Software Engineering and Applications 软件工程与应用, 2024, 13(4), 593-605
Published Online August 2024 in Hans. https://www.hanspub.org/journal/sea
https://doi.org/10.12677/sea.2024.134061

Hans汉斯

融合时间特征的联邦矩阵分解推荐算法

贾毅恒，何利文

南京邮电大学物联网学院，江苏 南京

收稿日期：2024年7月3日；录用日期：2024年8月21日；发布日期：2024年8月28日

摘 要

目前矩阵分解推荐系统在集中环境下存在隐私泄露的风险，且更多的数据拥有者不愿提供自身的数据，应用于分布式环境下的联邦矩阵分解推荐系统应用而生。传统的联邦矩阵分解模型在数据稀疏的情况下推荐准确率低，没有考虑用户的兴趣随时间变化的动态性。本文针对以上问题，引入联邦矩阵分解模型与时间隐语义模型相结合，提出一种融合时间特征的联邦矩阵分解推荐算法TF-FedMF (Federated Matrix Factorization Recommendation Algorithm with Temporal Feature Integration)。该算法在联邦矩阵分解框架中加入时间特征，用于捕捉用户行为随时间变化的趋势，提高了推荐系统的时效性和准确性；同时，结合同态加密对上传的梯度信息进行加密，增强算法的安全性。通过MovieLens数据集进行实验对比，实验结果表明，所提出的算法较其它算法在兼顾用户隐私安全性的同时，具有较高的推荐准确性。

关键词

矩阵分解，联邦学习，同态加密，时间特征，隐语义模型

Federated Matrix Factorization Recommendation Algorithm with Temporal Feature Integration

Yiheng Jia, Liwen He

School of Internet of Things, Nanjing University of Posts and Telecommunications, Nanjing Jiangsu

Received: Jul. 3rd, 2024; accepted: Aug. 21st, 2024; published: Aug. 28th, 2024

Abstract

At present, the matrix decomposition recommendation system has the risk of privacy leakage in a centralized environment, and more data owners are unwilling to provide their own data. Therefore,

文章引用: 贾毅恒, 何利文. 融合时间特征的联邦矩阵分解推荐算法[J]. 软件工程与应用, 2024, 13(4): 593-605.
DOI: 10.12677/sea.2024.134061

the application of the federated matrix decomposition recommendation system in a distributed environment has emerged. The traditional federated matrix decomposition model has low recommendation accuracy when the data is sparse, and does not consider the dynamic nature of user interests changing over time. In view of the above problems, this paper introduces the combination of the federated matrix decomposition model and the temporal latent semantic model, and proposes a federated matrix decomposition recommendation algorithm TF-FedMF (Federated Matrix Factorization Recommendation Algorithm with Temporal Feature Integration) with temporal feature integration. The algorithm adds temporal features to the framework of federated matrix decomposition to capture the trend of user behavior changing over time, thereby improving the timeliness and accuracy of the recommendation system; at the same time, the uploaded gradient information is encrypted by combining homomorphic encryption to enhance the security of the algorithm. An experimental comparison is carried out on the MovieLens dataset. The experimental results show that the proposed algorithm has higher recommendation accuracy than other algorithms while taking into account user privacy and security.

Keywords

Matrix Factorization, Federated Learning, Homomorphic Encryption, Temporal Features, Latent Semantic Model

1. 引言

随着大数据时代政府隐私保护法规和法律的盛行，隐私保护机器学习在学术界和工业界获得了迅速增长的兴趣，在实现隐私保护机器学习的各种技术中，联邦(机器)学习(Federated Learning, FL)最近受到了高度关注。FL 最初的想法是由 Google [1]提出的，其目标是根据分布在每个用户手机上的个人信息来学习一个中心化的模型。更重要的是，在模型训练过程中，不会将用户的原始个人信息传输到中央服务器，从而确保了隐私保护。此外，可以证明，与基于用户原始数据学习的传统模型相比，学习到的隐私保护模型具有几乎相同的预测能力。这凸显了 FL 的实用性。国内的王健宗等人[2] [3]在 2020 年对联邦学习方法作了总结。

随着联邦学习技术研究的不断深入，将联邦学习应用于推荐系统方面也逐步有一些探索。2019 年，Ammad-Ud-Din 等人[4]首次将联邦学习框架应用到协同过滤算法中，提出一种联邦学习联邦协同过滤算法，该算法由推荐服务器和参与方联合训练协同过滤模型。随后，微众银行在当前推荐系统中最常用的矩阵分解[5]和因子分解机[6]算法的基础上，提出了联邦矩阵分解(Federated Matrix Factorization, FedMF) [7]、联邦分解机(Federated factorization machine, FedFM)等联邦推荐算法。Qi 等人在 2020 年提出了一种联邦新闻推荐(Federated News Recommendations, FedNewsRec)方法[8]，用户在新闻平台(网站或应用程序)上的行为存储在用户本地设备上，不上传到服务器。服务器用于维护新闻推荐模型，并通过来自大量用户的模型梯度进行更新。Flanagan 等人于 2021 年提出了一种联邦多视图矩阵分解(Federated Multi-View Matrix Factorization, FED-MVMF)方法，这是第一个带有辅助信息源的联邦多视图矩阵分解方法[9]。但上述算法依然存在着用户隐私泄露问题。

McSherry 等人[10]首次在推荐系统中应用差分隐私，通过向项目之间的相似度协方差矩阵添加严格

定义的随机化噪声，验证其在推荐系统中的有效性。Liu 等人[11]将差分隐私机制引入模型的目标函数，实现了整个模型训练过程中的隐私保护。Yang 等人[12]针对推荐系统中的隐私问题，提出一种包括两种扰动方式的差分隐私协同过滤推荐框架，以防止推理攻击对用户造成的威胁。尽管上述差分隐私保护技术在一定程度上保护了个人数据隐私，但也可能因噪声过大而掩盖数据的原始特征，导致推荐系统准确性下降。

Erkin 等人[13]使用安全多方计算和部分同态加密技术保护推荐系统的隐私，该方法通过引入半信任的第三方在用户和推荐服务器之间进行数据隔离，从而保护用户隐私。Kim 等人[14]采用全同态加密技术对输入的用户评分矩阵进行加密，并利用这些已加密的数据训练矩阵分解模型来保护用户隐私。同年，张永棠[15]使用一种代换加密机制对客户端的用户评分数据进行加密，并将其发送到推荐服务器，推荐服务器再基于密文数据训练协同过滤推荐模型。相比基于隐私的随机推荐方法，上述基于加密的推荐方法能有效保证数据的原始性，但无法解决用户原始数据离开本地的问题。

FedMF 方法利用矩阵分解技术，采用分布式学习和同态加密方案，将每个用户的偏好数据(评分)保存在本地，利用传统的随机梯度下降法[16]，在每个参与者与第三方服务器之间不断交互更新加密梯度，最终满足最小损失函数，并得到每个参与者对于项目的预测评分，达到向参与者进行推荐的目的。FedMF 解决了传统矩阵分解推荐系统中用户偏好数据和特征向量泄露的隐私问题，确保用户信息始终留在本地，在防止隐私泄露的基础上对用户做出相应的推荐。对于用户来说，如何在保护个人隐私的同时，提高推荐准确率、对用户做出精准推荐成为亟待解决的问题。

基于此，本文提出了一种融合时间特征的联邦矩阵分解推荐算法(TF-FedMF)，为上述问题的解决提供了思路。该算法通过引入时间特征到矩阵分解中，旨在处理用户偏好的动态变化。为了应对推荐系统中的数据共享和数据隐私问题，算法采用了联邦学习机制。与此同时，采用同态加密技术，实现在密文上进行推荐模型的建立和推荐结果的求解，对上传梯度进行加密传输和存储，不仅在一定程度上保证了数据的真实性，在提供隐私保护的同时，实现了高精度的推荐服务。该算法不仅能够有效保护用户隐私，还能显著提升推荐系统的精度和性能。

2. 理论基础

2.1. 联邦矩阵分解

图 1 展示了 FedMF (联邦矩阵分解)框架。该算法解决了传统矩阵分解推荐系统中用户偏好数据和特征向量等隐私泄露问题，保证用户信息不离开本地，并为终端用户提供严格的隐私保护，同时算法的准确性接近集中式矩阵分解推荐算法的准确性，在这个框架中涉及两类参与者：服务器和用户。假设服务器是诚实但好奇的，用户是诚实的，并且用户的隐私受到服务器的保护。

在开始矩阵分解过程之前，需要对一些参数进行初始化。物品矩阵在服务器端初始化，用户矩阵在本地由每个用户初始化。具体训练过程如下：

1) 服务器使用公钥对物品矩阵 V 进行加密，得到密文 C_V。从此，最新的 C_V 为所有用户的下载做准备。

2) 每个用户从服务器下载最新的 C_V，并使用密钥对其进行解密，得到 V 的明文。利用 V 进行局部更新并计算梯度 G，然后使用公钥对 G 进行加密，得到密文 C_V。然后建立 TLS/SSL 安全信道，C_V 通过该安全信道发回服务器。

3) 在收到用户的密文梯度后，服务器使用该密文更新物品矩阵：$C_V^{t+1} = C_V^t - C_G$。之后，为用户下载准备最新的 C_V。

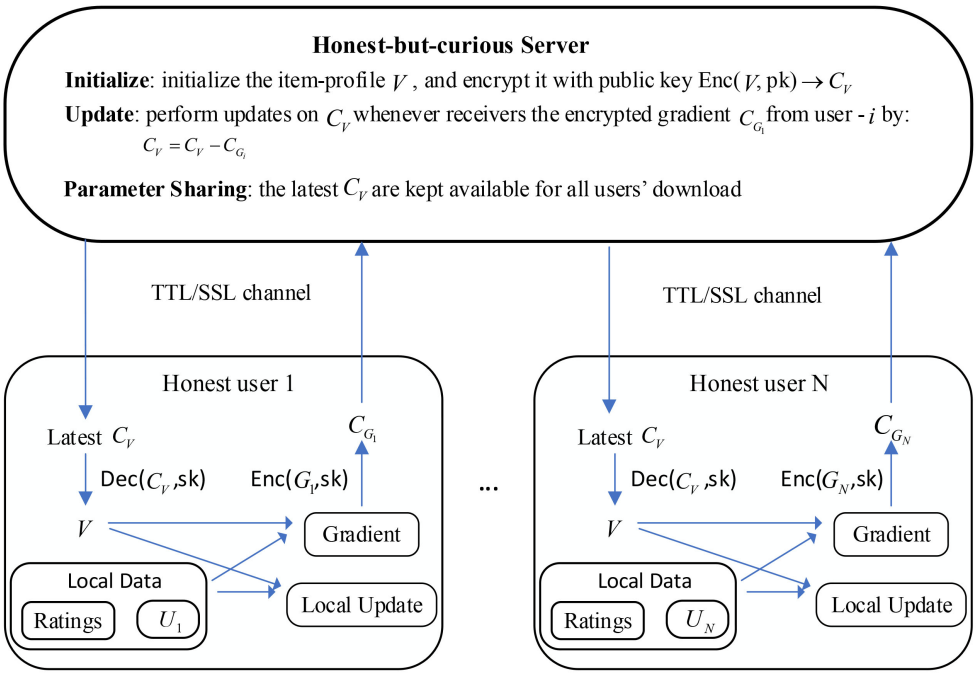

Figure 1. FedMF framework
图 1. FedMF 框架

2.2. 加法同态加密

同态加密(*HE*) [17]常被用于联邦学习中，通过提供加密参数交换来保护用户的隐私。*HE* 是一种特殊的加密方案，允许任何第三方对加密后的数据进行操作而无需事先解密。如果公式(1)成立，则称加密方案为关于一个运算“$*$”的同态：

$$E(m_1) \star E(m_2) = E(m_1 \star m_2), \forall m_1, m_2 \in M \tag{1}$$

其中，E 是一个加密算法，M 是所有可能的数据的集合[18]。

加法同态加密是关于加法的同态加密，通常，它由以下功能组成：

KeyGen：生成密钥对(*pk*, *sk*)，其中 *pk* 是公钥，*sk* 是私钥。

$Enc(m, pk)$：使用公钥 *pk* 加密消息 m，得到密文 c。

$Dec(c, sk)$：使用私钥 *sk* 解密密文 c，得到明文 m。

$Add(c_1, c_2)$：对两个密文 c_1 和 c_2 进行加法运算 $Enc(c_1 + c_2)$，得到密文 c_a。

$DecAdd(ca, sk)$：使用私钥 *sk* 解密密文 c_a，得到明文之和 m_a。

本文采用 Paillier [19]加法同态加密方案，该方案确保了密文计算结果和明文计算结果的同态性，即在密文下进行加法运算与在明文下进行加法运算遵循相同的规则。通过这种方式，可以直接对加密后的用户数据进行操作，从而在一定程度上保护了用户数据的安全。加密公式如式(2)所示：

$$E(m_1) + E(m_2) = E(m_1 + m_2), \forall m_1, m_2 \in M \tag{2}$$

公式(2)表示密文的求和等同于求和后的密文，E 是加密算法，M 是所有可能数据的集合。

2.3. 隐语义模型

隐语义模型也称因子模型[20]，是 Simon Funk 在矩阵分解技术的基础上，通过梯度下降法改进而来

的，核心思想是通过用户项目之间的隐含特征联系二者，挖掘分析用户历史行为，得到其潜在兴趣偏好。从矩阵分解的角度来看，隐语义模型是将评分矩阵 R 分解为两个低维矩阵的乘积来表示用户和物品之间的隐含关系，可表示为：

$$\hat{R}_{ui} = U \cdot V^{\mathrm{T}} \tag{3}$$

其中，$R_{ui} \in R^{A \times B}$ 表示 A 个用户和 B 个产品的评分集合，评分范围在 1~5 之间，U 是用户特征矩阵，V 是物品特征矩阵，$U,V \in R^{f}$。

用户 u 对物品 i 的评分预测矩阵为：

$$\hat{R}_{ui} = U^{\mathrm{T}} \cdot V + w_u + w_i + \mu \tag{4}$$

其中，μ 是全局偏置，w_u 和 w_i 分别是用户 u 和物品 i 的偏置，U 是用户 u 的特征向量，V 是项目 i 的特征向量。

3. 融合时间特征的联邦矩阵分解算法

3.1. 融合时间特征的矩阵分解推荐算法

用户对产品的评价和受欢迎程度会随着时间的推移不断变化，用户偏好和项目流行度也同样是动态变化的。而传统的矩阵分解(Matrix Factorization, MF)技术本质上是静态的。在公式(3)中定义的用户对物品评分的预测，是通过用户和物品的潜在特征交互，结合用户和物品的偏差项以及全局平均评分来计算得出的。这种矩阵分解方法在处理大规模推荐系统数据集时非常有效，能够捕捉用户和项目的潜在偏好和特性。然而，传统的 MF 方法无法处理用户－项目交互的动态效果。为了解决用户－项目交互中的时间效应和动态效应问题，提出了一种将时间特征加入到矩阵分解中的算法，即 TF-MF 算法(Matrix Factorization Recommendation Algorithm with Temporal Feature Integration)。该方法能够更好地捕捉用户和项目随时间变化的动态偏好和特性，从而提高推荐系统的准确性和适应性。

在公式(3)中加入时间依赖特征，如 $U_u(t)$、$w_u(t)$ 和 $w_i(t)$，以使其在 t 时刻对评分 $\hat{R}_{ui}(t)$ 进行动态预测，并在公式(4)中定义：

$$\hat{R}_{ui}(t) = U_u V_i^{\mathrm{T}} + w_u(t) + w_i(t) + U_u \times t_o + U_u(t) + \mu \tag{5}$$

其中，$U_u(t)$ 是时间的函数，V_i 保持不变，$U_u \times t_o$ 表示物品根据用户的不同而变化。由公式(5)定义的目标函数如下：

$$L = \sum \left(R_{ui}(t) - \hat{R}_{ui}(t) \right)^2 + \lambda \left(V_i + U_u + w_i + w_u + U_u(t) \right) \tag{6}$$

通过随机梯度下降(SGD) [21]更新模型参数如下：

$$U_{u,f} = U_{u,f} - \eta \frac{\partial L}{\partial U_{u,f}} \tag{7}$$

$$V_{i,f} = V_{i,f} - \eta \frac{\partial L}{\partial V_{i,f}} \tag{8}$$

$$w_u(t) = w_u(t) - \eta \frac{\partial L}{\partial w_u(t)} \tag{9}$$

$$w_i(t) = w_i(t) - \eta \frac{\partial L}{\partial w_i(t)} \tag{10}$$

进一步化简公式(7)~(10)分别如下：

$$U_{u,f} = U_{u,f} + \eta\left(e_{ui} \cdot V_{i,f} - \lambda U_{u,f}\right) \tag{11}$$

$$V_{i,f} = V_{i,f} + \eta\left(e_{ui} \cdot U_{u,f} - \lambda V_{i,f}\right) \tag{12}$$

$$w_i(t) = w_i(t) + \eta\left(e_{ui} - \lambda w_i(t)\right) \tag{13}$$

$$w_u(t) = w_u(t) + \eta\left(e_{ui} - \lambda w_u(t)\right) \tag{14}$$

其中，$e_{ui} = y_{ui} - \hat{y}_{ui}$ 是误差项，η 是学习率。

3.2. 融合时间特征的联邦矩阵分解推荐算法

基于 TF-MF 算法，引入联邦学习的概念，即 TF-FedMF 算法(Federated Matrix Factorization Recommendation Algorithm with Temporal Feature Integration)，以实现不收集用户偏好和项目流行度随时间的变化的信息，通过客户端本地为用户提供准确的产品推荐。在 TF-FedMF 算法中，数据源是分布式的，不存储在第三方服务器上，用户数据及用户带有时间特征的数据仅在用户本地客户端可用，而物品数据及物品带有时间特征的数据则在第三方服务器上存储和共享。同时，为了解决这种信息泄露问题，提出对梯度进行编码，使服务器无法对编码过程进行反转，编码数据也不会泄露任何信息。同时，服务器仍然应该能够使用编码后的梯度进行更新。实现这样一个目标的方法之一就是使用同态加密。基于以上，提出 TF-FedMF 算法，该算法能够在保护用户隐私的同时，实现准确的产品推荐。

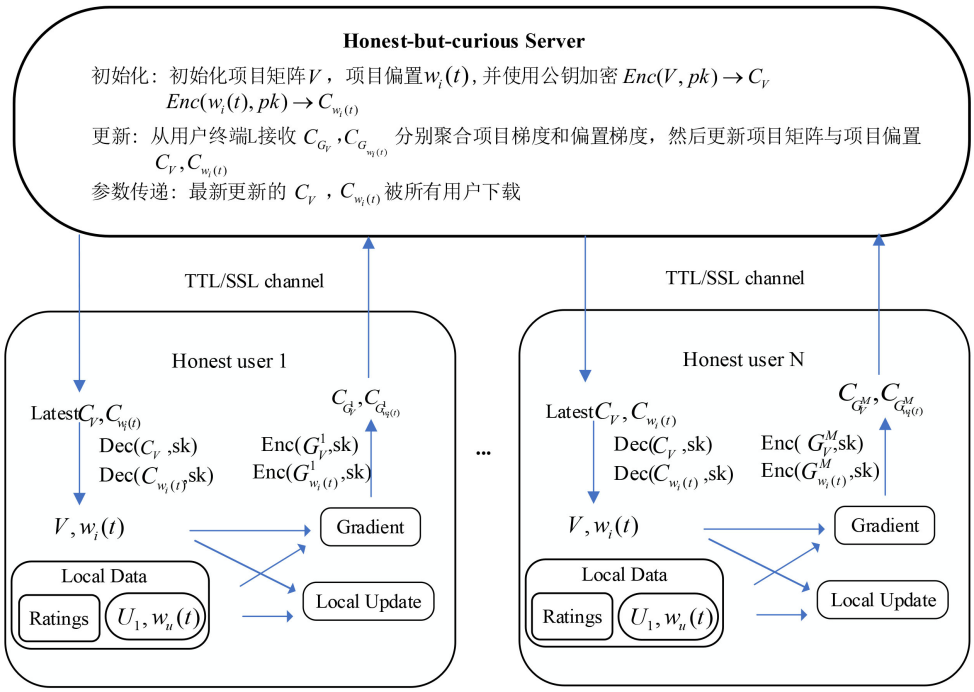

Figure 2. TF-FedMF framework
图 2. TF-FedMF 框架

如图 2 所示，展示了 TF-FedMF 算法框架。在这个框架中涉及两类参与者：服务器和用户。假设服务器是诚实但好奇的，用户是诚实的，并且用户的隐私受到服务器的保护。

密钥生成：章节 2.2 作为同态加密中涉及的典型函数，首先生成公钥和私钥。密钥生成过程是在其

中一个用户上进行的。公钥为包括服务器在内的所有参与者所知。并且密钥只在用户之间共享，需要对服务器进行保护。密钥生成后，会为公钥和私钥建立不同的 TLS/SSL。

参数初始化：在启动矩阵分解过程之前，需要对一些参数进行初始化。服务器端初始化物品矩阵 V，物品偏置项 $w_i(t)$，每个用户在本地初始化用户偏置项 $w_u(t)$ 及用户矩阵 U。

具体过程如下：

1) 参数初始化：服务器端初始化项目矩阵 V，用户偏置项 $w_u(t)$，物品偏置项 $w_i(t)$，并使用公钥进行加密，得到密文 $C_V = Enc(V, pk)$ 及 $C_{w_i(t)} = Enc(w_i(t), pk)$，最新的 C_V 及 $C_{w_i(t)}$ 为所有用户的下载做准备。

2) 用户终端更新。用户终端从服务器下载最新的 C_V 及 $C_{w_i(t)}$，并使用密钥对其进行解密，获得 $w_i(t) = Dec(C_{w_i(t)}, sk)$，$C_V = Dec(C_V, sk)$，利用 V 和 $w_i(t)$ 进行局部更新并计算梯度：

$$\begin{aligned}\Delta U_{u,f} &= \eta\left(e_{ui}\cdot V_{i,f} - \lambda U_{u,f}\right)\\ \Delta V_{i,f} &= \eta\left(e_{ui}\cdot U_{u,f} - \lambda V_{i,f}\right)\\ \Delta w_i(t) &= \eta\left(e_{ui} - \lambda w_i(t)\right)\\ \Delta w_u(t) &= \eta\left(e_{ui} - \lambda w_u(t)\right)\end{aligned} \tag{15}$$

使用公钥对梯度进行加密，得到密文 $C_{G_V}, C_{G_U}, C_{G_{w_u(t)}}, C_{G_{w_i(t)}}$。建立 TLS/SSL 安全信道，$C_{G_V}, C_{G_U}, C_{G_{w_u(t)}}, C_{G_{w_i(t)}}$ 通过该安全信道发回服务器。

3) 在收到用户的密文梯度后，服务器使用该密文更新项目配置文件：

$$\begin{aligned}C_U &= C_U - \sum_{u=1}^{N} C_{G_U}\\ C_V &= C_V - \sum_{u=1}^{N} C_{G_V}\\ C_{w_i(t)} &= C_{w_i(t)} - \sum_{i=1}^{N} C_{G_{w_i(t)}}\\ C_{w_u(t)} &= C_{w_u(t)} - \sum_{u=1}^{N} C_{G_{w_u(t)}}\end{aligned} \tag{16}$$

4) 迭代以上步骤

根据 TF-FedMF 算法的训练过程，算法在执行过程中，客户端和服务器端仅交替更新模型参数。用户的原始评分信息始终保留在客户端本地，但评分信息仍然能够被共享。传送到服务器的仅是经过同态加密后的密文。只要同态加密系统确保密文的不可区分性，以抵御选择明文攻击[22]，任何比特的信息都不会泄露给服务器。这不仅能够实现用户隐私的有效保护，还能够提供准确的推荐结果。与传统的 FedMF 算法相比，本文提出的算法在客户端和服务器端之间引入了时间因素，证明了时间相关的用户和物品偏差能够被正确地加密、传输和更新。该改进能够显著提升推荐系统的准确性。在确保数据隐私的同时，TF-FedMF 算法通过融入时间特征，增强联邦学习框架下的推荐精度和模型的动态适应性。

4. 实验分析

针对所提的 TF-FedMF 算法，本节将在真实的电影评分数据集上进行实验分析，以验证所提算法能为用户评分数据提供严格的隐私保护，同时还具有较高的推荐准确性。

4.1. 实验数据集

本章实验选取公开 MovieLens 电影评分数据集中新的 ml-latest-small 数据集版本作为实验数据集。

ml-latest-small 数据集包含用户对电影的评分、评分时间、用户和电影的标签等信息，其中每个用户至少评分了 20 部电影。数据集的具体统计信息如表 1 所示。

Table 1. Statistical information of experimental dataset
表 1. 实验数据集统计信息

数据集	用户数量	电影数量	评分记录	评分范围	评分稀疏度
ml-latst-small	610	9724	100,000	1~5	98.3%

4.2. 评价指标

推荐准确度是衡量推荐算法优劣的关键指标。本文使用平均绝对误差(Mean Absolute Error, MAE)和均方根误差(Root Mean Square Error, RMSE)两个常用的评价指标对所提算法的准确性进行评估。MAE 表示预测评分与真实评分之间的绝对误差的平均值。MAE 值越小，精度越高。其定义如下：

$$\mathrm{MAE}=\sum_{i,j}\frac{\left|\hat{r}_{i,j}-r_{i,j}\right|}{n} \tag{17}$$

RMSE 表示预测评分与真实评分的偏方差的平方根与预测次数 n 的比值。RMSE 更苛刻地衡量了推荐系统的准确性。RMSE 越小，精度越高。其定义如下：

$$\mathrm{RMSE}=\sqrt{\frac{1}{n}\sum_{i,j}\left(\hat{r}_{i,j}-r_{i,j}\right)^2} \tag{18}$$

4.3. 参数设置

实验参数值的设定如下：正则化参数的默认值为 1e−6，学习率的默认值为 1e−3，隐含特征值 k 的默认值为 10，迭代次数 d 的默认值为 100。在本次实验中，每次不同的输出步长都重复实验五次，并将指标的平均值报告为最终结果。

同态加密参数设置：公钥长度设置为 1024，通信带宽设置为 1 Gb/s，矩阵分解过程中，用户和物品维度设置为 100。

实验环境：使用 5.1 GHz，6 核 CPU、32 GB RAM，操作系统为 Windows 11，编程语言为 Python，使用 gmpy 模块加速 python 中的同态加密部分。

4.4. 实验结果

1) 验证时间特征融入是否正确

为探究 FedMF 模型中时间特征融入矩阵分解可以正确处理用户偏好的动态性，从测试集的 RMSE 和 MAE 值进行具体测量，如表 2 所示。

Table 2. Comparison of time feature integration
表 2. 时间特征融入对比

	TF-MF	TF-FedMF	二者不同
RMSE	0.8229	0.8239	0.1%
MAE	0.6345	0.6346	0.01%

由表 2 可得出，TF-MF 和 TF-FedMF 在 RMSE 和 MAE 方面的差异非常小。这表明在该数据集和实

验设置下，联邦学习的分布式训练方法并没有显著影响模型的预测精度。两者的性能几乎相当，表明联邦学习在保持隐私的同时，仍然能够提供与集中式方法相近的预测精度。从而证明了时间特征矩阵分解可以在联邦学习框架下的正确融入。面对未来在大规模的场景训练中，很多数据无法进行集中学习，联邦学习的效果将会更加优秀，且具有更好的隐私保护安全性。

2) 分析迭代次数对算法性能的影响

由图 3 可知，在相同迭代次数下，TF-FedMF 比 FedMF 具有更小的 RMSE 值，说明 TF-FedMF 比 FedMF 具有更好的性能。

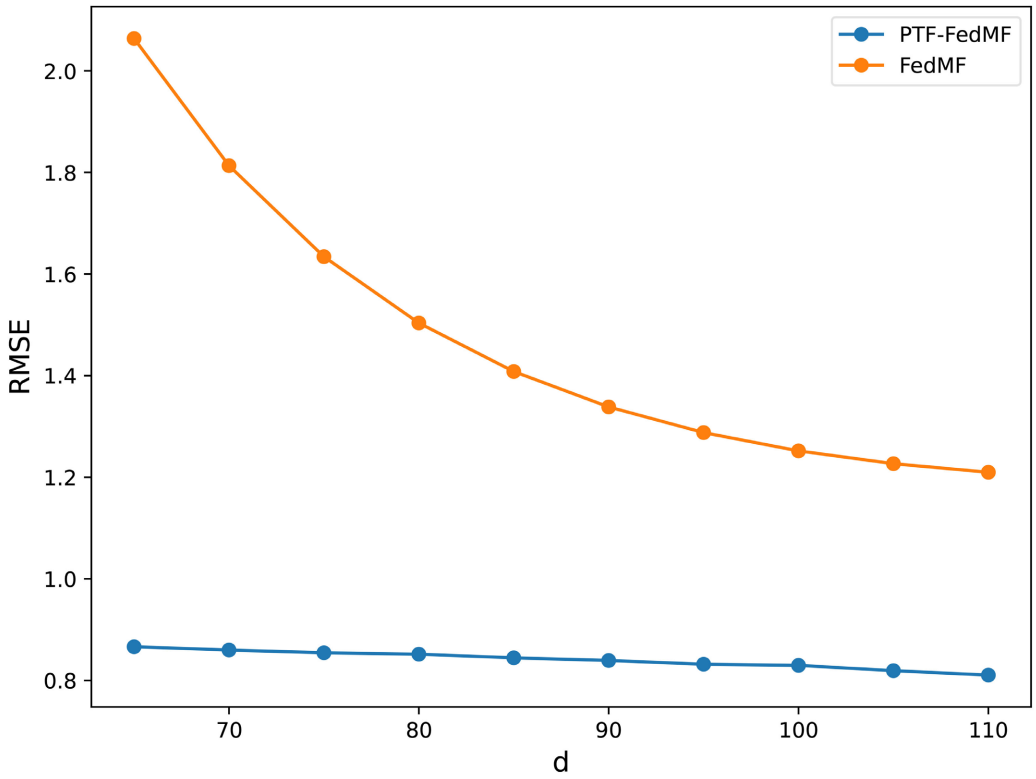

Figure 3. Comparison between TF-FedMF and FedMF
图 3. TF-FedMF 与 FedMF 对比

在整个训练过程中，TF-FedMF 的 RMSE 值相对稳定，波动较小，保持在 0.8 到 0.9 之间。相比之下，FedMF 的 RMSE 表现出显著的下降趋势，随着训练轮数(epochs)的增加，从最初的约 2 逐渐下降到接近 1.2。TF-FedMF 在训练期间的 RMSE 值更低且更稳定，表明其预测误差较小，模型性能较好。虽然 FedMF 在训练过程中的 RMSE 值有所下降，但其初期误差较大，最终的 RMSE 值仍高于 TF-FedMF，说明其预测误差较大，模型性能相对较差。

尽管 FedMF 在训练过程中表现出显著改进，但其最终的 RMSE 值仍高于 TF-FedMF。TF-FedMF 模型在不同的训练轮数下都表现出更低且更稳定的 RMSE 值，表明其在推荐系统中的预测性能优于 FedMF 模型。因此，在这两种模型中，TF-FedMF 具有更好的表现。

3) 学习率 lr 对算法性能影响

设置隐含特征值为 10，其他参数保持不变，根据前期的试验性训练，设置学习率的范围为 0.0005 到 0.001，学习率影响梯度下降过程中的步长。过大或过小的学习率都会影响模型的训练效果，而 0.0005 到

0.001 这个范围较小，使算法在一个相对稳定的区域内进行微调，从而找到相对较优的学习率。过大的学习率可能会导致模型训练过程中的震荡和发散，模型无法收敛到最优解。过小的学习率则可能导致模型收敛速度过慢，甚至陷入局部最优解，无法充分训练模型。

如图 4 和图 5 所示的实验结果可以看出：学习率为 0.0007 和 0.0008 时，模型在两个指标(RMSE 和 MAE)上均表现出较低的误差，说明这两个学习率能够较好地平衡模型的收敛速度和稳定性，是较优的选择。学习率为 0.0005 和 0.0006 时，模型在两个指标上均表现出较高的误差，说明这两个学习率过小，导致模型无法充分训练，误差较大。学习率为 0.0009 和 0.001 时，模型的误差相对稳定，但不如 0.0007 和 0.0008 表现优异，说明这些学习率虽然能够使模型收敛，但可能存在一定的震荡或未能达到全局最优。由此，可得出在给定的条件下，选择学习率为 0.0007 或 0.0008 能够获得较低的误差，是较为理想的学习率选择。

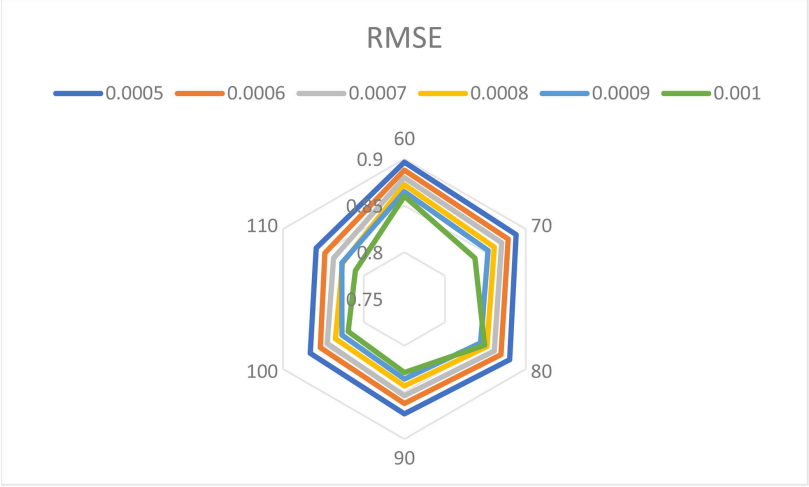

Figure 4. RMSE values under different learning rates
图 4. 不同学习率下的 RMSE 值

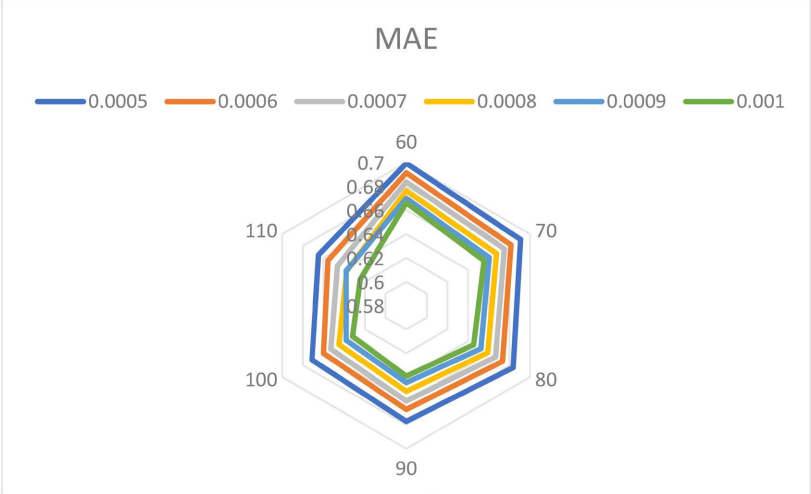

Figure 5. MAE values under different learning rates
图 5. 不同学习率下的 MAE 值

4) 分析隐含特征 L 对算法性能的影响

两个图均展示了随着隐含特征值的增加，模型的误差(无论是 RMSE 还是 MAE)均逐渐减小。这表明在给定的条件下，增加隐含特征值能够有效提升模型的性能，减少预测误差。

具体而言，如图 6 和图 7 所示，当隐含特征值达到 50 时，模型的误差达到最低值。通过这两个图，可以得出在给定的条件下，增加隐含特征值能够显著降低模型的误差，提高模型的预测准确性。选择较大的隐含特征值(如 50)是一个较优的选择。

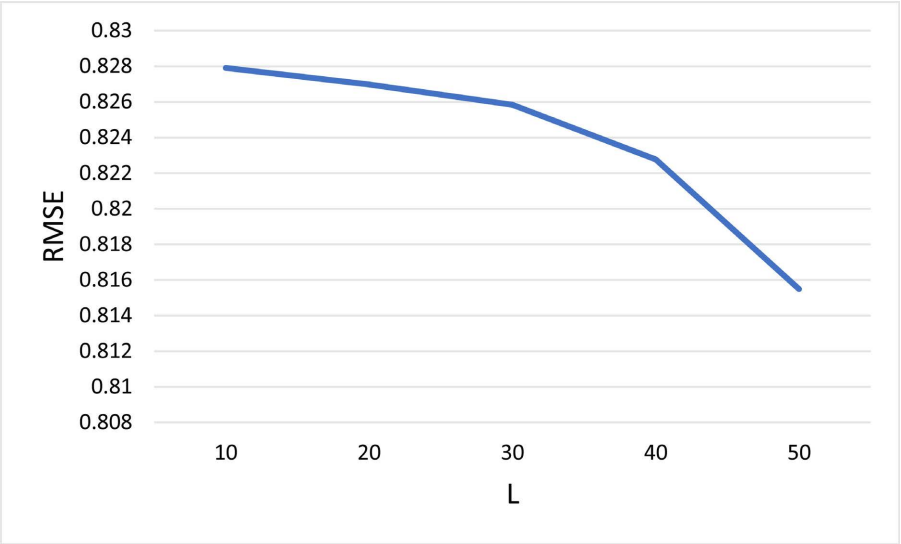

Figure 6. RMSE values under different implicit eigenvalues
图 6. 不同隐含特征值下的 RMSE 值

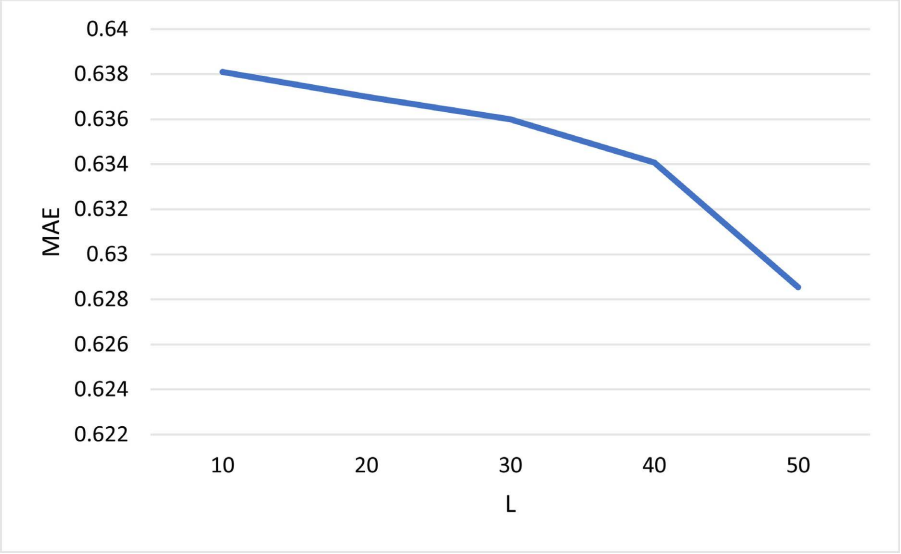

Figure 7. MAE values under different implicit eigenvalues
图 7. 不同隐含特征值下的 MAE 值

5. 结论

本文提出了一种融合时间特征的联邦矩阵分解推荐算法 TF-FedMF。该算法考虑时间特征，将时间特征与矩阵分解模型相结合，将用户因素作为时间的函数，保持项目因素不变，物品根据用户的不同而

变化，向用户推荐个性化物品。加入联邦学习后，在各用户评分数据不离开本地的条件下，提高了预测评分的准确性。引入同态加密技术，增强了算法的安全性。通过在基准电影数据集上的实验表明，TF-FedMF 在推荐精度上优于当前的 FedMF 模型，更适用于真实的推荐场景。但在 TF-FedMF 模型训练过程中，时间效率较低，如何提高 TF-FedMF 模型的时间效率将是我们未来进一步研究的方向。

参考文献

[1] Konecný, J., McMahan, H.B., Yu, F.X., *et al*. (2016) Federated Learning: Strategies for Improving Communication Efficiency. arXiv: 1610.05492.

[2] 王健宗, 孔令炜, 黄章成, 等. 联邦学习算法综述[J]. 大数据, 2020, 6(6): 64-82.

[3] 杨庚, 王周生. 联邦学习中的隐私保护研究进展[J]. 南京邮电大学学报(自然科学版), 2020, 40(5): 204-214.

[4] Ammad-Ud-Din, M., Ivannikova, E., Khan, S.A., *et al*. (2019) Federated Collaborative Filtering for Privacy-Preserving Personalized Recommendation System. arXiv: 1901.09888.

[5] Koren, Y., Bell, R. and Volinsky, C. (2009) Matrix Factorization Techniques for Recommender Systems. *Computer*, **42**, 30-37. https://doi.org/10.1109/mc.2009.263

[6] Rendle, S. (2012) Factorization Machines with libFM. *ACM Transactions on Intelligent Systems and Technology*, **3**, 1-22. https://doi.org/10.1145/2168752.2168771

[7] Chai, D., Wang, L., Chen, K. and Yang, Q. (2021) Secure Federated Matrix Factorization. *IEEE Intelligent Systems*, **36**, 11-20. https://doi.org/10.1109/mis.2020.3014880

[8] Qi, T., Wu, F., Wu, C., Huang, Y. and Xie, X. (2020) Privacy-Preserving News Recommendation Model Learning. *Findings of the Association for Computational Linguistics*: *EMNLP* 2020, Online, November 2020, 1423-1432. https://doi.org/10.18653/v1/2020.findings-emnlp.128

[9] Flanagan, A., Oyomno, W., Grigorievskiy, A., Tan, K.E., Khan, S.A. and Ammad-Ud-Din, M. (2021) Federated Multi-View Matrix Factorization for Personalized Recommendations. *Machine Learning and Knowledge Discovery in Databases*: *European Conference*, *ECML PKDD* 2020, Ghent, 14-18 September 2020, 324-347. https://doi.org/10.1007/978-3-030-67661-2_20

[10] McSherry, F. and Mironov, I. (2009) Differentially Private Recommender Systems: Building Privacy into the Netflix Prize Contenders. *Proceedings of the* 15*th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Paris, 28 June-1 July 2009, 627-636. https://doi.org/10.1145/1557019.1557090

[11] Liu, Z., Wang, Y. and Smola, A. (2015) Fast Differentially Private Matrix Factorization. *Proceedings of the* 9*th ACM Conference on Recommender Systems*, Vienna, 16-20 September 2015, 171-178. https://doi.org/10.1145/2792838.2800191

[12] Yang, J., Li, X., Sun, Z. and Zhang, J. (2019) A Differential Privacy Framework for Collaborative Filtering. *Mathematical Problems in Engineering*, **2019**, Article ID: 1460234. https://doi.org/10.1155/2019/1460234

[13] Erkin, Z., Veugen, T., Toft, T. and Lagendijk, R.L. (2012) Generating Private Recommendations Efficiently Using Homomorphic Encryption and Data Packing. *IEEE Transactions on Information Forensics and Security*, **7**, 1053-1066. https://doi.org/10.1109/tifs.2012.2190726

[14] Kim, S., Kim, J., Koo, D., Kim, Y., Yoon, H. and Shin, J. (2016). Efficient Privacy-Preserving Matrix Factorization via Fully Homomorphic Encryption. *Proceedings of the* 11*th ACM on Asia Conference on Computer and Communications Security*, Xi'an, 30 May-3 June 2016, 617-628. https://doi.org/10.1145/2897845.2897875

[15] 张永棠. 基于代换加密的隐私保护协同过滤推荐算法[J]. 新疆大学学报(自然科学版), 2017, 34(4): 446-451.

[16] Bottou, L. (2010) Large-Scale Machine Learning with Stochastic Gradient Descent. *Proceedings of COMPSTAT*'2010: 19*th International Conference on Computational Statistics*, Paris, 22-27 August 2010, 177-186. https://doi.org/10.1007/978-3-7908-2604-3_16

[17] Acar, A., Aksu, H., Uluagac, A.S. and Conti, M. (2018) A Survey on Homomorphic Encryption Schemes: Theory and Implementation. *ACM Computing Surveys*, **51**, 1-35. https://doi.org/10.1145/3214303

[18] Albrecht, M., Chase, M., Chen, H., Ding, J., Goldwasser, S., Gorbunov, S., *et al*. (2021) Homomorphic Encryption Standard. In: Lauter, K., Dai, W. and Laine, K., Eds., *Protecting Privacy through Homomorphic Encryption*, Springer International Publishing, 31-62. https://doi.org/10.1007/978-3-030-77287-1_2

[19] Paillier, P. (1999) Public-Key Cryptosystems Based on Composite Degree Residuosity Classes. In: *International Conference on the Theory and Applications of Cryptographic Techniques*, Springer, 223-238. https://doi.org/10.1007/3-540-48910-x_16

[20] Koren, Y. (2008) Factorization Meets the Neighborhood: A Multifaceted Collaborative Filtering Model. *Proceedings of the* 14*th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Las Vegas, 24-27 August 2008, 426-434. https://doi.org/10.1145/1401890.1401944

[21] Behera, G. and Nain, N. (2021) Collaborative Recommender System (CRS) Using Optimized SGD-Als. *Advances in Computing and Data Sciences*: 5*th International Conference*, *ICACDS* 2021, Nashik, 23-24 April 2021, 627-637. https://doi.org/10.1007/978-3-030-81462-5_55

[22] Oded, G. (2009) Foundations of Cryptography: Volume 2, Basic Applications. Cambridge University Press.