E-Commerce Letters 电子商务评论, 2024, 13(2), 3933-3944
Published Online May 2024 in Hans. https://www.hanspub.org/journal/ecl
https://doi.org/10.12677/ecl.2024.132479

Hans汉斯

基于改进的协同过滤的电子商务网站推荐系统

王　豪，谢本亮

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1日；录用日期：2024年4月12日；发布日期：2024年5月31日

摘　要

随着互联网的发展以及普及，电子商务网站的访问量与数量庞大，但是发现电子商务网站对用户检索意愿的考虑较少。针对此问题，本文使用一种基于增量改进的协同过滤(CF)的推荐算法(ICFR)，首先，通过CF算法来获取用户偏好与所推荐商品和电子商务网站之间的关系；其次，通过分析网络日志来获取用户的浏览信息，并将其归一化作为评分值；最后，通过所设计的增量算法完成历史用户偏好数据信息的更新。我们通过一些基于ICFR模型案例说明ICFR模型适用于电子商务网站的推荐。

关键词

协同过滤，增量算法，推荐算法，个性化电子商务网站

Recommendation System for E-Commerce Websites Based on Improved Collaborative Filtering

Hao Wang, Benliang Xie

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

Received: Apr. 1st, 2024; accepted: Apr. 12th, 2024; published: May 31st, 2024

Abstract

With the development and popularization of the Internet, the number of visits to personalized e-commerce website is huge. However, it was found that e-commerce websites gave less consideration to users' search intentions. To solve this problem, this paper uses an incremental Improved Collaborative Filtering (CF) Recommendation Algorithm (ICFR), firstly, the CF algorithm is used to obtain the relationship between user preferences and recommended products and e-commerce

文章引用: 王豪, 谢本亮. 基于改进的协同过滤的电子商务网站推荐系统[J]. 电子商务评论, 2024, 13(2): 3933-3944.
DOI: 10.12677/ecl.2024.132479

websites. Secondly, the user's browsing information was obtained by analyzing the network logs, and it was normalized as the scoring value. Finally, the designed incremental algorithm is used to update the historical user preference data information. We illustrate the application of the ICFR model to personalized e-commerce website recommendations through some examples based on the ICFR model.

Keywords

Collaborative Filtering, Incremental Algorithm, Recommendation Algorithm, Personalized E-Commerce Website

1. 引言

在互联网经济的快速发展下，推荐系统是其中的重要研究课题之一[1]，其应用范围广泛，包括旅游[2]、虚拟电商[3]、云购物和社交媒体[4]、股票等。京东、腾讯、亚马逊、阿里巴巴、抖音等企业都已经成功地在其网站或产品中使用了推荐系统。通过调研分析互联网上的大多数网站都可以归类于个性化网站(各大电商网站和知识网站等)。这些电子商务网站商品数量关系更注重网站内容的构建，而忽略了用户的检索意图，从而导致用户体验较差。研究如何去提高检索质量是个性化电子商务网站优化研究的重点[5]。

推荐算法分为三类[6]：基于内容的推荐[7] [8]、基于知识的推荐和协同过滤(CF)推荐[9]。CF 算法在推荐系统中具有重要的应用[10]，其使用程度最高。该算法通过挖掘用户的历史行为数据来研究用户的偏好[11]，然后根据不同的偏好对用户进行分组，并向他们推荐相似偏好的项目(商品和网站) [12]。

尽管对推荐系统的研究很多，但目前还没有在个性化电子商务网站中得到广泛应用。如果一些个性化电商网站想要实现用户意图的预测性推荐功能，需要做的工作是十分复杂的。并且发现一些不同的电子商务网站之间存在一定的重复度，大量网站之间的重复性工作导致了各种资源的浪费。对此，我们对个性化的电子商务网站的结构特征进行分析，提出了一种基于增量改进的 CF 的推荐算法，其中电子商务网站中的每个可访问链接都可以被视为 CF 模型中的一个项目。

与传统的基于 CF 算法的推荐系统相比，ICFR 模型具有以下优点：(a) 适用范围更广。此推荐系统新定义了文章中 CF 算法的元素(项目和评分)，避免了要求用户明确对项目进行评分的局限。(b) 增量更新算法实现了计算量的减少。系统根据用户的浏览行为自动更新用户的评分数据。(c) 该推荐系统的功能的模块化。电子商务网站开发人员可以在不更改或增加大量的代码的情况下实现基于 ICFR 的个性推荐。

2. 相关工作

协同过滤

基于用户的 CF 推荐算法是使用率最高、反馈最好的方法之一[13]。近年来，基于用户的 CF 推荐算法在许多推荐系统中得到了广泛的应用[14]，并且有许多研究对 CF 推荐算法改进。Jia 等人[15]开发了一种基于用户的旅游景点推荐系统。CF 算法在旅游领域得到有效应用。

CF 算法是使用具有相似偏好的群体来推荐所需的信息。CF 算法的核心是计算相似度[16]。为了识别具有相似倾向的用户，CF 推荐系统中常用的相似度计算方法有三种：皮尔逊相关系数(PCC) [17]、余弦

相似度和修正余弦相似度[18]。

皮尔逊相关系数计算公式定义如下：

$$value_{Pearson}=\frac{\sum_{c\in I_{ij}}\left(r_{i,c}-\overline{r}_i\right)\left(r_{j,c}-\overline{r}_j\right)}{\sqrt{\sum_{c\in I_i}\left(r_{i,c}-\overline{r}_i\right)^2}\sqrt{\sum_{c\in I_j}\left(r_{j,c}-\overline{r}_j\right)^2}} \tag{1}$$

其中，I_{ij} 看作两个用户 i 和 j 共同评论的集合，c 是这个集合的元素，$r_{i,c}$ 表示用户 i 对元素 c 的评价，$\overline{r}_i$ 和 $\overline{r}_j$ 分别为用户 i 和 j 的评价平均值。

余弦相似度、修正余弦相似度计算公式定义如下：

$$sim\left(x_i,x_j\right)_{value}=cos\left(x_i,x_j\right)=\frac{x_i\cdot x_j}{\left\|x_i\right\|_2*\left\|x_j\right\|_2} \tag{2}$$

其中，x_i,x_j 分别是用来分别表示元素的向量。

$$sim\left(x_i,x_j\right)_{value}=cos\left(x_i,x_j\right)=\frac{\sum_{u\in U}\left(R_{u,x_i}-\overline{R}_u\right)\left(R_{u,x_J}-\overline{R}_u\right)}{\sqrt{\sum_{u\in U}\left(R_{u,x_i}-\overline{R}_u\right)^2}\sqrt{\sum_{u\in U}\left(R_{u,x_j}-\overline{R}_u\right)^2}} \tag{3}$$

其中，U 看作两个用户 i 和 j 共同评论过的用户的集合，R_{u,x_i} 表示用户 u 对元素 x_i 的评价，$\overline{R}_u$ 为用户的评价平均值。

CF 推荐系统中的主要计算量是相似度计算过程。传统的 CF 算法主要在静态离线设置中表现最佳[19]。由于数据在不断变化和更新，系统需要一次重新计算当前用户与其他用户之间的相似度。因此，Thomas 等人[20]设计了共聚类算法的增量和并行版本，并用它来构建一个高效的实时 CF 框架。Guo Kehua 等人[21]提出了一种基于用户间相似性增量更新的可伸缩性问题解决方法，并提供了高质量推荐的潜力。

3. 系统模型及问题描述

3.1. 推荐系统模型

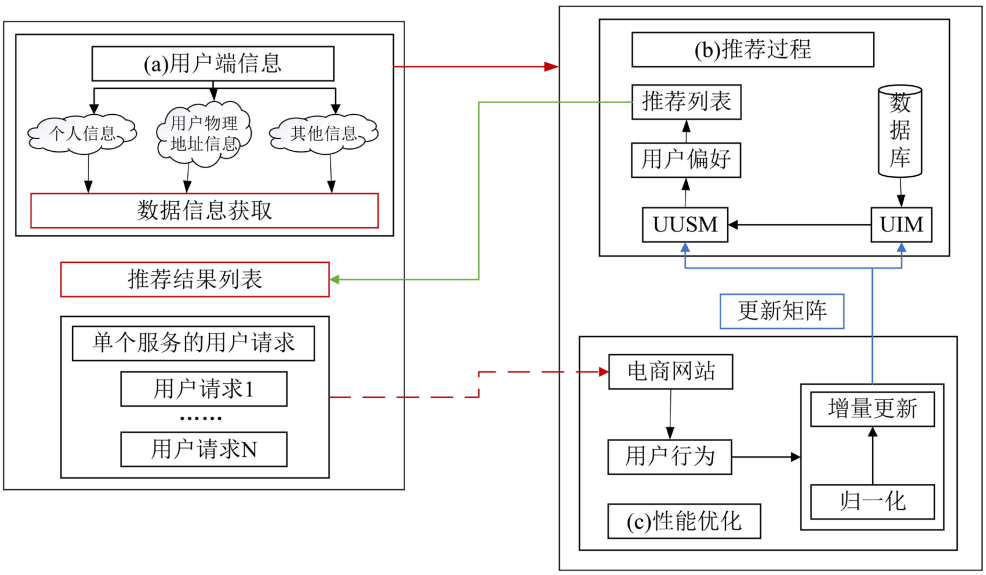


Figure 1. Architecture and work process of ICFR
图 1. ICFR 的结构和工作流程

图 1 显示了服务器的架构结构和工作流程，通过处理用户的访问行为向用户端推荐合适的项目列表并增量更新数据。该过程主要由 3 个部分组成：(1) 用户行为信息获取与规范化，(2) 项目推荐过程，(3) 增量更新计算。

在 ICFR 算法中，我们首先在用户端获取用户端信息(账户、用户端物理地址、附加需求等)用以识别当前用户。并且将用户的浏览行为(用户浏览时间、收藏夹)均记录在网站日志中，间接衡量用户对网页的偏好程度，从而用于用户项矩阵的建立(如图 1(a))。然后，服务器启动由 CF 算法实现的推荐过程。通过计算相似度矩阵，从用户－物品矩阵得到用户－用户相似度矩阵。当前用户的偏好可以通过 k 近邻算法从与当前用户更相似的其他用户中预测出来(如图 1(b))。在当前用户退出网站或关闭与服务器的当前会话时，系统将分析网站日志并将提取的数据规范化为用户对项目的评分。然后，采用增量更新算法对历史数据进行更新(如图 1(c))。ICFR 实现工作的流程如图 1 所示。

3.2. 用户信息获取

ICFR 模型采用 CF 算法来预测用户偏好。在 CF 算法中，用户、物品和评分是用户－物品矩阵的三个关键元素。传统上，我们简单地将某一类容易区分的事物视为“物品”，而“评分”则直接来源于用户评分。但在 ICFR 模型中，我们使用隐式评分的方法来衡量用户对物品的兴趣程度。

我们在 CF 算法中重新定义了“item”，将网站中所有可访问的链接都作为“item”元素，并不区分是首页、栏目或者内容页的链接。

定义 3.1：

项目集(*IS*)：给定一个网站 W_{pw} 具有 W 可访问的链接，且 $IS=\left\{n_1,n_2,\cdots,n_{|W|}\right\}$ 满足：

$$n_i=\left(n_{identity},link\right) \tag{4}$$

其中：

1) W_{pw} 是网站 W 中任意一个可访问的链接，且 $1\ll i\ll W$。

2) $n_{identity}$ 是连续的恒等式。

定义“rating”的获取方法是 ICFR 体系中项目外的另一个重要部分。目前，CF 算法中的两种主要的评分模型是显性评分和隐式评分。依赖于在电子商务网站的使用中用户不需要对项目进行评分，但是用户在网站浏览时的行为与他们的兴趣密切相关，我们选用隐式评分模型，通过从网络日志中收集用户的网站浏览行为数据，并将数据量化到一定的值范围作为用户对项目的评分。

一般来说，用户的浏览行为包括浏览次数、页面滚动时间、收藏书签、收集等动作。从网页日志中提取到用户的浏览行为数据后，我们将其量化到一定的数值范围。在 ICFR 模型中，我们可以预先在系统中提供一个规则类型的用户行为列表。我们假设有 n_{ub} 类型的用户行为预先存储在一组 $BS=\left\{M_1,M_2,\cdots,M_{ub}\right\}$ 中($M_i=\left\{m_{identity},value_{type},b_{value}\right\}$，$\forall M_i\in BS$)。身份是行为的唯一标识符号，$value_{type}$ 表示数据类型(布尔值或数值)，而 b_{value} 暂时是空值。此外，N_{ub} 的值由开发人员根据自己的需要所确定。

在 ICFR 模型中，用户集 *US* 被定义为 $US=\left\{u_1,u_2,\cdots,u_N\right\}$，满足 $u_i=\left(u_{identity},client\right)$(其中 $u_{identity}$ 是每个用户的独特标志，*client* 代表了用户或其他符号信息)。假设用户 u 在一段时间内向服务器发出请求，因此我们可以分析网站日志并从中提取一组用户行为。

定义 3.2：

用户行为集：$\Gamma=\left\{H_1,H_2,\cdots,H_S\right\}$，一组用户的一组行为满足：$M_i=\left\{u_{identity},n_{identity},BS\right\}$

其中：

1) b_{value} 的 *BS* 在设置**是从**网站日志中提取的。

2) S 表示用户访问过的项目数。

3) $u_{identity}, n_{identity}$ 分别表示用户或用户端和项目的身份。

每个用户只能为一个项目打分，因此，用户行为集 Γ 中的 b_{value} 进行归一化。在归一化过程中，我们分别处理用户行为的两种数据类型，归一化算法为表 1 所示。

Table 1. Normalization algorithm program flow
表 1. 归一化算法程序流程

Input: $\Gamma = \{H_1, H_2, ..., H_S\}$;
Output: $\gamma = \{T_1, T_2, ..., T_S\}, \{1 \le i \le S, T_i = (u_{identity}, n_{identity}, rating)\}$;

$for\ i = 1\ to\ S\ do$:
  $rating = 0$;
  $for\ j = 1\ to\ N_{ub}\ do$:
    $if\ value_{type}\ is\ Boolean\ then$:
      $if\ b_{value}\ is\ True\ then$:
        $rating = rating + 1;$
      end;
    $else$:
      $min = Min(\{M_j, b_{value} | M_j \in BS, value_{type} = numerical\ value\})$;
      $max = Max(\{M_j, b_{value} | M_j \in BS, value_{type} = numerical\ value\})$;
      $X = \frac{(M_j, b_{value} - min)}{max - min}$;
      $rating = rating + X$;
    end;
  end;
  $T_i = (u_{identity}, n_{identity}, rating)$;
  $\gamma.add(T_i)$;
end;
$return\ \gamma$;

3.3. 推荐算法

UBCF 和 IBCF 是推荐算法的两个主要算法，适合不同的应用场景，但它们的思想相似。UBCF 与 IBCF 的不同点：(1) 目的不同。(2) 建议的多样性。(3) 用户特征对算法的影响。在本文工作中，考虑到个性化电子商务网站中存在大量的商品项目和内容更新频率高的特点，对比 UBCF 和 IBCF 的不同算法之间的差异与适用性，采用 UBCF 算法，系统根据用户的共同兴趣向用户提供推荐。相似度的计算和最近邻的寻找是 UBCF 算法的两个主要工作。ICFR 的推荐过程如图 2 所示。

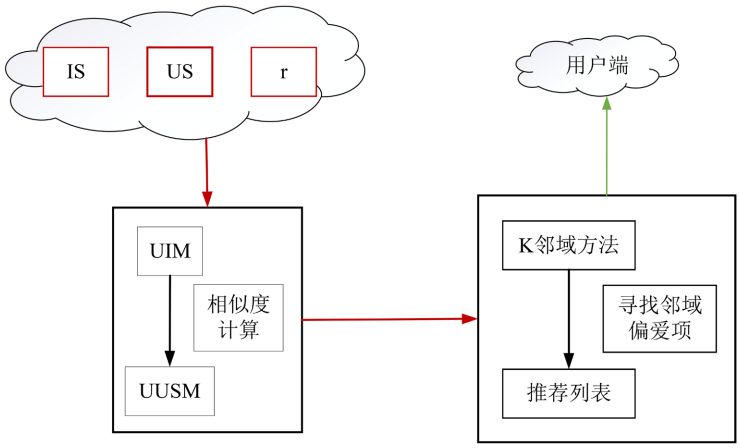


Figure 2. Process of recommendation system in the server
图 2. 服务器中推荐系统过程

此外，推荐过程中所需的数据集(IS,US,γ)由前一部分获得的。用户项矩阵$(UIM)N*W$由IS,US,γ构建。在UIM中，列表示项目，行表示用户。其中N为用户数量，W为项目数量，$r_{i,j}$表示用户i对项目j的评分，结构如下式(5)所示：

$$UIM=\begin{pmatrix} r_{11} & \cdots & r_{1W} \\ \vdots & r_{ij} & \vdots \\ r_{N1} & \cdots & r_{NW} \end{pmatrix} \tag{5}$$

用户或商品项目之间的相似性计算是 UBCF 的关键步骤。在 ICFR 模型中，选用 PCC 作为计算用户间相似度的方法。但是，在传统 CF 推荐系统中使用 PCC 方法，缺少对于用户之间共同评分的重要性的考虑。本文采用了一种改进的 PCC 方法解决共同评分的占比问题。

传统的 PCC 公式定义如下：

$$sim(u_1,u_2)=\frac{\sum_{n_i\in I_{u_1,u_2}}\left(r_{u_1,n_i}-\overline{r}_{u_1}\right)\left(r_{u_2,n_i}-\overline{r}_{u_2}\right)}{\sqrt{\sum_{n_i\in I_{u_1,u_2}}\left(r_{u_1,n_i}-\overline{r}_{u_1}\right)^2}\sqrt{\sum_{n_i\in I_{u_1,u_2}}\left(r_{u_2,n_i}-\overline{r}_{u_2}\right)^2}} \tag{6}$$

其中I_{u_1,u_2}代表的项集用户(u_1,u_2)共同评价，即$I_{u_1,u_2}=\{n_i\mid n_i\in IS,r_{u_1,n_i}\neq 0,r_{u_2,n_i}\neq 0\}$，$\overline{r}_{u_1}=\frac{1}{\left|I_{u_1,u_2}\right|}\sum_{n_i\in I_{u_1,u_2}}r_{u_1,n_i}$。在式(6)的基础上，我们在其中加入了普通额定值的权重因子CW。$CW=\sqrt{\frac{S_{u_1}^{(u_1,u_2)}}{S(u_1)}}\cdot\sqrt{\frac{S_{u_2}^{(u_1,u_2)}}{S(u_2)}}$。$S(u_i)(i\in[1,N])$表示用户对所有网页的贡献率之和。$S_{u_i}^{(u_1,u_2)}(j,i\in[1,N])$表示用户$u_i$和用户$u_j$的物品的总和。计算公式如下：

$$S(u_i)=\sum_{n_K\in I_{u_i}}r_{u_i,n_K} \tag{7}$$

$$S_{u_i}^{(u_i,u_j)}=\sum_{n_K\in I_{u_i,u_j}}r_{u_i,n_K} \tag{8}$$

因此，改进的 PCC 相似度公式表示为

$$sim(u_1,u_2)'=CW\cdot sim(u_1,u_2) \tag{9}$$

计算所有用户之间的相似度值后，得到用户－用户的相似度矩阵(user-user similarity matrix, UUSM)，其记录了每两个用户之间的相似度值，如下所示：

$$UUSM=\begin{pmatrix} S_{11} & \cdots & S_{1N} \\ \vdots & \cdots & \vdots \\ S_{N1} & \cdots & r_{NN} \end{pmatrix} \tag{10}$$

k 邻域和基于阈值的邻域是解决当前用户的最近邻寻找的问题的两种常用方法。对比两种算法的特点，考虑 ICFR 中可能的冷启动问题，本文采用 k 邻域算法，通过算法选择当前用户的最近邻。将 NNV 赋值为邻的个数，u_c作为暂存在的用户，最近邻的集为$NNS=\{u_k\mid u_k\in US,1\ll K\ll N\}$作为输出结果表示当前用户最近的邻。最后，给出了以下公式对 UUSM 中的空白评分的比例实现预测：

$$P(R_{u_c,n_i})=\frac{S(u_c)}{\left|I_{u_c}\right|}+\frac{\sum_{u_x\in NNS}sim(u_c,u_x)'\cdot\left(r_{n_i,u_x}\frac{S(u_c)}{I_{u_x}}\right)}{\sum_{u_x\in NNS}sim(u_c,u_x)'} \tag{11}$$

其中 $P\left(R_{u_c,n_i}\right)$ 代表了用户 u_c 所对应项 n_i 预测的比例，$\frac{S\left(u_c\right)}{\left|I_{u_c}\right|}$ 代表所有用户 u_c 的平均比例。

对于当前的用户 u_c 贡献率，我们已经为那些用户尚未评分的项目填写了空白评分。通过对集合 $P\left(R_{u_c,n_i}\right)$ 进行降序排列，推荐项目列表 RIL 定义为 $RIL\left(u_c\right)=\left\{n_i \mid \forall n_i \in IS, P\left(R_{u_c,n_i}\right)>P\left(R_{u_c,n_{i+1}}\right) r_{n_i,u_x}=0\right\}$。

3.4. 增量更新计算

对于 CF 算法的可扩展性作为研究 CF 算法所面临的巨大挑战之一。传统 CF 算法上，如果 UIM 矩阵发生一些变化，则需要重新计算整个 UUSM 矩阵，这个过程的时间复杂度很高。因此，本文使用增量更新去计算程序在用户与服务器关闭当前会话时的触发。假设用户 u_c 在浏览 W_{PW} 网站 $T_{session}$ 时间后，关闭了会话，在 u_c 访问 W_{PW} 网站时，在网络日志中记录了该网站的 u_c 访问行为。可以计算出 u_c 所对应的分数并把它们表示成一个集合 $R_{\tau_{u_c}}=\left(\tau_{u_c,n_1},\tau_{u_c,n_2},\cdots,\tau_{u_c,n_{|W|}}\right)$，其中 τ_{u_c,n_i} 表示用户 u_c 对项目 n_i 的归一化评分。基于 $R_{\tau_{u_c}}$，通过增量更新计算算法直接更新了用户－项目 UIM 矩阵和用户－用户 UUSM 相似度矩阵。其中 UIM 矩阵容易被所更新的新评分集 $R_{\tau_{u_c}}$ 影响。通过公式(9)来计算用户之间的新的相似度与其他用户之间的相似度 $\forall u_\alpha \in US\left(C\neq 0\right)$。其中公式(9)分解为四个因素：$X=\sum_{n_i\in I_{u_c,u_\alpha}}\left(r_{u_c,n_i}-\overline{r}_{u_c}\right)^2$，$Y=\sum_{n_i\in I_{u_c,u_\alpha}}\left(r_{u_\alpha,n_i}-\overline{r}_{u_\alpha}\right)^2$，$P=\sum_{n_i\in I_{u_c,u_\alpha}}\left(r_{u_c,n_i}-\overline{r}_{u_c}\right)\cdot\left(r_{u_\alpha,n_i}-\overline{r}_{u_\alpha}\right)$ 设 CW',X',Y',P' 分别表示更新后的分解因子，则更新后的相似度值 $sim\left(u_c,u_\alpha\right)$ 应为：

$$sim\left(u_c,u_\alpha\right)''=CW'\cdot\frac{P'}{\sqrt{X'}\sqrt{Y'}} \tag{12}$$

由式(6)和式(9)可以看出，$S_{i,j}$ 中的元素值，随着 I_{u_c,u_α} 和 I_{u_i} 的变化而变化。所以我们定义了 3 个集合，它们是 I_{u_i,u_j}、$Iuc_{I_{u_c}}$、$Chg_{I_{u_c}}$，其中($\forall n_t\in Iuc_{I_{u_c}}$，$\tau_{u_c,n_t}\neq 0$，$\tau_{u_c,n_t}\in R_{\tau_{u_c}}$)($Chg_{I_{u_c}}=Iuc_{I_{u_c}}\bigcap I_{u_c,u_\alpha}$)。这 3 个集合的变化是增量更新计算的关键因素，有两种情况可能发生：

1) $Chg_{I_{u_c}}=\varnothing$ 和 $Iuc_{I_{u_c}}=\varnothing$

基于这个条件，我们可以得到在 I_{u_c,u_α} 和 UUSM 中值是没有变化。

2) $Chg_{I_{u_c}}\neq\varnothing$ 和 $Iuc_{I_{u_c}}=\varnothing$，这表示在 $R_{\tau_{u_c}}$ 的所有项目在此之前没有被 u_c 评价过。然而 $Iuc_{I_{u_c}}$ 在 $\left(Iuc_{I_{u_c}}\cup I_{u_c,u_\alpha}\right)\cap I_{u_\alpha}$ 的变化，$\overline{r}_{u_c}\rightarrow\overline{r}_{u_c}{}'=\frac{S_{u_\alpha}^{(u_c,u_\alpha)'}}{\left|I_{u_c,u_\alpha}{}'\right|}$。

$$\begin{cases} X'=X+\sum_{n_i\in I_{u_c,u_\alpha}}\left(r_{u_c,n_i}-\overline{r}_{u_c}{}'\right)^2 \\ Y'=Y+\sum_{n_i\in I_{u_c,u_\alpha}}\left(r_{u_\alpha,n_i}-\overline{r}_{u_\alpha}{}'\right)^2 \\ P'=P+\sum_{n_i\in I_{u_c,u_\alpha}}\left(r_{u_c,n_i}-\overline{r}_{u_c}{}'\right)\cdot\left(r_{u_\alpha,n_i}-\overline{r}_{u_\alpha}{}'\right) \\ CW'=\sqrt{\frac{S_{u_c}^{(u_c,u_\alpha)'}}{S\left(u_c\right)'}}\cdot\sqrt{\frac{S_{u_\alpha}^{(u_c,u_\alpha)'}}{S\left(u_\alpha\right)'}} \end{cases} \tag{13}$$

额外的存储空间来存储高速缓存因子计算方法如下表 2 所示。

尽管增量更新法计算当前用户与其他用户之间的相似度的时间复杂度为 O(n)，与总更新法时间几乎相同，该方法相比所具有的优势：(1) 降低了计算密度。(2) 将增量更新计算分为四种情况，与 TUM 相

比，计算量更少。

Table 2. Cache factor algorithm flow
表 2. 高速缓存因子算法流程

Input: $u_c; I_{u_c}; Iuc_{I_{u_c}}; X; Y; P; R_{\tau_{u_c}}; UIM;$
Output: $UUSM; Updated\ UUSM;$

$Normalizing\left(R_{\tau_{u_c}}\right) \rightarrow update\ matrix\ UIM;$
$for\ \alpha = 0\ to\ N\ do:$
  $get\ I_{u_\alpha}\ and\ I_{u_c,u_\alpha}\ from\ matrix\ UIM;$
  $Chg_{I_{u_c,u_\alpha}} = Iuc_{I_{u_c}} \cap I_{u_c,u_\alpha};$
  $if\ Chg_{I_{u_c}} = \emptyset\ and\ Iuc_{I_{u_c}} = \emptyset:$
    $there\ is\ no\ change\ in\ matrix\ UUSM;$
  $else\ if\ Chg_{I_{u_c}} \neq \emptyset\ and\ Iuc_{I_{u_c}} = \emptyset\ then:$
    $execute\ formula(8);$
  $end;$
  $update\ X; Y; P; sim(u_c, u_\alpha)\ in\ UUSM;$
$end;$
$return\ UUSM;$

4. 实验结果

4.1. 实验

为了评估性能，我们基于 ICFR 模型实现了一个场景，服务器端模拟了一个普通的电子商务网站平台。在模拟过程中，我们从真实电子商务网站抓取近 1500 条电商信息，分别注册 910 个用户作为 CF 算法的项目集和用户集。项目集包含来自三个级别的网页的链接，可以访问主页、专栏或内容页，其中每个链接都被视为一个项目。将 ICFR 方法应用于电子商务网站个性化服务时，存在 CF 推荐算法面临冷启动问题。为了缓解这一问题，随机生成了 910 个用户的一些访问日志，并构造了用户项矩阵和用户相似度矩阵。其中的访问网站日志并不是完全随机的。根据链接的文本信息将所有项目分成 8 个主题，并让每个用户对其中一个主题存在偏好。规定了用户访问行为的评分范围归一化为 0~5。为了比较，基于以下推荐准确率和时间成本标准 ICFR 模型的性能评价。

4.2. 推荐系统的准确性

ICFR 模型中推荐的准确性是通过推荐列表中每个结果的期望来估计的。因此，将推荐准确率定义为：

$$accuracy = \frac{1}{count}\sum_{i=1}^{count} Exp(i) \tag{14}$$

其中，$count$ 是推荐结果的个数，$Exp(i)$ 表示用户对项目的期望，其范围为 $0 \ll Exp(i) \ll 1$。

在 ICFR 模型中，预测评分中当前用户的邻域的数量会在很大程度上影响准确率。然后，计算推荐列表中第一个 N_{res} 的 $accuracy$，其中 N_{res} 表示从开始计算的推荐列表中的结果个数。举例说明了 N_{res} 在不同 Num_{neigh} 的变化时 $accuracy$ 的情况如下图 3 所示。

由图 3 可以得到两个主要结论：

1) 当 Num_{neigh} 不变时，随着 N_{res} 的增大，$accuracy$ 先不变化，但 N_{res} 继续增大时，$accuracy$ 从某一点开始减小。

2) 当 N_{res} 较小且不变时，$accuracy$ 几乎是相同的，但随着 N_{res} 增大到一定值后，$accuracy$ 随着 Num_{neigh} 的增加而增加。

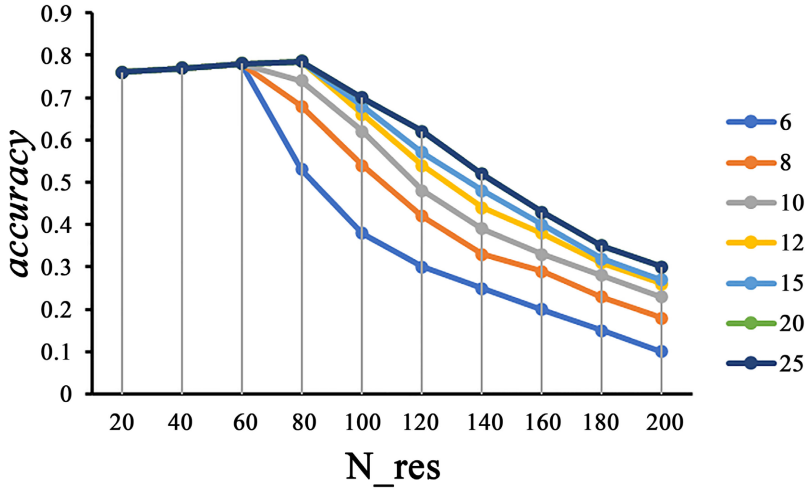


Figure 3. Accuracy of different Num_{neigh}

图 3. 不同 Num_{neigh} 的准确性

考虑到增加 Num_{neigh} 导致更多的计算量的影响。在模拟环境中，我们设置的 Num_{neigh} 的值为 12。在接下来的实验中， Num_{neigh} 保持相同的值 12， N_{res} 设为 120。

4.3. 时间成本评估

本实验比较了 IUM 和 TUM 的时间成本。在使用增量更新计算时分析了四种情况。单个增量更新过程的时间消耗 T_{inc} 包括两个用户之间的相似性计算时间和其他处理过程的时间。其定义如下：

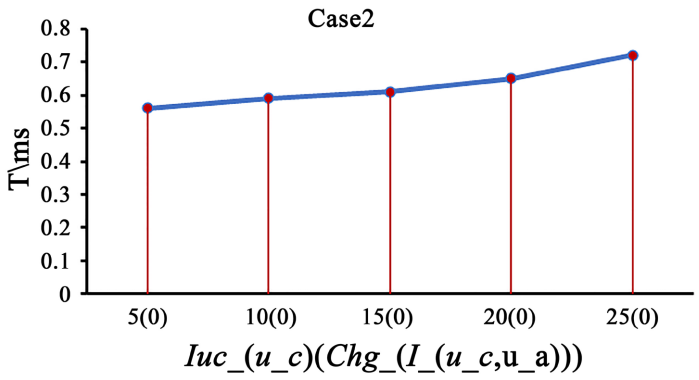
$$T_{inc} = t_{oth} + t_{sim} \tag{15}$$

其中， t_{sim} 为相似度计算时间，满足：

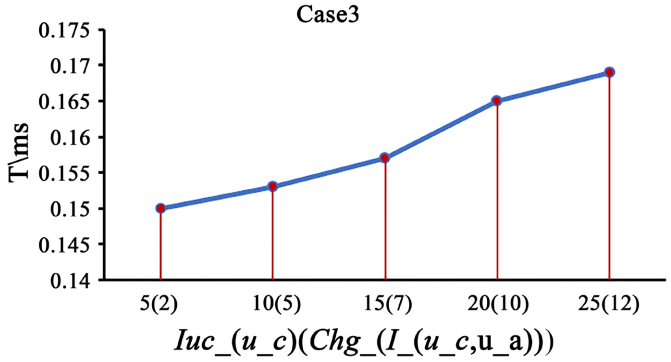
$$t_{sim} = \sum_{i=1}^{4} t_{ci} * num_{ci} \tag{16}$$

其中， t_{ci} 为 ci 情况发生的时间， num_{ci} 为 ci 情况发生的频率。

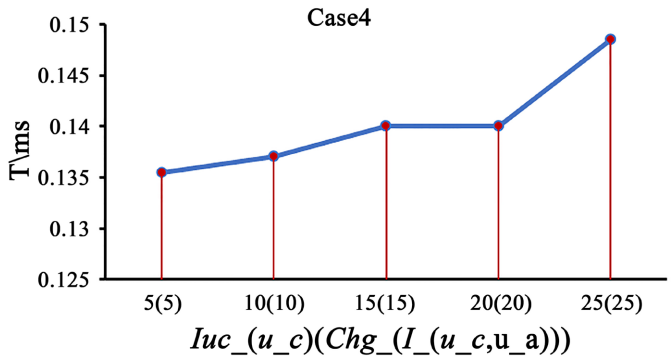
对不同的 $Chg_{I_{u_c,u_\alpha}}$ 和 $Iuc_{I_{u_c}}$ 关系设定进行时间成本实验分析，我们假设所有元素在 $Chg_{I_{u_c,u_\alpha}}$ 都来于 $Iuc_{I_{u_c}}$ ，(a) Case2， $Chg_{I_{u_c,u_\alpha}} = 0$ 。(b) Case3， $Chg_{I_{u_c,u_\alpha}} = \frac{Iuc_{I_{u_c}}}{2}$ 。(c) Case4， $Chg_{I_{u_c,u_\alpha}} = Iuc_{I_{u_c}}$ 。其结果如图 4 所示。



(a) $Chg_{I_{u_c,u_\alpha}} = 0$ 时的时间成本



(b) $Chg_{I_{uc},u_\alpha} = \frac{Iuc_{I_{uc}}}{2}$ 时的时间成本



(c) $Chg_{I_{uc},u_\alpha} = Iuc_{I_{uc}}$ 时的时间成本

Figure 4. Time cost in IUM
图 4. IUM 时间成本

实验结果从图 4 中可以得知，当 Case1 时，$t_{c0} = 0$。通过对比分析，发现在 Case2 的情况时时间成本是最大，远远大于 Case3 和 Case4 两种情况，在 Case4 中 Chg_{I_{uc},u_α} 和 $Iuc_{I_{uc}}$ 的关系所进行设定时的时间成本为最小。

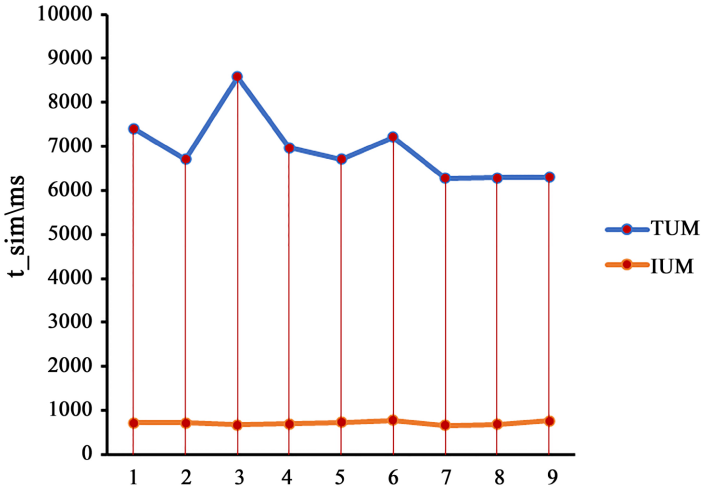


Figure 5. Time cost comparison between IUM and TUM
图 5. IUM 和 TUM 的时间成本对比实验图

另一个时间成本对比实验是 IUM 和 TUM 在 ICFR 模型中的时间成本对比实验。ICFR 中的 IUM 只需要计算当前用户的相似度信息。但是在 TUM 中，需要更新整个相似矩阵。因此，在上述两种实验中进行了 9 种不同的更新过程。实验结果如图 5 所示。

从图 5 可以发现，使用 TUM 方法在不同的更新状态下所需要的时间成本变化明显，第 3 种更新状态的情况所需要时间最长，而使用 IUM 方法在不同的更新状态下所需要的时间成本为大致相等，呈现为较低的时间成本现象。通过对两种不同的方法对比分析 TUM 花费的时间远远超过 IUM，后者约占前者的 10.3%。在本实验所使用的 IUM 方法通过时间成本估计衡量性能是远远优于 TUM 的性能。

通过两组不同的时间成本估计实验，对本文所构建的 ICFR 模型中所使用方法与参数设置的合理性进行充分的证明。

5. 结论及未来工作

本文提出了一种电子商务网站的推荐系统模型——ICFR。通过基于用户的协同过滤算法的选择实现模型的构建并且应用于电子商务网站的推荐应用中。对 UBCF 算法中所涉及主要元素(用户、商品、评分)的重新定义；本文介绍了如何将 UBCF 算法应用于电子商务网站的推荐具体方案。该方案采用改进的 PCC 相似度算法计算用户之间的相似度；讨论 UBCF 更新机制，与传统的更新方法相比，该方法减少了推荐系统的计算量。所构建的基于增量改进的协同过滤的电商推荐系统——ICFR，通过模拟的电商平台对其模型进行实验结果其模型的效果拥有较高的准确性，可以高效的实现向不同偏好的用户推荐符合其偏好的商品和电子商务网站，实验证明本文提出的模型取得了满意的推荐效果。在未来的工作中，计划为协同过滤推荐算法寻找更好的增量更新策略。

参考文献

[1] 黄玲, 黄镇伟, 黄梓源, 等. 图卷积宽度跨域推荐系统[J]. 计算机研究与发展, 2024, 61: 1-17.

[2] Meehan, K., *et al*. (2013) Context-Aware Intelligent Recommendation System for Tourism. 2013 *IEEE International Conference on Pervasive Computing and Communications Workshops* (*PERCOM Workshops*), San Diego, 18-22 March 2013, 328-331. https://doi.org/10.1109/PerComW.2013.6529508

[3] Wang, Y.Z., *et al*. (2016) A Mobile Recommendation System Based on Logistic Regression and Gradient Boosting Decision Trees. 2016 *International Joint Conference on Neural Networks* (*IJCNN*), Vancouver, 24-29 July 2016, 1896-1902. https://doi.org/10.1109/IJCNN.2016.7727431

[4] 赵付春, 邓少军. 社交媒体对科技创新网络的影响[J]. 中国科技论坛, 2015(2): 32-36.

[5] Guo, K., *et al*. (2017) SOR: An Optimized Semantic Ontology Retrieval Algorithm for Heterogeneous Multimedia Big Data. *Journal of Computational Science*, **28**, 455-465. https://doi.org/10.1016/j.jocs.2017.02.005

[6] Zhu, Y. (2023) Personalized Recommendation of Educational Resource Information Based on Adaptive Genetic Algorithm. *International Journal of Reliability, Quality and Safety Engineering*, **30**, Article ID: 2250014. https://doi.org/10.1142/S0218539322500140

[7] Ziogas, I., Streviniotis, E., Papadakis, H., *et al*. (2022) Content-Based Recommendations Using Similarity Distance Measures with Application in the Tourism Domain. *Proceedings of the* 12*th Hellenic Conference on Artificial Intelligence*, Corfu, 7-9 September 2022, Article No. 31. https://doi.org/10.1145/3549737.3549772

[8] Alghamdi, S., Sheta, O. and Adrees, M.S. (2022) A Framework of Prompting Intelligent System for Academic Advising Using Recommendation System Based on Association Rules. 2022 9*th International Conference on Electrical and Electronics Engineering* (*ICEEE*), Alanya, 29-31 March 2022, 392-398. https://doi.org/10.1109/ICEEE55327.2022.9772526

[9] Mohammed, A.A. and Hamad, M.M. (2023) Recommender Systems and Machine Learning Techniques for Large Educational Data: A Survey. 2023 16*th International Conference on Developments in eSystems Engineering* (*DeSE*), Istanbul, 18-20 December 2023, 782-787. https://doi.org/10.1109/DeSE60595.2023.10469586

[10] Zhao, D., Xiu, J., Bai, Y., *et al*. (2016) An Improved Item-Based Movie Recommendation Algorithm. 2016 4*th International Conference on Cloud Computing and Intelligence Systems* (*CCIS*), Beijing, 17-19 August 2016, 278-281. https://doi.org/10.1109/CCIS.2016.7790269

[11] Wu, D., Xiao, E., Zhu, Y., *et al*. (2023) Efficient Retrieval of the Top-*k* Most Relevant Event-Partner Pairs. *IEEE Transactions on Knowledge and Data Engineering*, **35**, 2529-2543. https://doi.org/10.1109/TKDE.2021.3118552

[12] Yang, W.K., Wang, Z.Y. and Sun, C.Y. (2015) A Collaborative Representation Based Projections Method for Feature Extraction. *Pattern Recognition*, **48**, 20-27. https://doi.org/10.1016/j.patcog.2014.07.009

[13] Pu, X. and Zhang, B. (2020) Clustering Collaborative Filtering Recommendation Algorithm of Users Based on Time Factor. 2020 *Chinese Control and Decision Conference* (*CCDC*), Hefei, 22-24 August 2020, 364-368. https://doi.org/10.1109/CCDC49329.2020.9164315

[14] (2019) Collaborative Filtering for Predicting and Tracking Performance of Measurement Apparatus on Different Applications. Research Disclosure. https://xueshu.baidu.com/usercenter/paper/show?paperid=13250mb0726k00t08g070mv0um256815&site=xueshu_se

[15] Jia, Z.Y., *et al*. (2015) User-Based Collaborative Filtering for Tourist Attraction Recommendations. *IEEE International Conference on Computational Intelligence & Communication Technology*, Ghaziabad, 13-14 February 2015, 22-25. https://doi.org/10.1109/CICT.2015.20

[16] Liu, J., Li, D., Gu, H., *et al*. (2023) Personalized Graph Signal Processing for Collaborative Filtering. *Proceedings of the ACM Web Conference*, Austin, 30 April-4 May 2023, 1264-1272. https://doi.org/10.1145/3543507.3583466

[17] Huang, H.C., Zheng, S. and Zhao, Z. (2010) Application of Pearson Correlation Coefficient (PCC) and Kolmogorov-Smirnov Distance (KSD) Metrics to Identify Disease-Specific Biomarker Genes. *BMC Bioinformatics*, **11**, P23. https://doi.org/10.1186/1471-2105-11-S4-P23

[18] Zarei, M.R., Moosavi, M.R. and Elahi, M. (2022) Adaptive Trust-Aware Collaborative Filtering for Cold Start Recommendation. *Behaviormetrika*, **50**, 541-562. https://doi.org/10.1007/s41237-022-00161-3

[19] Anelli, V.W., Bellogín, A., Di Noia, T., *et al*. (2022) Top-N Recommendation Algorithms: A Quest for the State-of-the-Art. *UMAP* '22: *Proceedings of the* 30*th ACM Conference on User Modeling*, *Adaptation and Personalization*, Barcelona, 4-7 July 2022, 121-131. https://doi.org/10.1145/3503252.3531292

[20] George, T. and Merugu, S. (2005) A Scalable Collaborative Filtering Framework Based on Co-Clustering. *IEEE International Conference on Data Mining*, Houston, 27-30 November 2005, 4 p.

[21] Guo, K., *et al*. (2018) Transparent Learning: An Incremental Machine Learning Framework Based on Transparent Computing. *IEEE Network*, **32**, 146-151. https://doi.org/10.1109/MNET.2018.1700154