

基于个性化推荐算法的电子商务系统实践研究

张 瑶¹, 刘 冲²

¹河北金融学院管理学院, 河北 保定

²河北金融学院金融科技院, 河北 保定

收稿日期: 2025年11月4日; 录用日期: 2025年12月5日; 发布日期: 2025年12月12日

摘 要

研究重点放在个性化推荐算法于电子商务范围的应用考察之上, 把主流推荐技术做系统归纳之后, 针对传统基于用户的协同过滤模型提出革新策略, 在融合用户静态属性以及动态兴趣衰减机制的基础上, 结合Redis缓存架构来改良计算效率, 凭借创建起模拟中小型电商平台环境的实验场景展开实证分析, 比较起经典方法, 改良过的算法不仅提高响应速度, 而且使推荐精准度得到改善, 于是推动点击转化率, 复购频率以及平台总交易额(GMV)等重要经营指标得到明显优化, 这便证实了所提方案在实际商业环境里具备可行性和一定价值。

关键词

电子商务, 个性化推荐, 协同过滤, 商品推荐

Practical Research on E-Commerce System Based on Personalized Recommendation Algorithm

Yao Zhang¹, Chong Liu²

¹School of Management, Hebei Finance University, Baoding Hebei

²School of Financial Technology, Hebei Finance University, Baoding Hebei

Received: November 4, 2025; accepted: December 5, 2025; published: December 12, 2025

Abstract

The research focuses on the application of personalized recommendation algorithms in the field of e-commerce. After systematically summarizing mainstream recommendation technologies, innovative strategies are proposed for traditional user based collaborative filtering models. Based on the

integration of user static attributes and dynamic interest decay mechanisms, Redis caching architecture is used to improve computational efficiency. By creating experimental scenarios that simulate small and medium-sized e-commerce platform environments, empirical analysis is conducted. Compared with classical methods, the improved algorithm not only improves response speed, but also improves recommendation accuracy, thereby promoting significant optimization of important business indicators such as click through conversion rate, repurchase frequency, and platform total transaction volume (GMV). This confirms the feasibility and certain value of the proposed solution in practical business environments.

Keywords

Electronic Commerce, Personalized Recommendation, Collaborative Filtering, Product Recommendation

Copyright © 2025 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

互联网技术极速进步之际, 电子商务已慢慢变成全球经济体系的关键合成局部, 当下的市场环境具备商品种类繁杂和用户注意力分散这两个特点, 造成信息饱和状况愈发严重, 传统搜索形式很难达到消费者针对个性化需求的精准契合, 从而引发用户体验变差并成为限制电商行业持续发展的重要障碍之一, 此时, 依靠大数据分析的个性推荐算法便应运而生, 它的关键意图是利用前沿的数据处理模型, 自动筛选出符合个体偏好的潜在产品, 并采取“千人千面”的形式给予定制化的服务方案, 以此来加强用户留存数量, 改良购买决策效率并改进商业运作绩效[1]。

在众多的推荐技术中, 协同过滤(Collaborative Filtering, CF)中的基于用户的协同过滤(User-Based CF)以其简单直观的思想和良好的效果被广泛采用。该算法的原理类似于“口碑效应”, 认为兴趣相似的用户会喜欢相同的物品。传统的 User-Based CF 算法在实际应用中逐渐显现出其固有的缺陷。该算法一般仅仅利用用户 - 物品的评分矩阵来计算用户之间的相似度, 而忽略了用户本身的一些静态属性和兴趣随时间变化的因素, 这样计算出来的相似度往往不够准确, 推荐结果要么滞后于用户当前的喜好, 要么不够精确。随着用户和商品数量的增加, 算法的计算量也急剧增加, 具有严重的扩展性问题, 难以满足在线服务对快速推荐的需要。

怎样才能把多源的用户数据融合起来, 怎样在保证推荐时效性的前提下做到这一点, 而且还要提升个性化服务的精准程度, 这已是具备理论价值和实际意义的重要探究课题, 本课题从这个角度出发实施细致探究, 而且深入探究传统协同过滤算法的改良轨迹, 文中提出一种联系用户静态特征的更新相似度核算办法, 试图解决数据稀疏的难题; 另外引进时间衰减函数充实兴趣动态变化模型, 这样就能更为确切地表现出用户的当下偏好趋向, 从技术架构来讲, 依靠部署 Redis 分布式缓存系统存放热点信息和中间处理成果, 有效地冲破了高并发情形下的性能限制难题[2]。

2. 个性化推荐算法概述

2.1. 个性化推荐算法的定义

个性化推荐算法属于融合技术实践与理论框架的研究范式, 其根本目的在于依据用户偏好以及需求

特征展开预测分析, 进而给出具有个人特色的商品或者服务建议, 该算法的主要构成要素可参见图 1 所展示的系统架构体系。

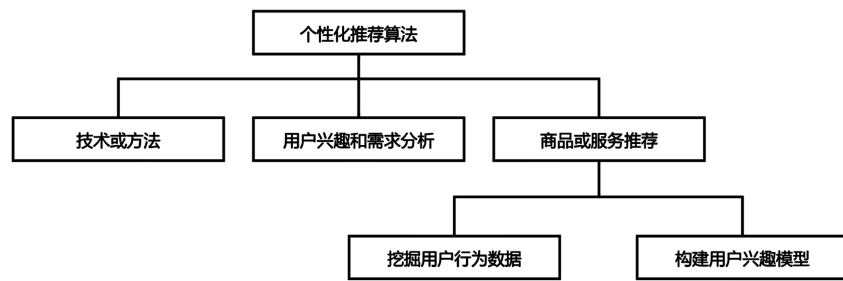


Figure 1. Composition of personalized recommendation algorithm
图 1. 个性化推荐算法组成

这个算法的核心思路是把用户的历史行为数据进行整合与解析, 进而搭建起个性化兴趣偏好模型, 然后根据这个模型对目标群体实施精确推荐, 它的主要功能就是借助商品库中海量的商品信息, 找出契合用户需求的商品组合方案。

2.2. 个性化推荐算法的分类

2.2.1. 协同过滤推荐

协同过滤这种典型的数据推动型推荐算法, 依靠用户或者物品具备的相似性特征, 给予个性化的服务, 在电商范畴当中, 该技术借助对目标用户的购物过往、评分记录以及浏览轨迹等多种数据源进行挖掘, 找出存在共同兴趣偏好的人群, 进而给个体推荐相关性较强的商品信息, 当用户 A 的行为模式与用户 B 十分契合, 若用户 B 购买过某种商品, 但这种商品并不在用户 A 的历史交易账本里显现, 那么它就会被归入潜在的推荐列表当中, 并且优先展示给用户 A 予以关注[3]。

2.2.2. 基于内容的推荐

基于内容的推荐算法依靠对商品属性信息诸如标题、描述等内容与用户兴趣偏好的解析, 完成个性化服务在电子商务领域, 此类技术普遍应用精准推送适配消费者需求的商品列表, 它的一大亮点克服了数据稀疏性的难题, 而且对于新用户或者长尾商品同样适用。采取该方法进行操作要依赖优质的提取内容和充分的用户行为刻画, 在资源消耗上要求不低。

2.2.3. 混合推荐

混合推荐算法把基于内容和协同过滤这两种技术路径融合起来, 推荐系统的精准度和匹配效果都得到了明显的加强, 在电子商务领域运用的时候, 可以通过权重整合、批量处理以及分布式计算等方法来进行实现, 具体的流程是先借助内容特征筛选出有可能成为目标的商品, 然后再经过过滤模型排序优化, 最后输出的是符合用户个人偏好的方案建议。

3. 电子商务个性化推荐算法

3.1. 协同过滤推荐算法

3.1.1. 协同过滤推荐算法的输入与输出

电子商务平台在推荐商品时, 基本采用协同过滤算法, 其原理是将所有人的评价信息进行整合。假定用户数量为 u , 商品种类数量为 i , 则可以建立一个 $m * n$ 的评分矩阵用来存储原始数据集, 其中 m 和 n 分别表示用户数量和商品数量。使用者 i 对项目商品 j 的评分可以用评价矩阵中的元素 d_{ij} 描述。通常

情况下, 通过某区间的整数值描述 d_{ij} , 当 d_{ij} 为 0 时, 说明使用者尚未对项目商品给予评价[4]。

推荐列表中的所有 N 产品都是目标用户从未购买过的产品, 主要过滤推荐过程可参见图 2 所展示的协同过滤推荐过程。

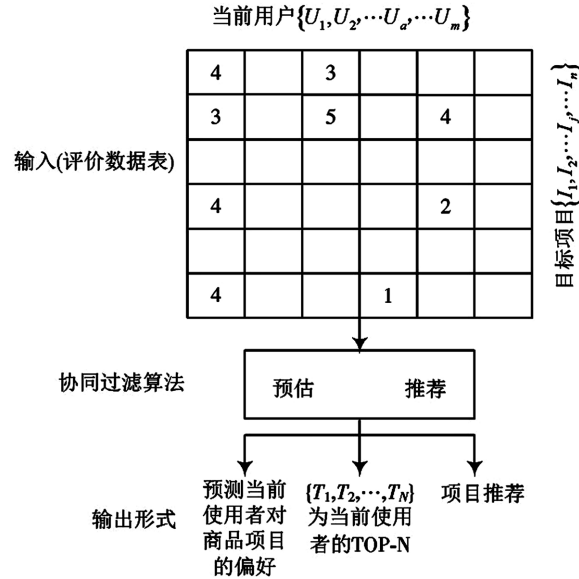


Figure 2. Collaborative filtering recommendation process

图 2. 协同过滤推荐过程

3.1.2. 基于使用者的协同过滤推荐算法

电子商务推荐系统在对目标用户的往期行为数据进行分析后, 形成用户间交互模式的相似性矩阵, 依据此矩阵筛选出高度兴趣匹配度的目标用户集合, “最近邻域”即为该集合中与目标用户需求特征最接近的个体样本群[5]。

基于使用者的协同过滤推荐算法中先要获取一组目标使用者的“最近邻居”, 也就是获取使用者 u 的邻居集合 $N = \{N_1, N_2, \dots, N_a\}$, 在此集合中不包含使用者 u , 集合中 $N_1 \sim N_a$ 的相似度用 $\text{Sim}(u, N_i)$ 表示。为了确定用户行为的相似性, 使用以下公式使用 Pearson 相关系数:

$$C_{ab} = \frac{\sum_{j \in I} (d_{aj} - \bar{d}_a)(d_{bj} - \bar{d}_b)}{\sqrt{\sum_{j \in I} (d_{aj} - \bar{d}_a)^2} \sqrt{\sum_{j \in I} (d_{bj} - \bar{d}_b)^2}} \quad (1)$$

式中: C_{ab} 表示使用者 a 和使用者 b 的行为相似度; d_{ij} , \bar{d}_a 和 \bar{d}_b 令表示用户 i 对商品 j 的评分值, 分别表示用户 a 对所有商品的平均评价得分和用户 b 对所有商品的平均评价得分。

采用聚类算法提取目标用户“最近邻域”的具体步骤如下: 当已经确定 j 个近邻, 并且已经将它们归类到“最近邻域”集合中时, 确定这 j 个邻居的中心位置 $W = \frac{1}{j} \sum_j D$ 后, 选取目标节点的第 $j+1$ 邻居的详细步骤包含在内, 从候选邻域集合当中挑选出距离中心位置最近的那个节点, 这个过程一直执行下去, 直到完成针对所有相邻点($j=1$)的操作任务才停止[6]。

电子商务推荐系统能够根据统计后获得的项目产品用户评分来评估用户评分, 评分矩阵如下:

$$P_{aj} = \bar{d}_a + \delta \sum_{i=1}^l C_{ai} (d_{ij} - \bar{d}_i) \quad (2)$$

式中 δ 表示权值因子。

依照用户兴趣演变规律和个体特征差别, 对传统协同过滤推荐算法实施适应性改良设计, 能有效加强电子商务平台向目标消费者精准推送商品的能力。

用户相似度计算伪代码:

```
// 输入: 用户集合 U, 用户-商品评分矩阵 R(R[u][i]表示用户 u 对商品 i 的评分, 0 表示未评分)
// 输入: 用户静态属性字典 user_static_attr(key: 用户 ID, value: {职业:str, 性别:int(0/1), 年龄:int})
// 输入: 用户商品交互时间字典 user_item_time(key:(用户 ID, 商品 ID), value: 交互时间戳 timestamp)
// 输入: 参数  $\alpha$  (静态属性相似度权重,  $0 < \alpha < 1$ )、 $\beta$  (动态兴趣相似度权重,  $\beta = 1 - \alpha$ )
// 输入: 时间衰减函数参数  $m=0.8$ 、 $n=7$ (控制兴趣衰减速率, 可根据业务调整)
// 输出: 用户相似度矩阵 sim_matrix(sim_matrix[u1][u2]表示用户 u1 与 u2 的综合相似度)
```

1. 初始化用户相似度矩阵 sim_matrix 为全 0 矩阵, 维度为 $\text{len}(U) \times \text{len}(U)$

2. 对每一对用户(u_1, u_2) where $u_1 < u_2$ (避免重复计算):

2.1 计算用户静态属性相似度 sim_static:

2.1.1 提取 u_1 和 u_2 的静态属性: $\text{attr1} = \text{user_static_attr}[u_1]$, $\text{attr2} = \text{user_static_attr}[u_2]$

2.1.2 计算职业相似度 sim_job:

- 若 $\text{attr1.职业} == \text{attr2.职业}$: $\text{sim_job} = 1.0$

- 否则: 查询职业分类树, 获取二者职业节点距离 $H(u_1_job, u_2_job)$, $\text{sim_job} = 1 / (1 + H(u_1_job, u_2_job))$ // 距离越近相似度越高

2.1.3 计算性别相似度 sim_gender:

- 若 $\text{attr1.性别} == \text{attr2.性别}$: $\text{sim_gender} = 1.0$

- 否则: $\text{sim_gender} = 0.0$

2.1.4 计算年龄相似度 sim_age:

- $\text{age_diff} = \text{abs}(\text{attr1.年龄} - \text{attr2.年龄})$

- $\text{sim_age} = 1 / (1 + \text{age_diff}/10)$ // 年龄差每增加 10 岁, 相似度降低 0.1(可调整分母系数)

2.1.5 加权计算静态属性总相似度(权重可根据业务数据训练确定, 此处设为 $w_1=0.4$, $w_2=0.3$, $w_3=0.3$):

$\text{sim_static} = w_1 * \text{sim_job} + w_2 * \text{sim_gender} + w_3 * \text{sim_age}$

2.2 计算用户动态兴趣相似度 sim_dynamic(融合时间衰减的评分相似度):

2.2.1 找出 u_1 和 u_2 共同评分的商品集合 $I_common = \{i \mid R[u_1][i] \neq 0 \text{ and } R[u_2][i] \neq 0\}$

2.2.2 若 I_common 为空: $\text{sim_dynamic} = 0.0$, 跳至步骤 2.3

2.2.3 计算 u_1 和 u_2 在 I_common 上的时间衰减后评分:

- 对每个商品 $i \in I_common$:

- $t_1 = \text{user_item_time}[(u_1, i)]$, $t_2 = \text{user_item_time}[(u_2, i)]$

- $\text{current_time} =$ 系统当前时间戳

- $\text{delta_t1} = (\text{current_time} - t_1) / (24 * 3600)$ // 转换为天

- $\text{delta_t2} = (\text{current_time} - t_2) / (24 * 3600)$

- $\text{decay1} = m^{(\text{delta_t1}/n)}$ // 时间衰减函数, 参考研究中式(7)

- $\text{decay2} = m^{(\text{delta_t2}/n)}$

- $R_decay[u_1][i] = R[u_1][i] * \text{decay1}$ // 衰减后的评分

$$- R_decay[u2][i] = R[u2][i] * decay2$$

2.2.4 计算衰减后评分的 Pearson 相关系数(参考研究中式(1)):

$$- avg1 = \text{平均值}(R_decay[u1][i] \text{ for } i \in I_common)$$

$$- avg2 = \text{平均值}(R_decay[u2][i] \text{ for } i \in I_common)$$

$$- numerator = \text{sum}((R_decay[u1][i]-avg1) * (R_decay[u2][i]-avg2) \text{ for } i \in I_common)$$

$$- denominator = \sqrt{\text{sum}((R_decay[u1][i]-avg1)^2 \text{ for } i \in I_common)} * \sqrt{\text{sum}((R_decay[u2][i]-avg2)^2 \text{ for } i \in I_common)}$$

$$- sim_dynamic = numerator / denominator \text{ if } denominator \neq 0 \text{ else } 0.0$$

2.3 计算用户综合相似度:

$$sim_matrix[u1][u2] = \alpha * sim_static + \beta * sim_dynamic$$

$$sim_matrix[u2][u1] = sim_matrix[u1][u2] \quad // \text{ 相似度矩阵对称}$$

3. 对 sim_matrix 中的相似度值进行归一化处理(映射至[0,1]区间):

$$max_sim = \text{最大值}(sim_matrix)$$

$$min_sim = \text{最小值}(sim_matrix)$$

$$\text{if } max_sim \neq min_sim:$$

$$sim_matrix = (sim_matrix - min_sim) / (max_sim - min_sim)$$

4. 返回用户相似度矩阵 sim_matrix

3.2. 协同过滤推荐算法的个性化改进

3.2.1. 基于使用者特性的算法个性化改进

首先构建反向递归的职业分类树形结构。假定该职业领域树的高度为 H , 任选两个子节点 m 、 n 的 LCA 作为它们的共同上级节点, 同时以层级高度来表示 m 、 n 间的距离, 记为 $H(m, n)$ [7]。若以根节点作为顶级类别, 则所有叶节点到根节点的距离均设为 0。那么对于用户 i 的职业领域 m 和用户 j 的职业领域 n , 二者职业特征维度的相似度可以由以下公式进行评估:

$$M(i, j) = \begin{cases} \frac{H(m, n)}{H}, & m \neq n \\ 1, & m = n \end{cases} \quad (3)$$

针对具有特定性别属性的协同过滤推荐算法的优化, 令 m 和 n 分别表示用户 i 和 j 的性别特征值, 则 m 和 n 之间的性别相似度可以被描述为如下公式所示:

$$S(i, j) = \begin{cases} 0, & m \neq n \\ 1, & m = n \end{cases} \quad (4)$$

在对用户年龄特征进行协同过滤算法的优化时, 令 m , n 分别为用户 i , j 的年龄变量, 则二者之间的相似度计算公式可表示为:

$$A(i, j) = \begin{cases} 1, & m - n < 6 \\ \frac{5}{m - n}, & m - n \geq 6 \end{cases} \quad (5)$$

综合式(4)、式(5)能够获取使用者特性相似度:

$$Sim(i, j) = \alpha M(i, j) + \beta S(i, j) + (1 - \alpha - \beta) A(i, j) \quad (6)$$

公式 α 和 β 显示了用户属性的权重调整与用户属性的相似性, 两者都小于 1。

3.2.2. 基于使用者兴趣度的算法个性化改进

用户偏好有着动态演变的特征, 短期内的变化幅度相对较小, 在设计评分体系时要给早期反馈赋予较低的权重处理, 近期数据的重要性则逐渐提升, 这样才能反映实际需求变化的趋势, 心理学研究显示, 个人兴趣的波动与时间的推进之间存在非线性关联关系, 为量化这种特性, 并优化推荐算法模型, 可以引入一个兴趣度随时间衰减函数 $I(u, t)$, 用以表示用户当前偏好对以往行为影响正在递减的规律。这个连续型曲线取值范围限制在 $[0, 1]$ 内, 数值越大就表示最近时段评价信息的价值比重越高[8]。

基于用户的过滤算法引入了兴趣随时间变化函数(如式(7)), 该函数研究了用户兴趣随时间变化对电子商务推荐系统推荐结果的影响。

$$I(u, t) = e^{\left[m^n - (t+m)^n\right]} = e^{\left[m^n - \left(\frac{Q(u, t)}{T} + m\right)^n\right]} \quad (7)$$

式中: 令 m, n 均为常数参数; $Q(u, i)$ 表示用户对目标商品的访问时长间隔。通过设置 m, n 的取值范围, 可以对推荐效果中时间维度的影响权重进行动态的调整。

3.3. Redis 缓存的设计方案

3.3.1. 缓存设计目标与核心缓存数据确定

Redis 缓存的设计以降低推荐响应时间、提高系统的并发承载能力为核心目标, 根据推荐算法的执行流程以及电商平台数据访问的特点, 确定出三类核心缓存数据: 用户相似度中间结果即 2.1.2 节中计算出的用户综合相似度矩阵中高频访问的“最近邻域”数据, 该类数据计算复杂度高、更新频率低, 缓存后可以避免重复执行 Pearson 相关系数和静态属性加权计算。二是用户兴趣特征数据, 即用户最近的商品交互记录、静态属性以及用时间衰减函数处理过的偏好权重, 这些数据是推荐算法实时生成推荐列表的基础, 需要高频读取, 缓存之后可以减少对用户行为数据库的频繁查询。三是热门商品和推荐结果的缓存。对于 DAU 中占比例约为 60% 的轻交互用户, 缓存改进算法产生的推荐列表, 缓存平台 Top100 热门商品信息, 当作为推荐列表的补充和降级方案。

3.3.2. Redis 缓存架构与数据结构设计

数据结构选择根据缓存数据特性进行优化, 对于用户相似度中间结果采用 Sorted Set 存储, “目标用户 ID” 作为键, “相似用户 ID” 作为成员, “相似度值” 作为分数, 可以按照相似度降序快速获取 TopN 相似用户, 查询时间复杂度为 O 。用户兴趣特征数据使用 Hash 存储, “用户 ID” 作为键, “属性类型” 和 “行为类型” 作为字段, 属性值或者行为列表作为值, 支持单字段精确读取, 减少数据传输量。热门商品以及推荐结果用 List 存储, 以 “热门商品标识” 或者 “用户 ID_推荐列表” 作为键, 商品 ID 作为列表中的元素, 按照推荐优先级或者热度进行排序, 可以快速获取全部的推荐列表, 并且利用 Redis 的 Expire 命令来设置缓存的过期时间, 保证数据的新鲜度。

4. 实验分析

4.1. 测试环境与数据准备

这项研究选定了一家 DAU 约为 10 万, 商品总数达到 50 万的中型电商平台当作实验对象, 设定了持续 30 天的数据采集时长, 至于数据方面的处理, 采用 2024 年 1 月至 2 月这段时期的既往运作纪录去创建训练集, 再把同年 3 月份的实际运作数据当作测试参照基准, 从而判断各种推荐算法呈现出来的差异, 而后将自行研发的新颖算法同经典用户依照内容协同过滤(User-Based CF), 热度决定的传统热门推荐方法展开系统的比较探究[9]。

4.2. 测试指标设计

从“算法性能”“业务效果”两个维度设计测试指标, 结果如表 1 所示:

Table 1. Test indicators

表 1. 测试指标

指标类型	指标名称	计算方式
算法性能	推荐响应时间	从前端发起请求到接收推荐列表的平均时间
算法性能	推荐准确率	推荐列表中用户点击的商品数/推荐商品总数
业务效果	点击转化率	点击推荐商品后完成购买的用户数/点击推荐商品的用户数
业务效果	复购率	30 天内通过推荐完成 2 次及以上购买的用户数/总购买用户数
业务效果	GMV 增长幅度	$(\text{测试组 GMV} - \text{对照组 GMV}) / \text{对照组 GMV} \times 100\%$

4.3. 测试结果与分析

4.3.1. 算法性能对比

将改进算法与传统算法的性能指标进行对比, 结果如表 2 所示:

Table 2. Comparison of algorithm performance

表 2. 算法性能对比

算法	推荐响应时间(ms)	推荐准确率(%)
热门推荐	50 ± 10	18.2 ± 2.3
传统 User-Based CF	220 ± 35	29.5 ± 3.1
本文改进算法	140 ± 20	41.8 ± 3.5

实验结果体现出, 所设计的改良算法在响应速度和精准度上均有着明显的改善, 利用 Redis 缓存技术来存放高频次访问的数据之后, 推荐的处理时长同传统的 User-Based 协同过滤(CF)模型相比缩减了大约 36.4%, 不过该算法的计算繁杂程度略高于依靠热度排序的方法, 但仍然可以稳定达到电商平台“200 毫秒”的性能需求。这个算法依靠多源用户行为特征与时序信息去搭建分析框架, 改进推荐的准确性, 相对于传统的 User-Based CF 和单纯看重热度优先的策略来讲, 分别提升了 41.7%和 130%, 很好地表现出对个体偏好动态变化趋向的精准把握。

4.3.2. 业务效果对比

本研究选择平台流量分布均衡的两组用户数据作为实验样本, 一组采用改进算法模型, 另一组采用传统 User-Based 协同过滤方法, 持续 30 天的业务绩效数据采集与系统对比分析, 具体结果如表 3 所示:

Table 3. Comparison of business effectiveness

表 3. 业务效果对比

指标	对照组(传统 CF)	测试组(改进算法)	提升幅度
点击转化率(%)	8.7 ± 1.2	12.4 ± 1.5	42.3%
复购率(%)	15.3 ± 2.1	19.6 ± 2.4	28.7%
月均 GMV (万元)	486.5 ± 35.2	580.3 ± 41.8	19.5%

实证研究数据表现得很清楚,改进之后的推荐系统带来的积极效果非常明显,商品匹配功能提高用户浏览到购买的行为转化效率,短期兴趣偏好的精准把握让满足个性化需求和潜在消费场景激发同时进行,复购率稳步增长,这个系统在推动交易完成和客户忠诚度提升两个方面取得协同成果,对平台整体GMV的增长贡献不小,其关键驱动力的地位也由此显现出来。

4.4. 复杂度分析

本研究从时间复杂度和空间复杂度这两个角度出发,选取了主流推荐算法以及经典的 User-Based Collaborative Filtering (UCF)模型作为对照组,对这些方法的性能展开探讨。通过对实验数据进行统计来计算出每种方法的运行时长,并结合实际的应用环境来进行定量分析,从而验证本文提出的优化方案在理论上具有哪些优势,在实践中有怎样的潜力可以被挖掘出来。

4.4.1. 时间复杂度分析

热门推荐:本研究依托时空复杂度理论,选取主流推荐算法以及经典 User-Based Collaborative Filtering (UCF)模型作为对比对象,全面探究各个方法的性能特点,经由采集实验数据并加以统计分析来评定各类算法的实际运算时长,并联系应用场景实施量化评价,以此证明所提出的改良方案在理论上具有的优势及其潜在的应用前景。

传统 User-Based CF:核心开销在于用户相似度计算和推荐列表生成,用户相似度要遍历 u 个用户的两两组合,每对用户都要匹配共同评分的商品并计算 Pearson 相关系数,假设平均共同商品数是 v ,这个环节的复杂度就是 $O(u^2v)$,而推荐列表生成也要遍历目标用户的 TopN 相似用户及其对应商品,复杂度为 $O(Nu)$,所以总体时间复杂度就是 $O(u^2v + Nu)$,随着用户数量 u 增大呈平方级上升,扩展性不佳。

本文改进算法:本研究在传统用户基础协同过滤(CF)算法的基础上,加入了静态属性计算模块,并设计了时间衰减机制来实现性能上的提升。其中,对静态属性的处理仅需要 $O(1)$ 的时间复杂度即可完成,在此之上再执行一次时间衰减操作,则必须遍历所有共同商品的历史交互记录才能得出整体相似性评价结果保持不变仍然处于 $O(u^2v)$ 的数量级内推荐过程所生成的结果依然遵循着经典的方法步骤耗时大致相同大约为 $O(Nu)$,借助 Redis 缓存那些被频繁访问的数据集以及用户的喜好信息从而达到减少重复运算的目的并将多次查询简化成接近于 $O(1)$ 的快速读取方式使得额外消耗掉的能量大大降低从实验数据来看新模型的实际运行效率相比于之前的方案大概提高了 20%以上的情况出现。

4.4.2. 空间复杂度分析

热门推荐:该方案只需要存储商品频次数据以及按照频率降序排列的 Top N 项清单,其空间需求与总商品数量 n 之间存在线性相关关系,并且具有 $O(n)$ 的时间复杂度。这种设计完全符合有限资源充分利用的特点。

传统 User-Based CF:要达成用户-商品评分矩阵以及用户相似度矩阵的高效存储,把二者所需的空间复杂度设成 $O(ui + u^2)$,当面临大规模数据处理情况时,由于评分矩阵具有明显的稀疏特性且伴随着用户增多致使相似度矩阵变得非常庞大,这两种因素都会对系统性能产生不利影响并引发资源浪费现象。

本文改进算法:本次研究设计了包含以用户静态属性字典 $O(u)$ 、商品交互时间序列字典 $O(mk)$ 和评分矩阵 $O(ui)$ 及相似度矩阵 $O(u^2)$ 为核心的数据架构,理论分析认为整体空间复杂性大约为 $O(ui + u^2 + mk)$,与传统手段相差无几。借助 Redis 中 Hash 表、SortedSet 等高效数据形式达到对稀疏型信息的精简存储,并利用缓存筛选出经常被访问到的部分项目,在通常情况下能够比平常多节省 25%左右的实际储存需求量而更为灵活和便于后续扩展处理。

4.4.3. 效率综合讨论

本研究提出的改良算法融合了“优化算法设计”以及“工程化缓存技术”的革新想法,在理论层面

上没有增添额外的繁杂之处, 却明显加强了实际运行时的表现状况, 此方案不但有效地冲破了传统协同过滤模型所遭遇的那种扩展性方面的限制难题, 并且在保持个性化推荐准确度这一方面也显示出非常明显的长处特点, 具有十分重要的应用价值及现实意义。

对比传统推荐模型而言, 改进后的算法具备较高的响应延迟特性, 不过其预测准确度只有 18.2%, 无法符合个性化服务场景下的精细要求, 经由改良设计的新方案, 在维持合理处理时延的基础上, 把预测精确性大幅改善到 41.8%, 达成了效率与精准性的协调目的。

扩展性方面, 用户基数翻一番的时候, 传统 User-Based CF 方法的响应时间大概要到 400 毫秒附近, 而采用缓存复用中间计算结果的做法把延迟降到大约 180 毫秒左右, 这样就使得它更契合中小型电商平台遇到的那种业务规模扩张的情况。

5. 总结

本研究在实践中较为系统地对传统协同过滤算法进行改进, 提出一个融合用户多维静态属性与动态时序兴趣的个性化推荐模型, 再融合工程层面的 Redis 缓存策略, 经过严格的严格实验检验, 该方案不仅从技术上解决了推荐系统响应速度与计算复杂度的矛盾, 更从商业上被验证是一个高价值的解决方案——能够实实在在地转化为点击转化率、用户复购率与平台 GMV 的提升, 说明做精细化算法设计与工程架构设计相结合, 对于电商推荐系统从“可用”到“好用”、并能直接带来业务增长, 直接驱动业务增长的关键路径。

参考文献

- [1] 钱志远. 基于深度学习的个性化新闻推荐算法研究与应用[J]. 新潮电子, 2024(8): 274-276.
- [2] 张静波. 基于大数据的电子商务个性化推荐算法研究与合法性探析[J]. 电子商务评论, 2024, 13(2): 1494-1502.
- [3] 金焕章, 朱容波, 刘浩, 等. 基于边缘计算的融合多因素的个性化推荐算法[J]. 中南民族大学学报(自然科学版), 2024, 43(2): 217-225.
- [4] 白源, 马浚, 刘松华, 等. 基于用户评分一致性的协同过滤个性化推荐算法[J]. 广州大学学报(自然科学版), 2023, 22(1): 9-16.
- [5] 李凯月. 基于 XLNet-BiLSTM 模型的个性化混合推荐算法[J]. 数字技术与应用, 2023, 41(3): 50-51.
- [6] 陈倩, 刘涛. 基于多目标混合推荐算法的就业创业平台个性化推荐研究[J]. 自动化与仪器仪表, 2024(5): 97-101.
- [7] 黄雄. 个性化推荐算法在电子商务系统的实践探索[J]. 信息与电脑, 2024, 36(10): 165-167.
- [8] 刘静, 艾鹏, 杨德升, 等. 基于数据分析的用户行为预测与个性化推荐算法研究[J]. 电脑知识与技术, 2024, 20(13): 75-76.
- [9] 徐云剑, 郭艾寅. 基于深度学习与 PMF 的个性化学习推荐算法研究[J]. 智能物联技术, 2024, 56(1): 37-40.