E-Commerce Letters 电子商务评论, 2024, 13(3), 5659-5671
Published Online August 2024 in Hans. https://www.hanspub.org/journal/ecl
https://doi.org/10.12677/ecl.2024.133696

Hans汉斯

一种新的协同过滤算法在电子商务产品推荐中的应用

孙佑烻，马家君*

贵州大学大数据与信息工程学院，贵州 贵阳

收稿日期：2024年4月24日；录用日期：2024年5月11日；发布日期：2024年8月6日

摘 要

随着互联网科技日新月异的发展，网络中存储的数据总量正以前所未有的速度激增。在这种背景下，如何从海量数据资源中精准高效地提炼出我们所需要的特定信息，已然成为当下亟待解决的重要议题。在这项研究中，我们提出了一种新的协同过滤算法来生成电子商务产品中的推荐。这项研究有两个主要创新之处。首先，我们提出了一种嵌入时间行为信息的机制，以找到一个邻居集，其中每个邻居对当前用户或项目有非常重要的影响。其次，在概率矩阵分解中引入邻居集，提出了一种新的协同过滤算法。我们将所提出的方法与实际数据集上的几种最先进的替代方法进行了比较。实验结果表明，本文提出的方法优于现有的方法。

关键词

协同滤波算法，概率矩阵，产品推荐

Application of a New Collaborative Filtering Algorithm in E-Commerce Product Recommendation

Youxin Sun, Jiajun Ma*

College of Big Data and Information Engineering, Guizhou University, Guiyang Guizhou

Received: Apr. 24th, 2024; accepted: May 11th, 2024; published: Aug. 6th, 2024

Abstract

With the relentless and groundbreaking advancements in internet technology, the aggregate vol-

*通讯作者。

文章引用: 孙佑烻, 马家君. 一种新的协同过滤算法在电子商务产品推荐中的应用[J]. 电子商务评论, 2024, 13(3): 5659-5671. DOI: 10.12677/ecl.2024.133696

ume of data stored within network architectures is expanding at an unparalleled velocity. In this context, the task of precisely and efficiently extracting the targeted information from the deluge of data resources has emerged as a crucial and pressing issue demanding immediate attention and resolution in contemporary times. In this study, we propose a new collaborative filtering algorithm to generate recommendations in e-commerce products. There are two main innovations in this study. First, we propose a mechanism for embedding temporal behavior information to find a set of neighbors, where each neighbor has a very important influence on the current user or project. Secondly, a new collaborative filtering algorithm is proposed by introducing neighbor sets into probability matrix factorization. We compared the proposed method with several state-of-the-art alternatives on actual datasets. Experimental results show that the proposed method is superior to the existing methods.

Keywords

Collaborative Filtering Algorithms, Probability Matrix, Product Recommendations

1. 引言

面对互联网技术的快速发展所带来的指数级增长，网络内累积的数据量大幅飙升。从海量的数据中高效地提取精确而具体的信息已经成为一种日益迫切的需求。虽然传统的搜索引擎在某种程度上已经证明了满足用户检索需求的能力，但它们经常呈现同质化的排名，而不考虑个人用户的偏好。

因此，积极提供符合不同用户兴趣的个性化服务是一项挑战。因此，我们强烈需要一个推荐系统[1] [2]，推荐系统的核心功能在于汇聚并深度解析用户所输入的各项数据，以揭示其内在的兴趣偏好与行为模式，并在此基础上为每位用户精准推送个性化的信息服务。鉴于其能有效破解信息过剩难题，推荐系统已成功引起了学术研究领域与产业界的广泛瞩目与高度重视。

在推荐系统的所有各类实现策略中，协同过滤[3] [4]占据着广泛应用的主导地位。基本理念主要包括两个核心步骤：第一步，依据用户的实际行为记录解析并构建用户兴趣模型，并在整个用户群体中探寻与目标用户兴趣相契合的相似用户集合；第二步，对这些邻居的所有评价进行汇总，最终为目标用户生成兴趣决策，从而推荐最相关的项目。鉴于协同过滤算法对被推荐项目没有特别先验条件限制，其擅长处理那些难以通过文本属性精确刻画的领域，如音乐和电影。正因如此，协同过滤技术被广泛应用在推荐系统领域内，并展现出卓越的表现和重大的商业价值，特别是在电子商务场景中体现尤为突出。举例来说，根据 VentureBeat 发布的数据，亚马逊的推荐系统为他们提供了至少 35%的商品销售[5]。

虽然协同过滤推荐算法有许多成功的应用，但仍有几个重要的问题需要解决。一个重要的问题是，传统的协同过滤往往忽略了用户或项目之间的结构关系。在下面，我们只使用用户来表示“用户(项)”。有效地利用用户之间的关系，可以丰富单个用户的信息，从而更准确地识别用户的兴趣。基于这一概念，人们开发了许多充分考虑用户关系的协同过滤算法。他们促成了许多有价值的成果。获取用户之间的关系并将其量化的过程对协同过滤算法的有效性有重要影响。

目前解决方案主要分为两种，一种是利用显式社会网络中存在的关系，另一种则是通过对隐性标签信息的分析来推算用户间的相似性，从而构建用户间的关系网络。但在实际应用中，很难获得关于社会关系或标签的充分信息。而且，以上的解决方案都是基于模糊的假设，即用户之间的交互是无向的，这

在现实中是不合理的。例如，如果用户 A 非常欣赏用户 B，则用户 B 的行为对用户 A 的影响很大，而用户 A 对用户 B 的影响并不显著，甚至可以忽略不计。

为了解决上述问题，本文提出了运用时间消费行为建立最近邻模型的方法。通过构建以时间为维度的消费网络，将有利于深入洞察并识别用户间的交互。贡献可以总结如下：

- 提出的建模策略仅需依赖用户的消费时间数据，无需涉及用户标签、社会关系等复杂的信息元素。这种策略所实施的计算效应具有直接性特点，能够更为精确地揭示用户间的互动联系，并在邻居用户集合中有效辨识出影响力最大的成员。
- 在建模策略的基础上，提出了一种基于概率矩阵分解，利用邻居集进行协同过滤推荐的 TemporalMF 推荐算法。
- 在豆瓣的大规模推荐数据集上，我们进行了深入广泛的实验研究。实验结果显示，TemporalMF 算法在结合社交网络特征和标签信息的基础上，相较于传统的推荐算法，能够更有效地预估用户的得分，从而提高推荐的准确性。

2. 相关工作

2.1. 传统的协同过滤算法

协同过滤利用类似用户的行为(分数、点击等)来预测目标用户对特定项目的兴趣。然后，它会根据预测的兴趣做出相应的推荐。目前流行的协同过滤推荐算法主要分为基于邻域的推荐算法和基于模型的推荐算法。

采用基于邻域的协同过滤方法，首要步骤是依据用户的历史行为数据来量化评估用户间的相似性程度，随后，通过选取与目标用户具有较高相似性的邻居用户群体，借鉴这些邻居用户对未被目标用户评价过的其他物品的评分信息，以此作为依据来推测目标用户对特定物品可能存在的喜好程度。将用户的偏好程度作为推荐阈值。目前，基于邻居的协同过滤主要包括基于用户的过滤和基于项目的过滤。

相较于依赖于领域的协同过滤技术，基于模型的协同过滤方法更注重于构建并训练一个预测模型，该模型充分利用用户的评分数据作为输入信息。通过这一模型，能够对尚未获取评分的项目或商品进行有效的预测分析，从而揭示用户潜在的喜好倾向[6]。当前，这一领域的代表性技术主要包括聚类模型[7]、概率相关模型[8] [9]、潜在因素模型[10]-[12]和贝叶斯层次模型[13] [14]。近来，针对大数据处理的需求，Mnih 等[15]提出了一种概率矩阵分解(PMF)算法，该算法使用低维近似矩阵分解生成推荐。一般来说，它假设每个用户的兴趣偏好仅受到有限几个关键因素的影响。其核心机制在于将用户和物品映射到一个更低维度的特征空间内，而这一映射过程所依据的学习参数，则来自于用户实际提供的评价反馈数据。通过运用这些参数得到的用户特征向量，可以有效重建原始的评价矩阵。经实验验证，PMF 在处理大规模数据方面具有高效性和良好的预测准确性。为了克服在参数设置带来的负面影响，Salakhutdinov 等[16]提出了贝叶斯概率矩阵分解(BPMF)。其实验结果显示性能优于 PMF。

Lawrence 等[17]从理论上证实了 PMF 与概率主成分分析之间的准确性。并基于此提出了一种非线性 PMF，其结合高斯过程对 PMF 进行非线性扩展，旨在获得更好的性能。尽管传统的协同过滤算法在推荐效果上常能达到令人满意的效果，但其局限存在于仅依赖于评分数据，并未能充分利用与推荐紧密相连的时间信息和关系信息。基于此，研究者们提出了许多基于时间信息和关系信息的推荐算法。

2.2. 基于时间信息的推荐推荐算法

该模型具备学习数据动态变化的能力并以此优化推荐。Koren [18]提出了 TimeSVD++，该算法通过

将时间维度信息融入用户特征向量，有效地解决了兴趣随时间推移可能出现的漂移问题，实验结果表明了 TimeSVD++的优势。Xiong 等[19]以时间信息为第三维，利用张量分解对动态变化进行建模。Chen 等[20]动态地将用户分配到未使用的集群中，并基于进化共聚类提出进一步的推荐。Li 等[21]指出，在特定时间段内，用户的兴趣焦点往往会集中在少数几个特定领域，并基于此，提出了一种跨领域协同过滤框架。实验结果显示，该框架不仅能有效地捕捉并推荐符合用户兴趣迁移趋势的内容，还能实时追踪用户兴趣的动态变迁。Ren 等[22]观察到，当前推荐系统普遍忽视了用户的偏好模式及其动态变化特性。为此，他们将用户的偏好模式进行稀疏矩阵表示，并运用相应的子空间，分步骤地捕捉和构建个性化及全局范围内的用户偏好模式。

与以往的研究不同，本研究使用时间信息来构建用户或项目之间的结构关系。基于结构关系，计算相似度并将其整合到概率矩阵分解中，生成新的推荐框架。

2.3. 基于关系信息的推荐算法

传统协同过滤算法常常假定用户和项目之间不存在相互联系，故而在推荐过程中忽视了两者间可能存在的结构关联性。针对这一局限，将用户或项目间的关系网络纳入现有协同过滤框架，通过强化单一用户信息的多元性，提升推荐结果的精准度。

在基于关系挖掘的协同过滤技术中，一项至关重要的环节即是萃取用户之间的关联性。当前获取用户关系的途径主要分为显式和隐式两种。在[23]-[25]中，作者借助显式社交网络关系捕获了用户间的相互关联。Jamali 等[26]提出了一种基于用户信任关系网络的随机游走模型，该模型通过在网络中执行随机游走以查找到相似项目，并借此扩充目标项目预测评分的数据来源，从而减轻了数据稀疏性所带来的不利影响。然而，在实际应用场景中，搜集充足且有效的网络关系是一项颇具挑战的任务。Zhou 等[27]利用标签信息得到推荐的隐式关系矩阵。

鉴于前文所述的显式和隐式关系建模手段时常受限于大规模相关数据采集，本研究着重利用用户消费的时间序列信息挖掘隐含关联性，并将用户或物品关系集成到矩阵分解模型中。在此基础上，设计了一种基于时间序列行为的协同过滤推荐算法(称为 TemporalMF)。

3. 定义问题和分解概率矩阵

假设我们有一个推荐系统，它包含 N_u 个用户，表示为 $\Phi=\left\{\phi_1,\phi_2,\cdots,\phi_{N_u}\right\}$，包含 N_p 项，记为 $\psi=\left\{\varphi_1,\varphi_2,\cdots,\varphi_{N_P}\right\}$，用户－物品评分矩阵表示为 $\Re=\left[\Re_{i,j}\right]_{N_u\times N_p}$，其中每个元素 $\left[\Re_{i,j}\right]$ 表示第 i 个用户对第 j 个物品的评分。协同过滤算法使用概率矩阵分解模型学习用户或物品的特征向量，然后根据该特征向量预测未知的评级[28]。

设 $\Theta\in\mathbb{R}^{K\times N_u}$ 和 $\Xi\in\mathbb{R}^{K\times N_u}$ 分别为用户和项目的特征矩阵。根据上述定义，现有评级数据的条件概率定义如下：

$$p\left(\Re\mid\Theta,\Xi,\sigma_\Re^2\right)=\prod_{i=1}^{N_u}\prod_{j=1}^{N_p}\left[N\left(\Re_{i,j}\mid g\left(\Theta_i^T\Xi_j\right),\sigma_\Re^2\right)\right]^{S_{i,j}^\Re} \tag{1}$$

其中 $N\left(x\mid\mu,\sigma^2\right)$ 表示 μ 和 σ^2 的正态分布。$S_{i,j}^\Re$ 是一个 0~1 函数，其中如果第 i 个用户评价第 j 项，返回 1；否则，返回 0。$g(x)$ 是将 x 映射到[0, 1]的函数。

$$g(x)=1\Big/\left(1+\mathrm{e}^{-x}\right) \tag{2}$$

为了避免过拟合，假设用户和项目的特征向量都遵循高斯先验分布，$u=0$。

$$p\left(\Theta \mid \sigma_{\Theta}^{2}\right)=\prod_{i=1}^{N_u} N\left(\Theta_i \mid 0, \sigma_{\Theta}^{2} \mathrm{I}\right) \tag{3}$$

$$p\left(\Xi \mid \sigma_{\Xi}^{2}\right)=\prod_{i=1}^{N_u} N\left(\Xi_i \mid 0, \sigma_{\Xi}^{2} \mathrm{I}\right) \tag{4}$$

根据贝叶斯推理理论， Θ 和 Ξ 的后验概率可表示为：

$$\begin{aligned} & p\left(\Theta, \Xi \mid \Re, \sigma_{\Re}^{2}, \sigma_{\Theta}^{2}, \sigma_{\Xi}^{2}\right) \propto p\left(\Re \mid \Theta, \Xi, \sigma_{\Re}^{2}\right) p\left(\Theta \mid \sigma_{\Theta}^{2}\right) p\left(\Xi \mid \sigma_{\Theta}^{2}\right) \\ & =\prod_{i=1}^{N_u} \prod_{j=1}^{N_p}\left[N\left(\Re_{i, j} \mid g\left(\Theta_i^{T} \Xi_j\right), \sigma_{\Re}^{2}\right)\right]^{S_{i, j}^{\Re}} \times \prod_{i=1}^{N_p} N\left(\Theta_i \mid 0, \sigma_{\Theta}^{2} \mathrm{I}\right) \times \prod_{j=1}^{N_p} N\left(\Xi_j \mid 0, \sigma_{\Xi}^{2} \mathrm{I}\right) \end{aligned} \tag{5}$$

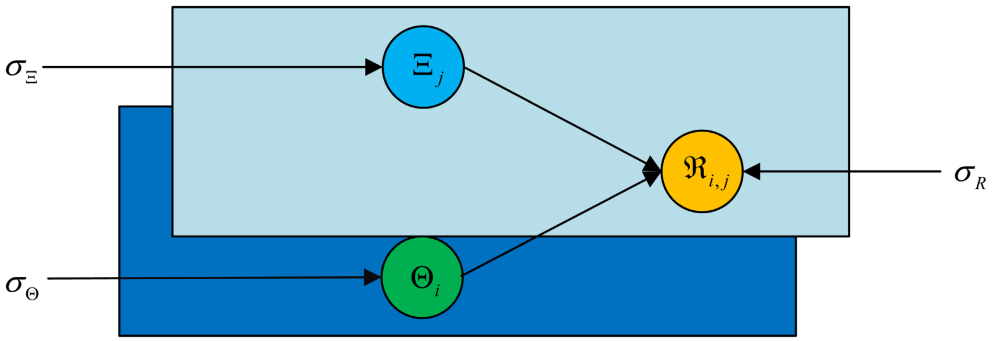

Figure 1. Probability matrix decomposition diagram
图 1. 概率矩阵分解图

图 1 给出了概率矩阵分解图。根据公式(3)，有了用户项评分矩阵，就很容易学习到相应的特征向量。

4. 提出的方法

4.1. 基于时间序列行为的协同过滤推荐算法

在基于关系的矩阵分解中，核心步骤之一是获取用户或项目关系。传统的协同过滤忽略了消费时间信息。但是，时间序列信息可以提供一些隐藏的模式，这些模式可用于挖掘用途和项之间的关系。为了发现这些潜在的关系，我们首先引入一个基于时间序列的用户消费网络，如图 2 所示。

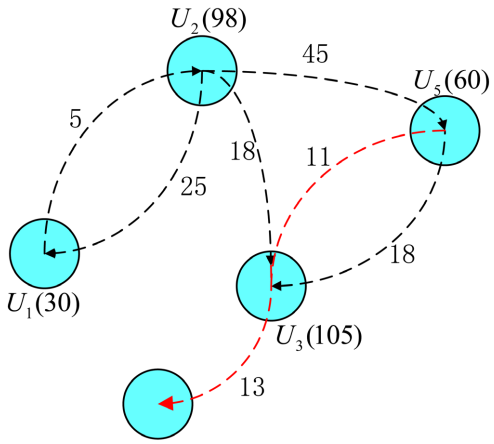

Figure 2. User consumption network diagram
图 2. 用户消费网络图

我们使用 $G=\{\Phi,\Delta,\Lambda\}$ 来表示图 2 所示的网络，其中 Φ 为用户集；Δ 表示边集；Λ 表示边权矩阵。“()”中的数字表示用户消费的物品数量。在给定的时间段内，如果用户 i 和 j 先后消费相同的物品，则边集 $\Delta_{i,j}$ 的权重 $\Lambda_{i,j}$ 增加 1。通过迭代所有乘积，我们可以更新边权矩阵 Λ。因此，我们可以定义影响关系权重如下：

$$\prod_{i,j}=\frac{\Lambda_{i,j}}{\vee\left(\Phi_i,\Phi_j\right)} \tag{6}$$

其中 $\vee(A,B)$ 表示集合 A 和集合 B 中同时包含的相同元素的个数。$\prod_{i,j}$ 可以解释为第 i 个用户对第 j 个用户的影响。例如，如果用户 1 和 2 消耗的物品数量为 100，则 1 对 2 的影响为 $\prod_{1,2}=5/100=0.05$。反之，2 对 1 的影响为 $\prod_{2,1}=25/100=0.25$。因此，我们观察到影响是有方向性的，这比无方向性的情况更合理。

类似地，基于时间序列的物品消费网络可以很容易地构建，图 3 给出了一个简单的例子。“()”中的数字现在表示消费该商品的用户数量；权重 $\Lambda_{i,j}$ 现在表示逐个消费商品 i 和 j 的用户数量。因此，影响关系权重可定义为：

$$\Omega_{i,j}=\frac{\Lambda_{i,j}}{\vee\left(\psi_i,\psi_j\right)} \tag{7}$$

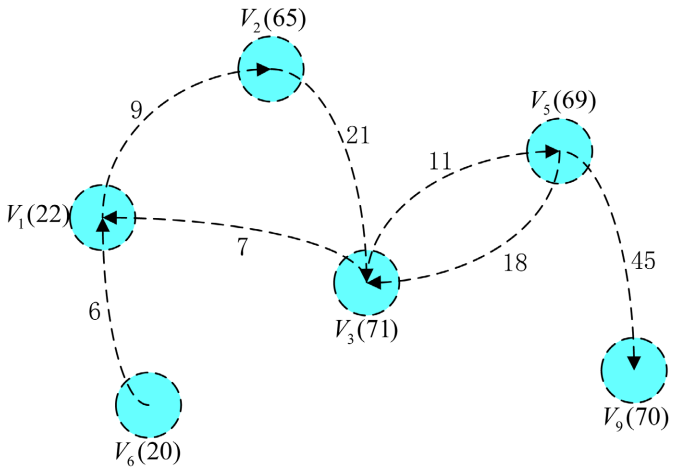

Figure 3. Project consumption network diagram
图 3. 项目消费网络图

4.2. 矩阵分解

在确定了用户或项目间相互作用关系并成功识别出最相近邻集合之后，将这一邻近关系引入到矩阵分解模型中。且最近邻的会影响到用户或项目的特征向量。换句话说，相似的用户或项目应该具有相似的特征向量。

$$\tilde{\Theta}_i=\sum_{v\in neg(i)}\prod_{v,i}\Theta_i \tag{8}$$

$$\tilde{\Xi}_j=\sum_{t\in neg(i)}\prod_{t,i}\Xi_t \tag{9}$$

其中 $neg(i)$ 表示 i 的邻居集(用户或项目)。$\tilde{\Theta}_i$ 和 $\tilde{\Xi}_j$ 分别表示近似特征向量。与[6] [8]不同的是，我们的算法综合考虑了用户或物品的内在特征和关系的双重因素。每个用户或项目的特征向量不仅遵循 $\mu=0$ 的

高斯先验分布以避免过拟合，而且需要与用户或项目关系特征向量相似。由于考虑了时间序列信息，影响关系更清晰，更符合实际情况。

$$p\left(\Xi \mid \Pi, \sigma_{\Theta}^{2}, \sigma_{\Pi}^{2}\right) \propto p\left(\Theta \mid \sigma_{\Theta}^{2}\right) \times p\left(\Theta \mid \Pi, \sigma_{\Pi}^{2}\right)$$
$$= \prod_{i=1}^{N_u} N\left(\Theta_i \mid 0, \sigma_{\Theta}^{2} \mathrm{I}\right) \times \prod_{i=1}^{N_u} N\left(\Theta_i \mid \sum_{v \in neg(i)} \prod_{v,i} \Theta_{v,\sigma_{\Pi}^{2} \mathrm{I}}\right) \tag{10}$$

$$p\left(\Xi \mid \Omega, \sigma_{\Xi}^{2}, \sigma_{\Omega}^{2}\right) \propto p\left(\Omega \mid \sigma_{\Xi}^{2}\right) \times p\left(\Xi \mid \Omega, \sigma_{\Omega}^{2}\right)$$
$$= \prod_{j=1}^{N_p} N\left(\Xi_j \mid 0, \sigma_{\Xi}^{2} \mathrm{I}\right) \times \prod_{j=1}^{N_p} N\left(\Xi_j \mid \sum_{t \in neg(j)} \Omega_{t,j} \Xi_{j,\sigma_{\Omega}^{2} \mathrm{I}}\right) \tag{11}$$

利用贝叶斯理论进行推到，该模型的概率图如图 4 所示。

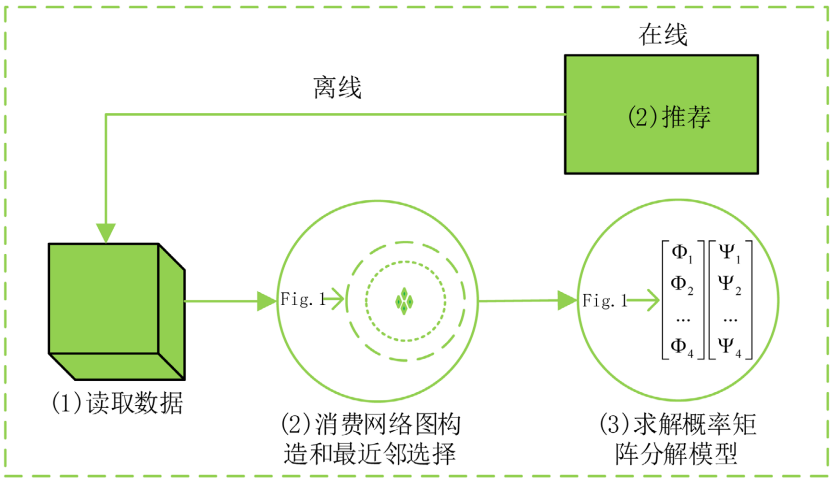

Figure 4. temporalMF recommendation algorithm framework
图 4. temporalMF 推荐算法框架图

4.3. 推荐算法步骤及计算复杂度分析

推荐算法分为四个步骤：

步骤一：数据读取：输入包含用户的评分信息和评分时间信息。

步骤二：邻居集构建：根据输入，构建用户消费网络和物品消费网络。然后，构造最近邻集。

步骤三：概率矩阵分解：我们将邻居集应用到概率矩阵分解模型中，学习用户和物品的特征向量。

步骤四：评分矩阵重构：基于用户和物品的特征向量，重构评分矩阵，得到相应的用户推荐。

在我们提出的算法中，消耗大量 CPU 秒的步骤来自两个方面。

一是消费网络建设和关系挖掘。二是根据得到的影响关系提出建议。我们以用户视角的网络为例，若每一件商品平均被 $\hat{r}$ 个用户购买，首先按照用户给商品打分的时间顺序对评分信息进行排列，随后分析这些用户所形成的网络连接权重是否有所增加。依照这个流程，构建单个用户消费网络所需的计算复杂度大致为 $\mathrm{O}\left(\hat{r}^2\right)$，构建涵盖所有用户的用户消费网络，其总体时间复杂度为 $\mathrm{O}\left(N_u \hat{r}^2\right)$。同样，假设每个用户平均消费的商品数量为 $\hat{t}$，则构建该商品的用户消费网络的时间复杂度为 $\mathrm{O}\left(N_p \hat{t}^2\right)$。因此，构建用户消费网络的时间复杂度为 $\mathrm{O}\left(N_u \hat{r}^2 + N_p \hat{t}^2\right)$。在建立这个网络之后，我们利用这个网络来挖掘影响关系。由于每个用户平均消耗 $\hat{t}$ 个物品，因此每个用户在网络中有 $\hat{t}$ 条边。

因此，依次计算每个用户对其他用户的影响并排序的时间复杂度为$O\left(N_u\hat{t}^2\right)$。同样，在基于物品的消费网络中，该时间复杂度为$O\left(N_p\hat{r}^2\right)$。根据上述分析，第一个方面的总时间复杂度为$O\left(\left(N_p+N_u\right)\times\left(\hat{r}^2+\hat{t}^2\right)\right)$。

根据[6]的分析，梯度计算的总时间复杂度为$O\left(N_u\hat{t}K+N_ul^2K+N_p\hat{r}K+N_pl^2K\right)$，$l$表示影响某个用户的邻居数。值得注意的是，构建网络的过程仅需顺序遍历评级信息，无需涉及迭代操作。与此同时，考虑到$\hat{r}$和$\hat{t}$数值相对较小，故模型的整体时间相对较低，因此适合应用于大数据的处理任务。

5. 实验结果

5.1. 数据集

为全面探讨各类信息及关系因素对推荐效果产生的差异化影响，构建的实验数据集应整合包括但不限于用户评分记录、项目标签信息以及用户社交网络链接等多元数据。鉴于此需求，在本次研究中，我们选定豆瓣平台作为数据来源，旨在针对性地获取并整合上述各类相关信息，以此为基础搭建起用于实证分析的实验数据集。豆瓣中的每位用户均可对各类书籍、音乐或电影作品进行 1 到 5 分的评分。此外，豆瓣还借鉴了 Facebook 的社交关系功能，允许用户通过电子邮箱地址查找并连接好友。因此，豆瓣所提供的丰富多样的数据资源恰好符合我们进行实验研究所需的条件。

本研究使用的数据集见表 1，其中包含两组数据。

一组数据专注于用户对图书的评价反馈，还包含了用户的社交网络关系数据以及相关的标签信息。另一组数据则围绕用户对电影的评分活动展开，同样提供了丰富的评分数据以及其他与之对应的各类相关信息。

Table 1. Douban data
表 1. 豆瓣数据

信息	数据类型	
	书籍	电影
用户	23,944	9601
项目	219,725	44,779
标签信息	74,095	50,530
评级记录	1,642,111	1,960,682
社会关系	588,269	91,945

5.2. 评估标准

本研究采用 RMSE [29]作为评价标准：

$$\mathrm{RMSE}=\frac{\sqrt{\sum_{i=1}^{C}\left(p_i-r_i\right)^2}}{C} \tag{12}$$

其中$\{p_1,p_2,\cdots,p_C\}$表示预测向量，$\{r_1,r_2,\cdots,r_C\}$表示真值向量。

5.3. 实验结果与分析

在本节中，我们选择了四种方法作为比较算法，它们是 PMF、BPTF、SocialMF 和 TagMF。从特征

向量不同维度下的性能、计算复杂度和鲁棒性分析三个方面报道我们的实验。

首先，我们将特征向量 K 的维度分别设置为 5、10 和 20。

表 2 显示了不同 K 下所有算法的均方根误差。我们观察到，我们提出的算法在每个 K 下的性能都是最好的。

Table 2. Root mean square error of all algorithms under different K values
表 2. 不同 K 值下所有算法的均方根误差

算法	K = 5		K = 10		K = 20	
	书籍	电影	书籍	电影	书籍	电影
PMF	0.7511	0.7358	0.7465	0.7327	0.7435	0.7311
BPTF	0.7317	0.7279	0.7267	0.7231	0.7242	0.7228
SocialMF	0.7339	0.7309	0.7307	0.7269	0.7289	0.7245
TagMF	0.7298	0.7267	0.7240	0.7235	0.7219	0.7218
TemporalMF	0.7294	0.7251	0.7238	0.7229	0.7217	0.7208

1) 当 K 值不断提升时，该算法的准确度呈现出逐渐增强的趋势。然而，必须留意的是，随着 K 值的增长，模型所对应的时间复杂度也将相应有所增加。

2) 与 PMF 相比，BPTF、TagMF、SocialMF 和本文算法的性能都有显著提高，表明时间信息与用户(物品)之间的关系对提高传统协同过滤算法的准确性有更大的作用。

3) 相较于 SocialMF，BPTF 在性能表现上有所提升，这一改进部分归因于 SocialMF 中所反映出的社会关系的稀疏性问题。与 TagMF 和 TemporalMF 相比，BPTF 的性能结果则稍显逊色。

4) 与 TagMF 和我们的方法相比，SocialMF 的表现略差，这主要是因为 SocialMF 没有考虑项目之间的关系。此外，SocialMF 不考虑朋友之间关系的方向。

5) 相较于 TagMF，本研究所提出的算法在性能上显著超越了前者，这一对比有力地证实了本文引入的影响关系对提升算法准确性的积极作用。另外，鉴于本文算法仅需依赖最基本的时间信息，从信息获取的简易性及适用范围的广泛性角度来看，该算法均展现出一定的优越性。

实验数据显示，相较于传统的概率矩阵分解算法，本文所提出的算法在性能表现上更为出色，有力验证了文中引入的影响关系概念在实践中的合理性与实效性。尽管改进效果并非大幅度飞跃，但在真实世界推荐系统环境中，SocialMF 和 TagMF 所依赖的社会关系数据或标签信息往往难以全面获取或极度稀疏，而本文算法只需利用易于获得的时间信息即可运作，体现出独特的优势。因此，该算法可以广泛应用于许多实际应用中。

表 3 给出了在 Intel(R) Core(TM) i5-12600KF 3.70 GHz，Windows 10 系统，16 GB 内存环境下，所有算法每次迭代的 CPU 秒数。

我们可以看到，BPTF 算法所需的时间成本远超过其他算法。主要原因在于 BPTF 在采用马尔可夫蒙特卡罗方法进行训练的过程中耗费了大量的计算时间，从而导致了其较高的计算复杂度水平。由表 3 也可以看出，关系越多，计算复杂度越高。一般来说，TemporalMF 占用的时间是可以接受的。对于 PMF 和 BPTF，处理书籍的时间要小于处理电影的 CPU 秒数。相比之下，其他的算法在运行时间上则有所不同。究其原因，PMF 和 BPTF 这两种算法并未将邻居关系纳入考量范畴，因此它们的计算复杂度主要取决于训练集中评分数据条目的总量。与电影的评级数据相比，书籍的评级数据很少，因此书籍需要更少的时间。其余的算法将邻居关系考虑在内。由于图书数据集比电影数据集包含更多的用户和项目，因此

算法为图书消耗更多的时间。

Table 3. CPU time consumed per iteration of all algorithms (s)
表 3. 所有算法每次迭代所消耗的 CPU 时间(秒)

算法	书籍	电影
PMF	1.7	1.9
BPTF	15.2	18.6
SocialMF	2.9	2.7
TagMF	4.2	3.9
TemporalMF	4.4	4.0

在 TemporalMF 算法中，参数 λ 起到了度量用户或项目对其关联信息敏感度的作用。λ 的数值增大时，用户或项目对其关系信息的依赖程度也随之提升。为了简化本次实验的复杂性，我们设置 $\lambda_{\Omega} = \lambda_{\Pi} = \lambda$。$\lambda$ 的值分别设置为 0.1，0.5，1，5，10 和 20。另外，我们将 K 设为 5。不同 λ 的 TemporalMF 在书籍和电影上的表现如图 5 所示。

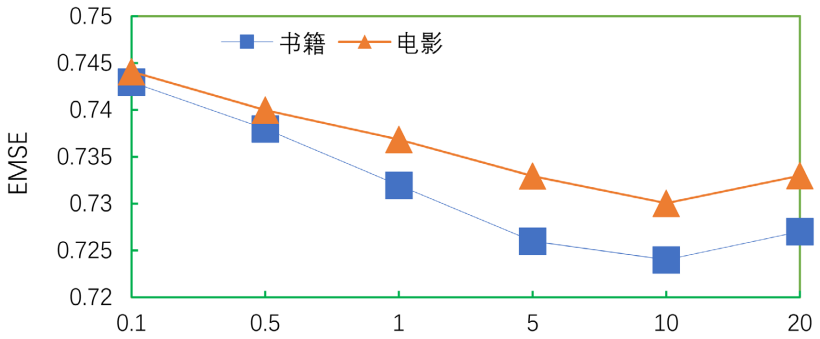

Figure 5. TemporalMF's performance in books and films under different λ
图 5. 不同 λ 下 TemporalMF 在书籍和电影的表现

从结果可以看出，参数 λ 对 TemporalMF 的影响较大。随着 λ 的增大，TemporalMF 的性能有所提高。这一现象有力验证了本文中通过时间信息所构建关系的可信度及其对改进算法性能的积极促进作用同时，观察到当 λ 值增至 20 时，其对 TemporalMF 的正面效应开始呈现递减趋势，这主要是由于 λ 值过大可能会引起算法对训练数据的过拟合现象，导致准确率下降。

为了比较算法在不同稀疏度条件下的效果，我们以用户评分数为基础，将训练集中的用户分为[0:10]，[10:100]，[100:500]和[500:inf]四组。图 6 展示了在两组数据中，各组用户占比的具体分布情况。

在对数据进行相应的划分后，我们使用训练集学习相应的模型，然后计算测试集上四组用户的 RMSE。我们观察到：

- 在数据极度稀疏的情况下(分数个数小于 10)，TemporalMF 的改进效果不明显，相较于那些融入额外信息的 SocialMF 和 TagMF 算法，其表现相对较弱。主要原因在于，过于稀疏的数据致使 TemporalMF 构建的关系网络也非常稀疏，这直接影响了其预测的准确性。BPTF 引入了参数训练方法，性能优于 TemporalMF。然而，TemporalMF 仍然比传统的 PMF 更好。
- 值得注意的是，即使在这样的情况下，TemporalMF 依然胜过传统的 PMF 算法。随着用户评分数据量的增加，TemporalMF 相较于 TagMF、SocialMF 以及 BPTF 表现出更佳的性能，这进一步证实了

本文所提出的基于时间序列影响关系的确能有效提升推荐系统的精确度。

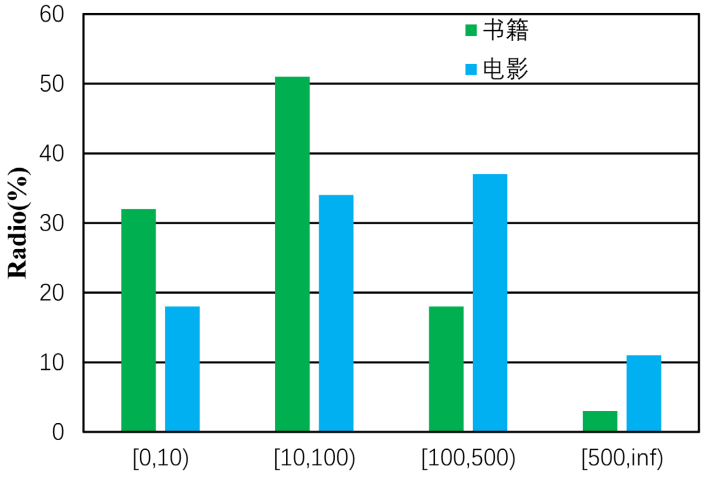

Figure 6. The proportion of users in each group in the two groups
图 6. 两组数据中每组用户占比

Table 4. Root-mean-square error of each algorithm with different user scores (books)
表 4. 不同用户分数下各算法的均方根误差(书籍)

算法	[0, 10)	[10, 100)	[100, 500)	[500, inf)
PMF	0.8411	0.7504	0.7321	0.7777
BPTF	0.8324	0.7402	0.7158	0.7320
SocialMF	0.8301	0.7423	0.7256	0.7541
TagMF	0.8227	0.7354	0.7148	0.7318
TemporalMF	0.8400	0.7258	0.7125	0.7320

Table 5. Root-mean-square error of each algorithm with different user scores (movie)
表 5. 不同用户分数下各算法的均方根误差(电影)

算法	[0, 10)	[10, 100)	[100, 500)	[500, inf)
PMF	0.8588	0.7501	0.7378	0.7458
BPTF	0.8574	0.7408	0.7305	0.7402
SocialMF	0.8361	0.7410	0.7304	0.7403
TagMF	0.8114	0.7408	0.7304	0.7406
TemporalMF	0.8362	0.7407	0.7302	0.7402

从表 4 和表 5 也可以看出，无论是使用 TemporalMF 还是其他算法，RMSE 结果都不会随着评分次数的增加而继续降低。当评分数达到 500 及以上时，算法的效果开始下降。这主要是因为当分数数量较少时，数据的稀疏性会算法的表现产生了消极影响，此时模型易陷入过拟合状态，尽管对训练数据的拟合程度较高，但在未知测试样本上的预测准确性却偏低。随着评分数量的不断增加，数据密度逐渐提高，稀疏性问题得到缓解，算法性能随之逐渐改善。积累了足够多的评分后，用户的兴趣偏好可能出现多样化转变，这使得模型难以捕捉到用户稳定的特征和偏好，从而影响模型的准确性。因此合理控制样本数

量对模型预测性能至关重要。从表 4 和表 5 可以观察到，对于本研究所使用的数据集而言，在用户评分样本数量位于(100~500)区间时，算法表现最优。

6. 结论

本研究利用时间信息建立用户(物品)消费网络。通过该网络，找到用户(项)潜在交互关系和最近邻集。进一步地，将此类动态关系信息巧妙融入到基于矩阵分解的协同过滤推荐算法体系中，从而有力提升了对用户评分预测的精确度。考虑到消费时间数据相比社交网络属性和标签信息更加易于取得，基于时间行为的协同过滤推荐方法在实际应用中表现出广泛的适应性和可行性。在对豆瓣推荐数据集进行的实证分析中，我们发现相较于传统的推荐算法，这种方法展现了一定的竞争优势。未来的研究工作中，我们将持续深化探究此方法在应对稀疏消费网络和用户与物品冷启动问题时所面对的挑战。此外，我们提出的关联性获取策略并不仅限于概率矩阵分解算法范畴，它同样适用于与其他类型的矩阵分解方法相结合。未来计划还包括进一步探索将此技术运用于电子商务产品推荐预测的优化提升。

参考文献

[1] Xu, X., Zhou, J.Y., Liu, Y., Xu, Z.Z. and Zha, X.W. (2015) Taxi-RS: Taxi-Hunting Recommendation System Based on Taxi GPS Data. *IEEE Transactions on Intelligent Transportation Systems*, **16**, 1716-1727. https://doi.org/10.1109/tits.2014.2371815

[2] Baral, R., Iyengar, S.S., Zhu, X., Li, T. and Sniatala, P. (2019) Hirecs: A Hierarchical Contextual Location Recommendation System. *IEEE Transactions on Computational Social Systems*, **6**, 1020-1037. https://doi.org/10.1109/tcss.2019.2938239

[3] Yang, B., Lei, Y., Liu, J. and Li, W. (2017) Social Collaborative Filtering by Trust. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **39**, 1633-1647. https://doi.org/10.1109/tpami.2016.2605085

[4] Hu, Y., Peng, Q., Hu, X. and Yang, R. (2015) Time Aware and Data Sparsity Tolerant Web Service Recommendation Based on Improved Collaborative Filtering. *IEEE Transactions on Services Computing*, **8**, 782-794. https://doi.org/10.1109/tsc.2014.2381611

[5] Liu, J., Zhou, T., Wang, B. (2009) Research Progress of Personalized Recommendation System. *Progress in Natural Science*, **19**, 1-15.

[6] Yang, H., Ling, G., Su, Y., Lyu, M.R. and King, I. (2015) Boosting Response Aware Model-Based Collaborative Filtering. *IEEE Transactions on Knowledge and Data Engineering*, **27**, 2064-2077. https://doi.org/10.1109/tkde.2015.2405556

[7] Zhang, Y., Chung, F. and Wang, S. (2020) Clustering by Transmission Learning from Data Density to Label Manifold with Statistical Diffusion. *Knowledge-Based Systems*, **193**, Article 105330. https://doi.org/10.1016/j.knosys.2019.105330

[8] John, D.J., Fetrow, J.S. and Norris, J.L. (2011) Continuous Cotemporal Probabilistic Modeling of Systems Biology Networks from Sparse Data. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, **8**, 1208-1222. https://doi.org/10.1109/tcbb.2010.95

[9] He, Z., Wu, J. and Li, T. (2015) Label Correlation Mixture Model: A Supervised Generative Approach to Multilabel Spoken Document Categorization. *IEEE Transactions on Emerging Topics in Computing*, **3**, 235-245. https://doi.org/10.1109/tetc.2014.2377559

[10] Ge, Z. (2016) Supervised Latent Factor Analysis for Process Data Regression Modeling and Soft Sensor Application. *IEEE Transactions on Control Systems Technology*, **24**, 1004-1011. https://doi.org/10.1109/tcst.2015.2473817

[11] Lopez-Lopera, A.F. and Alvarez, M.A. (2019) Switched Latent Force Models for Reverse-Engineering Transcriptional Regulation in Gene Expression Data. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, **16**, 322-335. https://doi.org/10.1109/tcbb.2017.2764908

[12] Luo, X., Zhou, M., Xia, Y., Zhu, Q., Ammari, A.C. and Alabdulwahab, A. (2016) Generating Highly Accurate Predictions for Missing QoS Data via Aggregating Nonnegative Latent Factor Models. *IEEE Transactions on Neural Networks and Learning Systems*, **27**, 524-537. https://doi.org/10.1109/tnnls.2015.2412037

[13] Kaser, T., Klingler, S., Schwing, A.G. and Gross, M. (2017) Dynamic Bayesian Networks for Student Modeling. *IEEE Transactions on Learning Technologies*, **10**, 450-462. https://doi.org/10.1109/tlt.2017.2689017

[14] Chien, J. (2018) Bayesian Nonparametric Learning for Hierarchical and Sparse Topics. *IEEE/ACM Transactions on*

Audio, Speech, and Language Processing, **26**, 422-435. https://doi.org/10.1109/taslp.2017.2779862

[15] Mnih, A. and Salakhutdinov, R.R. (2007) Probabilistic Matrix Factorization. *Proceedings of the* 20*th International Conference on Neural Information Processing Systems*, Vancouver, 3-6 December 2007, 1257-1264.

[16] Salakhutdinov, R. and Mnih, A. (2008) Bayesian Probabilistic Matrix Factorization Using Markov Chain Monte Carlo. *Proceedings of the* 25*th International Conference on Machine Learning*, Helsinki, 5-9 July 2008, 880-887. https://doi.org/10.1145/1390156.1390267

[17] Lawrence, N.D. and Urtasun, R. (2009) Non-Linear Matrix Factorization with Gaussian Processes. *Proceedings of the* 26*th Annual International Conference on Machine Learning*, Montreal, 14-18 June 2009, 601-608. https://doi.org/10.1145/1553374.1553452

[18] Koren, Y. (2009) Collaborative Filtering with Temporal Dynamics. *Proceedings of the* 15*th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Paris, 28 June-1 July 2009, 447-456. https://doi.org/10.1145/1557019.1557072

[19] Xiong, L., Chen, X., Huang, T., Schneider, J. and Carbonell, J.G. (2010) Temporal Collaborative Filtering with Bayesian Probabilistic Tensor Factorization. *Proceedings of the* 2010 *SIAM International Conference on Data Mining*, Columbus, 29 April-1 May 2010, 211-222. https://doi.org/10.1137/1.9781611972801.19

[20] Chen, J., Wei, L., Uliji and Zhang, L. (2018) Dynamic Evolutionary Clustering Approach Based on Time Weight and Latent Attributes for Collaborative Filtering Recommendation. *Chaos, Solitons & Fractals*, **114**, 8-18. https://doi.org/10.1016/j.chaos.2018.06.011

[21] Li, B. (2011) Cross-Domain Collaborative Filtering: A Brief Survey. 2011 *IEEE* 23*rd International Conference on Tools with Artificial Intelligence*, Boca Raton, 7-9 November 2011, 1085-1086. https://doi.org/10.1109/ictai.2011.184

[22] Ren, Y., Zhu, T., Li, G., *et al*. (2013) Top-N Recommendations by Learning User Preference Dynamics. *Advances in Knowledge Discovery and Data Mining*, Gold Coast, 14-17 April 2013, 390-401. https://doi.org/10.1007/978-3-642-37456-2_33

[23] Ma, H., Yang, H., Lyu, M.R. and King, I. (2008) SoRec: Social Recommendation Using Probabilistic Matrix Factorization. *Proceedings of the* 17*th ACM Conference on Information and Knowledge Management*, Napa Valley, 26-30 October 2008, 931-940. https://doi.org/10.1145/1458082.1458205

[24] Ma, H., King, I. and Lyu, M.R. (2009) Learning to Recommend with Social Trust Ensemble. *Proceedings of the* 32*nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, Boston, 19-23 July 2009, 203-210. https://doi.org/10.1145/1571941.1571978

[25] Guo, L., Ma, J., Chen, Z. and Jiang, H. (2012) Learning to Recommend with Social Relation Ensemble. *Proceedings of the* 21*st ACM International Conference on Information and Knowledge Management*, Maui, 29 October-2 November 2012, 2599-2602. https://doi.org/10.1145/2396761.2398701

[26] Jamali, M. and Ester, M. (2009) TrustWalker: A Random Walk Model for Combining Trust-Based and Item-Based Recommendation. *Proceedings of the* 15*th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Paris, 28 June-1 July 2009, 397-406. https://doi.org/10.1145/1557019.1557067

[27] Zhou, T., Ma, H., Lyu, M. and King, I. (2010) UserRec: A User Recommendation Framework in Social Tagging Systems. *Proceedings of the AAAI Conference on Artificial Intelligence*, **24**, 1486-1491. https://doi.org/10.1609/aaai.v24i1.7524

[28] 孙光福, 吴乐, 刘淇, 等. 基于时序行为的协同过滤推荐算法[J]. 软件学报, 2013, 24(11): 2721-2733.

[29] Zhang, Y., Ishibuchi, H. and Wang, S. (2018) Deep Takagi-Sugeno-Kang Fuzzy Classifier with Shared Linguistic Fuzzy Rules. *IEEE Transactions on Fuzzy Systems*, **26**, 1535-1549. https://doi.org/10.1109/tfuzz.2017.2729507