**	0.98895 (0.012)	**0.98998 (0.011)**	0.99104 (0.01)	0.99166 (0.009)
MLP 专业知识	0.9694 (0.027)	**0.97897 (0.022)**	0.98276 (0.018)	**0.98532 (0.016)**	0.98724 (0.014)	0.98893 (0.012)	0.98993 (0.011)	0.99106 (0.01)	0.99166 (0.009)
MLP Softmax	0.9693 (0.027)	0.97891 (0.022)	**0.98278 (0.018)**	0.98527 (0.016)	0.98728 (0.014)	0.98895 (0.012)	0.98998 (0.011)	0.99106 (0.01)	**0.99167 (0.009)**

(b) FilmTrust

模型 \组大小	2	3	4	5	6	7	8	9	10
GMF IPA GMF 平均值	0.98898 (0.014)	**0.99218 (0.01)**	**0.99401 (0.008)**	0.9949 (0.007)	0.99571 (0.006)	**0.99615 (0.005)**	0.99662 (0.005)	0.99679 (0.005)	0.99714 (0.004)
GMF 专业知识	0.98893 (0.014)	0.99209 (0.01)	0.99383 (0.008)	0.99479 (0.007)	0.99566 (0.006)	0.99609 (0.005)	0.99658 (0.005)	0.99674 (0.005)	0.99708 (0.004)
GMF Softmax	**0.98898 (0.014)**	0.99217 (0.01)	0.99399 (0.008)	**0.99492 (0.007)**	**0.99572 (0.006)**	0.99615 (0.005)	**0.99664 (0.005)**	**0.99679 (0.005)**	**0.99715 (0.004)**
多层感知机 IPA	0.98723 (0.015)	0.99062 (0.011)	0.99255 (0.009)	0.9936 (0.008)	0.99458 (0.007)	0.99507 (0.006)	0.99573 (0.006)	0.99591 (0.005)	0.99632 (0.005)
MLP 平均值 DeepGroup	0.9868 (0.016)	0.99023 (0.012)	0.99212 (0.01)	0.99307 (0.008)	0.99415 (0.007)	0.99459 (0.007)	0.99519 (0.006)	0.99537 (0.006)	0.99565 (0.005)
MLP 专业知识	0.9867 (0.016)	0.99021 (0.012)	0.9921 (0.01)	0.99312 (0.008)	0.99414 (0.007)	0.99458 (0.007)	0.99521 (0.006)	0.99542 (0.006)	0.9957 (0.005)
MLP Softmax	0.98684 (0.016)	0.99028 (0.012)	0.99211 (0.01)	0.99308 (0.008)	0.99416 (0.007)	0.9946 (0.007)	0.99521 (0.006)	0.99539 (0.006)	0.99566 (0.005)

(c) MyAnimeList

4 讨论

本研究的主要目标是评估不同的聚合技术，以便为群组提供推荐。如第3节所示，我们可以看到以下因素会带来不同趋势：a) 所使用的模型；b) 聚合群体信息的方式；c) 模型所应用的数据集；以及 d) 群组规模。

聚焦于模型，我们可以看到，具有多个隐藏层的多层感知机（MLP）取得了更低的平均绝对误差（MAE）；然而，较简单的模型广义矩阵分解（GMF）取得了更低的均方误差（MSE）。尽管多层感知机模型在这类问题上表现出强大的能力，但它似乎会过拟合，为群组中的部分用户生成非常好的推荐，却给其他用户带来较差的推荐，因此均方误差（MSE）值较高。另一方面，广义矩阵分解（GMF）模型能够降低各群组中的最大误差，这意味着群组中没有用户会受到推荐的严重影响。从结果中我们还可以观察到，最大误差较高的模型会导致物品排序与用户偏好更加不符，并且在NDCG指标上的表现更差。

观察用户聚合可以发现，表现最佳的用户聚合方式是平均值，其次是表现非常相近的 Softmax。然而，不使用 Softmax 的专家用户加权会产生更差的结果。根据结果可以发现，对于不采用包含多个隐藏层的深度架构的模型，当聚合函数是对潜在因子进行线性变换（广义矩阵分解）时，IPA 和 GPA 策略产生的结果相近。不过，我们也可以看到，多层感知机的非线性会导致这两种策略产生不同的结果。

对于不同的数据集，可以发现一个明显趋势：在用户、物品和投票数量众多的复杂数据集（如 Movilens 或 MyAnimeList）上，表现最佳的模型具有一致的趋势；而在投票数量较少的 FilmTrust 数据集上，则没有明显趋势。

就群组规模而言，群组中的用户越多，用户偏好之间出现差异的可能性就越大。因此，群组规模越大，所有误差指标的值也越高。

5 结论与未来工作

随着神经网络进入协同过滤领域，人们开始探索利用其从用户偏好中发现非线性模式、进而生成更优预测的能力。要使用这些系统为一组用户生成推荐，我们需要聚合他们的偏好。正如本研究所示，聚合可以在多个关键环节进行。GPA 策略会在模型之前或模型内部进行聚合，因此能够考虑整个群组的偏好，并且只需一个前馈步骤即可生成预测。与之不同的是，IPA 策略需要为每位用户生成多次预测，并在模型之后进行聚合。本研究测试了不同的 GPA 方法在

不同数据集上的表现，并对比了不同指标。

作为未来的工作，有两个关键因素需要考虑。首先，在本研究中，研究人员设计了提供给模型的用户聚合技术；未来的工作将探索这些函数在不同机器学习模型中的应用。第二个关键点是，在这项工作中，模型将从针对个体训练的模型中进行知识迁移，以生成群组预测；研究表明，尽管模型性能较高（平均绝对误差有所改善），但在群组场景中容易过拟合（群组预测误差更大，导致均方误差更差）。为了解决这一问题，未来的工作将在个体训练之后，尝试为群组进行专门化训练阶段。

声明

本文作者声明，他们不存在利益冲突。

本项工作由西班牙的 *Ministerio de Ciencia e Inno-vación* 和欧洲区域发展基金（FEDER）共同资助，资助项目为 PID2019-106493RB-I00(DL-CEMG) 和 *Comunidad deMadrid*，依据 *Convenio Plurianual*，并通过 *Universidad Politécnica de Madrid* *Programa de Excelencia para el Profeso-* 开展行动。 *rado Universitario*

数据可用性声明

MovieLens1M、FilmTrust 和 MyAnimeList 数据集，以及支持本研究发现的实验源代码，均可在 neural-cf-for-groups 的 GitHub 存储库中获取 [https:// github.com/KNODIS-Research-Group/neural-cf-for-groups]。

参考文献

[1] Roza Abolghasemi、Paal Engelstad、Enrique Herrera-Viedma 和 Anis Yazidi。基于成对偏好的个性化群组推荐系统。 *Information Sciences*，595:1–17，2022年。[2] Linas Baltrunas、Tadas Makcinskas 和 Francesco Ricci。结合排名聚合与协同过滤的群组推荐。载于 *Proceedings of the Fourth ACM Conference on Recommender Systems* 10, 119 126, 2010 ,

RecSys ' 页 – 纽约，纽约州，美国，。计算机协会。[3] Reza Barzegar Nozari 和 Hamidreza Koohi。一种基于成员影响力和领导者影响力的新型群组推荐系统。*Knowledge-Based Systems*，205:106296，2020年。[4] Zeynep Batmaz、Ali Yurekli、Alper Bilge 和 Cihan Kaleli。推荐系统深度学习综述：挑战与应对。*Artificial Intelligence Review*，52(1):1–37，2019年6月。

[5] J. Bobadilla, ´A Gonz´ález-Prieto、F. Ortega 和 R. Lara-Cabrera。通过深度学习特征选择揭示人口统计推荐系统中的因子。*Neural Computing and Applications*，33(12):7291–7308，2021年6月。[6] Jesus Bobadilla、Santiago Alonso 和 Antonio Hernando。用于协同过滤推荐系统的深度学习架构。*Applied Sciences*，10(7)，2020年。[7] Jes´us Bobadilla, ´Angel Gonz´ález-Prieto、Fernando Ortega 和 Ra´ul 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年2月。[8] Erion C¸ano 和 Maurizio Morisio。混合推荐系统：系统性文献综述。*Intell. Data Anal.*，21(6):1487–1524，2017年1月。[9] Sriharsha Dara、C. Ravindranath Chowdary 和 Chintoo Kumar。群体推荐系统综述。*Journal of Intelligent Information Systems* 54 2 271 295, 2020 , ( ): – 4 月。[10] Yashar Deldjoo 、 Markus Schedl 、 Paolo Cremonesi 和 Gabriella Pasi。利用多媒体内容的推荐系统。*ACM Comput. Surv.* 53 5 2020 , ( ), 9月。[11] Shanshan Feng、Huaxiang Zhang、Lei Wang、Li Liu 和 Yuchang Xu。检测多源信息中隐藏的潜在关联，以改进群组推荐。*Know.-Based Syst.*，171(C):56–68，2019年5月。[12]Saman Forouzandeh、Kamal Berahmand 和 Mehrdad Rostami。采用集成学习和图嵌入的推荐系统：以 MovieLens 为例。*Multimedia Tools and Applications* 80 5 7805 7832, 2021 , ( ): – 2 月。[13] Guibing Guo 、 Jie Zhang 和 Neil Yorke-Smith。一种新颖的推荐系统冈吐斯相似度度量。载于 *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence* 13, 2619 2625, 2013 , IJCAI ' 页 – 门洛帕克，加利福尼亚，。AAAI Press。[14] Junpeng Guo、Yanlin Zhu、Aiai Li、Qipeng Wang 和 Weiguo Han。一种用于群组推荐系统中群组用户建模的社会影响方法。 *IEEE Intelligent Systems*，31(5):40–48，2016年。[15] F. Maxwell Harper 和 Joseph A. Konstan。MovieLens数据集：历史与背景。 *ACM Trans. Interact. Intell. Syst.*，5(4)，2015年12月。[16] Xiangnan He、Lizi Liao、Hanwang Zhang、Liqiang Nie、Xia Hu 和 Tat-Seng Chua。神经协同过滤。载于 *Proceedings of the26th International Conference on World Wide Web*，WWW '17，第173–182页，日内瓦共和国和州，CHE，2017年。国际万维网会议指导委员会。[17] Liang Hu、Jian Cao、Guandong Xu、Longbing Cao、Zhiping Gu 和 Wei Cao。面向群组推荐的群组偏好深度建模。载于 *Proceedings of the Twenty-Eighth AAAI Conference*

on Artificial Intelligence, AAAI'14，第 1861–1867 页，帕洛阿尔托，加利福尼亚州，2014年。AAAI Press。[18] Tianlin Huang、De fu Zhang 和 Lvqing Bi。面向推荐系统的神经嵌入协同过滤。*Neural Computing and Applications* 1 15, 2020 ，第 – 页。[19] Firat Ismailoglu。在群组推荐系统中聚合用户偏好：一种众包方法。*Decision Support Systems*，152 113663，2022年：。[20] Saurabh Kulkarni 和 Sunil F. Rodd。上下文感知推荐系统：当前最先进技术综述。*Computer Science Review* 37 100255, 2020 ，：。

[21] F. Ortega、J. Bobadilla、A. Hernando 和 A. Guti´eRrez。将群体推荐纳入推荐系统：方案与性能。 *Inf. Process. Manage.*，49(4):895–901，2013年7月。[22] Fernando Ortega、Antonio Hernando、Jesus Bobadilla 和 Jeon Hyung Kang。使用基于矩阵分解的协同过滤为用户群组推荐物品。*Information Sciences* 345 313 324, 2016 ，：–。

[23] Sarina Sajjadi Ghaemmaghami 和 Amirali Salehi-Abari。*Deep-Group: Group Recommendation with Implicit Feedback*，第 3408–3412 页。美国计算机协会，纽约，纽约州，美国，2021年。[24] Ruslan Salakhutdinov 和 Andriy Mnih。概率矩阵分解。载于 *Proceedings of the 20th International Conference on Neural Information Processing Systems*，NIPS'07，第 1257–1264 页，雷德胡克，纽约州，美国，2007年。Curran Associates Inc。[25] Jyoti Shokeen 和 Chhavi Rana。社交推荐系统特征研究。*Artificial Intelligence Review*，53(2):965–988，2020年2月。[26] Xiangshi Wang、Lei Su、Qihang Zhou、Liping Wu 和 Yin Zhang。基于可信社交网络中成员偏好的群体推荐系统。 *Sec. and Commun. Netw.*，2020年，2020年1月。