Journal of Computing and Electronic Information Management
ISSN: 2413-1660 | Vol. 17, No. 2, 2025

Optimization of Weighted Hybrid Algorithm based on Collaborative Filtering and Content Filtering in Steam Game Recommendation

Siman Wang, Yi Xu, Sai Li, Juan Li *

Wuhan Business School, Wuhan, Hubei Province 4300010, China
* **Corresponding author**

Abstract: As the number of games on the Steam platform continues to grow, it becomes more and more difficult for users to quickly find the games that match their interests among the huge number of games. This study aims to optimize a set of weighted hybrid recommendation algorithms integrating collaborative filtering and content filtering to improve the accuracy and experience of personalized game recommendations. By writing a crawler program in Python, we collect game data on the Steam platform in real time, and construct an original dataset containing game types, ratings, vendors, and other dimensions. Subsequently, cluster analysis is used to model users' historical purchasing behavior, combined with content filtering algorithm to extract game feature vectors, and collaborative filtering algorithm is applied to calculate user similarity, and finally weighted hybrid strategy is used to generate recommendation results. The research results show that the method has good practical application value in alleviating the cold-start problem and improving recommendation diversity and accuracy.

Keywords: Content filtering; Collaborative filtering; Cluster analysis; Weighted hybrid algorithm; Game recommendation.

1. Introduction

In recent years, the gaming industry has shown explosive growth and become an important part of the global cultural entertainment industry. With the continuous development of Internet technology and hardware devices, the types, quality and number of users of games have grown significantly. Among them, Steam has become the world's leading digital game distribution platform with its powerful ecosystem and massive game resources. However, the surge in the number of games on the platform has also brought about the problem of information overload, and users are faced with a lot of interference and trouble in the selection process.

For this reason, the introduction of personalized recommendation is particularly important. Guan Fei (2021), on the other hand, optimized collaborative filtering recommendation algorithms, including SlopeOne-based algorithm and improved K-means algorithm. [1] Lu Hao-Yin (2025) studied the integration and application of collaborative filtering and content-based recommendation algorithms in movie recommendation system, and explored the effective integration strategies and methods of collaborative filtering and content recommendation algorithms[2] . Hongsheng Bian (2024) improved K-means clustering using center aggregation parameter,and proposed a collaborative filtering recommendation algorithm that integrates improved K-means clustering and item content[3] . Miao Wang and Dawei Li (2023) Advantages and Disadvantages of Content-based Recommendation Algorithm and User-based Collaborative Filtering Algorithm, combined the two algorithms and constructed a hybrid algorithm based recommender system, which improves the problem of data sparsity[4] . Zhao Yuanyuan, Zhang Xiaolei (2023) in-depth study of content-based and collaborative filtering recommendation algorithms based on the article proposed in the collaborative filtering recommendation algorithm based on the incorporation of the time factor and with the content-based recommendation algorithm right fusion of the music recommendation algorithm[5] . Pan Yue (2023) chose Movie Lens-1m dataset to study the hybrid recommendation based on content and collaborative filtering algorithm, and then compared the experiment with traditional collaborative filtering recommendation algorithm[6] . Liu Xu et al. (2022) used hybrid recommendation algorithm to calculate the weight matrix of users and markers, the similarity between the two, and then realize personalized recommendation, to complete the construction of the whole cloud vocal music teaching system[7] . Xueding Li et al. (2021) used a cascade approach to mix the two recommendation algorithms, which alleviated the user cold-start problem, the item cold-start problem and the data sparsity problem to a certain extent, and overcame the limitations of a single algorithm[8] . Gao Fei et al. (2022) applied the content-based recommendation algorithm and collaborative filtering algorithm in recommender system to the personalized learning of students' weak knowledge points and final assessment analysis respectively[9] . LeiF et al. (2021) proposed an online marketing recommendation algorithm based on content integration and collaborative filtering, which calculates the similarity between the marketing content and the fusion model to form a set of user scores combining features. ratings,and then through K-means clustering,the recommendation is finally realized[10] . GeethaG et al. (2018) constructed a more accurate movie recommendation system by mining the movie database and collecting all the important information that needs to be recommended, such as popularity, attractiveness, etc., by using content-based collaborative filtering and hybrid filtering techniques, and by combining the results of these two techniques[11] .

This study proposes a weighted hybrid recommendation method based on content filtering and collaborative filtering, aiming at realizing efficient and accurate game recommendation from user interests. Content filtering focuses on analyzing the characteristics of the games purchased by users, such as genres and manufacturers, to find similar content; collaborative filtering explores similar users through their purchasing behaviors, and realizes pushing with their preferred behaviors. On the basis of integrating the advantages of the two, a weighting strategy is used to combine the results of the two types of recommendations to further optimize the diversity and accuracy of recommendations.

In order to ensure the practical feasibility of game recommendation and data integrity, this paper adopts Python to write a crawler program to automatically collect game information on the Steam platform. The acquired data is pre-processed by filling missing values, cleaning duplicates and extracting feature vectors, which constitutes a complete data base and serves the recommendation algorithm modeling and optimization process. This study aims to improve the efficiency and intelligence of recommendation algorithms in practical application scenarios, and at the same time provides a technical path and optimization strategy for the research of game recommendation algorithms.

2. Personalized Recommendation Algorithms and Hybrid Model Optimization

2.1. Similarity calculation

In the field of data analysis and machine learning, similarity calculation aims to quantify the degree of association between different data objects, and its core idea is to assess the similarity by measuring the spatial distance between data vectors, in general, the shorter the distance between the vectors, the higher the similarity between the two. The details are as follows:

2.1.1. Cosine similarity

The cosine similarity method is often used when measuring the degree of similarity between user rating vectors or item feature vectors. This method places the vectors in space and takes the cosine value of the angle between the two vectors as the similarity metric. As shown in equation (1):

$$\cos(\theta) = \frac{\vec{A} \cdot \vec{B}}{|\vec{A}||\vec{B}|} = \frac{\sum_{i=1}^{n} a_i b_i}{\sqrt{\sum_{i=1}^{n} a_i^2}\sqrt{\sum_{i=1}^{n} b_i^2}} \tag{1}$$

2.1.2. Pearson's correlation coefficient

In a recommendation algorithm system, it can be used to measure the correlation between the ratings of two users for items that have been evaluated together, or the correlation between the ratings of two items that have been evaluated by the same users.

Suppose there are two vectors $\vec{X} = (\chi_1, \chi_2, \cdots, \chi_n)$ and $(\vec{Y} = y_1, y_2, \cdots, y_n)$, the computation is shown in Equation (2):

$$r_{XY} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{2}$$

where $\bar{x}$ and $\bar{y}$ are the mean values of $\vec{X}$ and $\vec{Y}$, respectively.

2.1.3. Euclidean distance

Evaluates the straight-line distance between two vectors in n-dimensional space. The closer the distance, the more similar the two vectors are. For two vectors $\vec{A} = (a_1, a_2, \cdots, a_n)$ and $\vec{B} = (b_1, b_2, \cdots, b_n)$, the computation is shown in equations (3):

$$d(\vec{A}, \vec{B}) = \sqrt{\sum_{i=1}^{n}(a_i - b_i)^2} \tag{3}$$

2.1.4. Jaccard Similarities

The Jaccard similarity coefficient accurately measures the degree of similarity between two sets, with values in the interval 0 to 1. The closer the value is to 1, the higher the degree of overlap between the two sets and the stronger the similarity. For two sets A and B, the Jaccard similarity $J(A, B)$, is calculated as shown in equation (4):

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} \tag{4}$$

2.2. K-Means clustering algorithm

The K-Means clustering algorithm, as a representative algorithm in the field of unsupervised learning, focuses on classifying and structuring data through an iterative optimization strategy that accurately aggregates a given dataset into K clusters with similar characteristics. The algorithm initially assigns K data points as cluster centers at random, and then classifies each sample point into the cluster of its closest center of mass based on Euclidean distance and other metrics. After the samples are classified, the center of mass is re-determined by calculating the mean value of all samples in the cluster.

In practical applications, data tends to change dynamically, e.g., game data is constantly updated and new games are constantly on the shelves. The stochastic nature of the clustering algorithm allows the algorithm to better adapt to such dynamic changes, and each time the algorithm is run, due to the different initial state, the algorithm is able to re-explore and clustering the new data distribution, thus providing recommendation results that are more in line with the current data situation.

In clustering algorithms, the distance calculation mostly used is Euclidean distance. For two sample points x and y, it is defined as in equation (5):

$$d(x, y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \tag{5}$$

The algorithm flow is to first initialize and randomly select k data points as the initial center of mass, i.e.,$\mu_1, \mu_2, \cdots, \mu_k$. Then each data point (x_i) is assigned to the cluster where the closest center of mass is located, and the calculation is shown in Equation (6):

$$C_i = \arg\min_j |x_i - \mu_j|^2 \tag{6}$$

The center of mass of each cluster is subsequently recalculated and the calculation is shown in Eqs. (7):

$$\mu_j = \frac{1}{|C_j|}\sum_{x_i \in C_j} x_i \tag{7}$$

Repeat the above two steps until the center of mass no longer changes or the maximum number of iterations is reached.

The optimization objective of K-means is to minimize the sum of the squared distances of all data points to the center of mass of the cluster to which they belong (known as inertia or intra-cluster sum of squares), defined as in Equation (8):

$$J = \sum_{j=1}^{k}\sum_{x_i \in C_j} |x_i - \mu_j|^2 \tag{8}$$

The algorithm keeps decreasing the value of J through iterations until it converges.

3. Steam Game Data Collection and Preprocessing

Data acquisition and preprocessing is an indispensable link in the optimization of game recommendation algorithms, which is of great significance in ensuring the effectiveness of recommendations, improving data quality and availability, optimizing performance and efficiency, and enhancing the adaptability and scalability of algorithmic functions.

3.1. Steam game data collection

In order to obtain detailed and accurate game data information and complete the optimization recommendation algorithm, this study crawls the game data from Steam official website (https://store.steampowered.com/search/%20?specials=1&page=%s). Using web crawler technology, the crawler part is also written in Python under PyCharm, and the crawling process is divided into the crawling of the game basic data and the crawling of the game detail page data, with the file names "spider.py" and "spiderdetail.py" respectively. ". After writing the code to import the relevant libraries, we should first configure and set up the Selenium library and Chrome driver for automation testing, the specific code is shown in Figure 1 (the basic crawling and detail page data crawling file of the same part of the code):

Figure 1 creates a Service object that specifies the path to the Chrome driver chromedriver.exe, which is used to control the automation of Chrome, and then creates a ChromeOptions object to set the startup options for Chrome, adding an experimental option to the Chrome browser using the add_ experimental_option method, setting the debugger address to localhost:9225, which means that the browser instance will be connected to a local instance running on port 9225. experimental_option method to add an experimental option for Chrome, set the debugger address to localhost:9225, which means that the browser instance will be connected to the debugger running locally on port 9225, which is convenient for crawling data with too many browsing windows resulting in crawling failures and errors. webdriver.Chrome method in conjunction with the Service and ChromeOptions objects created earlier to start a Chrome instance and assign it to the browser variable.

In crawling the basic information of the game data, because the page is scrolling, so in order to make it possible to crawl to the whole page of data, in the console of the page to find the documentscrollheight of the page, and then set the max_scroll in the code to 3000, with a while loop, and then set a wait time, so as to make the crawling smoother, the specific part of the code in Figure 2, the code will not be able to crawl, but will be able to crawl to the whole page. The specific part of the code is shown in Figure 2:

Then is the most important part of the crawler code, the game base data crawling using circular nesting, mainly is the cycle of crawling to the different URL and extract information, while the details of the crawling page is based on the base page to crawl to the detailLink link data for each game details link one by one crawling, both are the use of XPath to locate the game elements, and then crawling the information they need and save it to the database. information and save it to the database. The main code of this part is shown in Figures 3 and 4.

After crawling the data will be stored in the specified csv file, in order to prevent the data in the file character format, so we open the file in binary mode and detect the encoding, and then use the detected encoding to open the file and then insert the data into the database.

```python
s=Service("chromedriver.exe")
option=webdriver.ChromeOptions()
option.add_experimental_option("debuggerAddress","127.0.0.1:9225")
browser=webdriver.Chrome(service=s,options=option)
```

Figure 1. Configuration and setup code for automated testing

```python
max_scroll = 3000
current_scroll = 0
while current_scroll < max_scroll:
    browser.execute_script("window.scrollBy(0, 100)")
    time.sleep(0.1)
    current_scroll += 100
```

Figure 2. Setting Crawl Page Height Code

```python
game_list = browser.find_elements(By.XPATH, value='//a[@class="search_result_row ds_collapse_flag app_impression_tracked "]')
for game in game_list:
    try:
        title = game.find_element(By.XPATH, value='.//div[@class="responsive_search_name_combined"]/div[1]/span[@class="title"]').text
        icon = game.find_element(By.XPATH, value='.//div[@class="col search_capsule"]/img').get_attribute('src')
        times = game.find_element(By.XPATH, value='.//div[@class="responsive_search_name_combined"]/div[2]').text
        evaluate = ''
        try:
            evaluate_element = game.find_element(By.XPATH, value='.//div[@class="responsive_search_name_combined"]/div[3]/span')
            if re.search('mixed', evaluate_element.get_attribute('class')):
                evaluate = '一般'
            else:
                evaluate = '好评'
        except Exception as e:
            print(f'获取评价信息失败. URL: {url}. 错误信息: {e}')
        try:
            discount = 100 - int(re.search('\d+', game.find_element(By.XPATH, value='.//div[@class="discount_pct"]').text).group())
        except:
            discount = 0
        try:
            origin_price = re.search('\d+', game.find_element(By.XPATH, value='.//div[@class="discount_original_price"]').text).group()
        except:
            origin_price = ''
        try:
            now_price = re.search('\d+', game.find_element(By.XPATH, value='.//div[@class="discount_final_price"]').text).group()
        except:
            now_price = ''
        detailLink = game.get_attribute("href")
        print(detailLink)
        save_to_csv(title, icon, times, evaluate, discount, origin_price, now_price, detailLink)
    except Exception as e:
        print(f'处理游戏信息失败. URL: {url}. 错误信息: {e}')
```

Figure 3. Main code for crawling the game data base page

```python
for type in browser.find_elements(by=By.XPATH, value='//div[@class="glance_tags popular_tags"]/a'):
    if type.text:
        types.append(type.text)

try:
    summary = browser.find_element(by=By.XPATH, value='//div[@class="game_description_snippet"]').text
except:
    summary = '无'

recentlyComment = ''
allComment = ''
try:
    if re.search('mixed', browser.find_element(by=By.XPATH, value='//*[@id="userReviews"]/div[1]/div[2]/span[1]').get_attribute("class")):
        recentlyComment = '一般'
    else:
        recentlyComment = '好评'
except Exception:
    recentlyComment = '无'

try:
    if re.search('mixed', browser.find_element(by=By.XPATH, value='//*[@id="userReviews"]/div[2]/div[2]/span[1]').get_attribute("class")):
        allComment = '一般'
    else:
        allComment = '好评'
except Exception:
    allComment = '好评'
firm = browser.find_elements(by=By.XPATH, value='//div[@class="summary column"]/a')[0].text
try:
    publisher = browser.find_elements(by=By.XPATH, value='//div[@class="summary column"]/a')[1].text
except:
    publisher = ''

imgList = [x.get_attribute('src') for x in browser.find_elements(by=By.XPATH, value='//div[@class="highlight_strip_item highlight_strip_screenshot"]/img')]
try:
    video = browser.find_element(by=By.XPATH, value='//video').get_attribute('src')
except:
    video = ''
querys('UPDATE games SET types = %s, summary=%s, recentlyComment=%s, allComment=%s, firm=%s, publisher=%s, imgList=%s, video=%s WHERE id=%s',
       [json.dumps(types, ensure_ascii=False), summary, recentlyComment, allComment, firm, publisher,json.dumps(imgList, ensure_ascii=False), video, id])
```

Figure 4. Main code for crawling game data detail page

3.2. Data pre-processing

Data preprocessing is an indispensable step in Steam game recommendation, which is important for improving data quality, enhancing data availability and optimizing algorithm performance.

3.2.1. Missing value processing

Vacant data will directly affect the quality of the data, which in turn affects the accuracy and quality of the recommended games, and thus fails to meet the users' needs. By replacing the missing values, we can ensure the completeness of the data to a certain extent, and thus provide valuable recommendation information. For numeric columns, fill them with 0; for non-numeric columns, fill them with the string 'unknown', and finally save and return to the original file. The code is shown in Figure 5:

```python
import pandas as pd
import os
folder_path=r"C:\Users\86158\Desktop\steam网页游戏推荐"
for filename in os.listdir(folder_path):
    if filename.endswith('.csv'):
        file_path = os.path.join(folder_path, filename)
        try:
            df = pd.read_csv(file_path)
            df = df.dropna()
            numerical_columns = df.select_dtypes(include=['number']).columns
            df[numerical_columns] = df[numerical_columns].fillna(0)
            non_numerical_columns = df.select_dtypes(exclude=['number']).columns
            df[non_numerical_columns] = df[non_numerical_columns].fillna('unknown')
            df.to_csv(file_path, index=False)
            print(f"{filename}文件的缺失值处理完成。")
        except Exception as e:
            print(f"处理{filename}文件时出错:{e}")
```

Figure 5. Missing Value Handling

3.2.2. Repeat value processing

The duplication of data will directly affect the accuracy of the data and the recommendation effect later, in order to enhance the data quality and recommendation effect, we can use df.duplicated() method will return a Boolean Series, which is used to indicate whether each row is a duplicate row. This can be used as an index to filter out duplicate rows, and then the df.drop_duplicates() method removes the duplicate rows from the DataFrame. As shown in Figure 6:

```python
for filename in os.listdir(folder_path):
    if filename.endswith('.csv'):
        file_path = os.path.join(folder_path, filename)
        try:
            df = pd.read_csv(file_path)
            duplicate_rows = df[df.duplicated()]
            if not duplicate_rows.empty:
                print(f'{filename}文件中存在(len(duplicate_rows))条重复记录。')
                df = df.drop_duplicates()
                df.to_csv(file_path, index=False)
                print(f'{filename}文件的重复值处理完成。')
            else:
                print(f'{filename}文件中不存在重复记录。')
        except Exception as e:
            print(f'处理{filename}文件时出错:{e}')
```

Figure 6. Repeat Value Processing

3.2.3. Data type conversion

Among them, the game's type, rating and other data belong to categorical data, which can be converted to the Categorical type of pandas to save memory and improve processing speed. As shown in Figure 7:

The csv file after the above data cleaning is stored in the MySQL database, and the final data is shown in Figure 8:

```python
for filename in os.listdir(folder_path):
    if filename.endswith('.csv'):
        try:
            df = pd.read_csv(file_path)
            categorical_columns=['types','evaluate']
            for col in categorical_columns:
                if col in df.columns:
                    df[col] = df[col].astype('category')
            df.to_csv(file_path,index=False)
            print(f'{filename}文件的类别类型转换完成。')
        except Exception as e:
            print(f'处理{filename}文件时出错:{e}')
```

Figure 7. Type Conversion

id	title	icon	time	evaluate	discount	origin_p	now_pric	types	summary	recently	allComme	firm	publishe	imglist	video
1	Counter-S	https://s	2012-8-21	Positive	0	0	0	First-Per	For over	Positive	Positive	Valve	Free to F	https://s	https://v
2	Elden Rir	https://s	2025-5-29	Positive	0	198	198	Soulslike	Elden Rir	Positive	Positive	FromSoftw	Role-Play	https://s	https://v
3	PUBG: BAT	https://s	2017-12-2	Mixed	0	0	0	Survival|PUBG, the		Mixed	Mixed	PUBG Corp	Adventure	https://s	https://v
4	Apex Lege	https://s	2020-11-4	Mixed	0	0	0	Free to F	Developed	Mixed	Mixed	Respawn	Adventure	https://s	https://v
5	NBA 2K25	https://s	2024-10-2	Mixed	0	298	298	Sports|Ba	Dominate	Mixed	Mixed	Visual Co	Sports	https://s	https://v
6	FANTASY L	https://s	2025-5-21	Positive	0	268	268	Role-Play	Fish, coo	Positive	Positive	LEVEL5 In	LEVEL5 In	https://s	https://v
7	Split Fic	https://s	2025-3-6	Positive	0	198	198	Co-op|Spl	Delve int	Positive	Positive	Hazelight	Adventure	https://s	https://v
8	ARK: Surv	https://s	2017-8-27	Positive	70	87.4	17.4	Open-Worl	Built on	Positive	Positive	Studio Wi	Adventure	https://s	https://v
9	Delta For	https://s	2024-12-4	Mixed	0	0	0	Free to F	The class	Mixed	Mixed	Team Jade	Adventure	https://s	https://v
10	Rune Fact	https://s	2025-6-4	Positive	0	258	258	Role-Play	Experienc	Positive	Positive	Marvelous	Adventure	https://s	blob:http
11	POPUCOM	https://s	2025-6-1	Positive	10	69.4	59.4	Co-op|Adv	Together	Positive	Positive	Hypergryp	Adventure	https://s	blob:http
12	EA SPORTS	https://s	2024-9-26	Mixed	0	248	248	Sports|Sc	EA SPORTS	Mixed	Mixed	EA Canada	Sports	https://s	https://v
13	Dota 2	https://s	2013-7-9	Positive	0	0	0	Free to F	Every day	Positive	Positive	Valve	Strategy	https://s	https://v
14	Street Fi	https://s	2023-6-1	Positive	0	198	198	2D Fighti	The lates	Positive	Positive	CAPCOM Co	Adventure	https://s	https://v
15	NARAKA: B	https://s	2021-8-11	Positive	0	0	0	Battle Ro	This glob	Positive	Mixed	24 Entert	Adventure	https://s	https://v
16	Wallpaper	https://s	2018-11-1	Positive	0	22.9	22.9	Adult|Uti	You can u	Positive	Positive	Wallpaper	Indie	https://s	https://v
17	Once Huma	https://s	2024-7-10	Positive	0	0	0	Multiplay	Once Huma	Positive	Positive	Starry St	Adventure	https://s	https://v
18	Monster H	https://s	2018-8-10	Positive	67	115.84	48.84	Co-op|Mul	Monster H	Positive	Positive	CAPCOM Co	CAPCOM Co	https://s	https://v
19	Stardew V	https://s	2016-2-27	Positive	0	48	48	Farming S	You inher	Positive	Positive	Concerned	Role-Play	https://s	https://v
20	Yu-Gi-Oh!	https://s	2022-1-19	Positive	0	0	0	Card Game	The ultim	Positive	Mixed	KONAMI	Strategy	https://s	https://v
21	Sultan's	https://s	2025-3-31	Positive	0	80	80	Role-Play	Sultan is	Positive	Positive	Double Cr	Indie	https://s	https://v
22	Dying Lig	https://s	2022-2-4	Positive	67	132.34	65.34	Open Worl	Humanity	Positive	Positive	Techland	Adventure	https://s	https://v
23	Forza Hor	https://s	2021-11-9	Positive	0	248	248	Racing|Op	Drive the	Positive	Positive	Playgroun	Adventure	https://s	https://v
24	Monster H	https://s	2022-1-13	Positive	75	124.5	49.5	Action|Hu	Monster H	Positive	Positive	CAPCOM Co	CAPCOM Co	https://s	https://v
25	Mahjong S	https://s	2020-7-15	Mixed	0	0	0	Psycholog	Enjoy mah	Mixed	Mixed	Catfood S	Strategy	https://s	https://v
26	World of	https://s	2017-11-1	Positive	0	0	0	Free to F	Immerse y	Positive	Positive	Wargaming	MMO	https://s	blob:http
27	War Thund	https://s	2013-8-15	Positive	0	0	0	Free to F	War Thund	Positive	Positive	Gaijin En	MMO	https://s	blob:http
28	No Man's	https://s	2016-8-13	Positive	60	130	70	Open Worl	No Man's	Positive	Positive	Hello Gam	Adventure	https://s	https://v
29	F1 25	https://s	2025-5-30	Positive	0	298	298	Racing|Sp	Leave you	Positive	Positive	Codemaste	Simulatio	https://s	https://v
30	Stellaris	https://s	2016-5-9	Positive	0	168	168	Space|Gra	In this s	Positive	Mixed	Paradox I	Strategy	https://s	https://v
31	Don't St	https://s	2016-4-22	Positive	0	24	24	Survival|In Don't		Positive	Positive	Klei Ente	Adventure	https://s	https://v
32	The Outla	https://s	2024-3-5	Positive	60	114.4	54.4	Horror|Mu	Red Barre	Positive	Positive	Red Barre	Adventure	https://s	https://v
33	Raft	https://s	2022-6-21	Positive	0	68	68	Survival|Raft? thr		Positive	Positive	Redbeet I	Indie	https://s	https://v
34	Monster T	https://s	2025-5-21	Positive	0	92	92	Strategy|Monster T		Positive	Positive	Shiny Sho	Strategy	https://s	blob:http
35	Tom Clanc	https://s	2015-12-2	Positive	0	98	98	First-Per	Tom Clanc	Positive	Positive	Ubisoft M	Ubisoft M	https://shared.fas	
36	Tainted G	https://s	2025-5-23	Positive	0	152	152	Adventure	Step into	Positive	Positive	Questline	Adventure	https://s	https://v
37	Monster H	https://s	2025-2-28	Mixed	0	368	368	Hunting|A	Rugged ar	Mixed	Mixed	CAPCOM Co	Adventure	https://s	https://v
38	Marvel Ri	https://s	2024-12-6	Positive	0	0	0	Free to F	Marvel Ri	Positive	Mixed	NetEase G	Free to F	https://s	https://v
39	Phasmopho	https://s	2020-9-19	Positive	25	82	57	Horror|Or	Phasmopho	Positive	Positive	Kinetic G	Indie	https://s	https://v
40	Assassin'	https://s	2025-3-20	Positive	0	348	348	Action|Ac	In Assass	Positive	Positive	Ubisoft Q	Adventure	https://s	https://v
41	Crusader	https://s	2020-9-2	Positive	0	198	198	Strategy|Fall in l		Positive	Positive	Paradox I	Simulatio	https://s	https://v
42	Terraria	https://s	2011-5-17	Positive	0	42	42	Open-Worl	Dig, figh	Positive	Positive	Re-Logic	Adventure	https://s	https://v
43	It Takes	https://s	2021-5-26	Positive	0	0	0	Co-op|Mul	Download	Positive	Positive	Hazelight	Adventure	https://shared.fas	
44	It Takes	https://s	2021-3-26	Positive	0	198	198	Co-op|Mul	Play It T	Positive	Positive	Hazelight	Adventure	https://s	https://v
45	Summer Me	https://s	2020-6-15	Positive	80	91.8	11.8	Adult Cor	Summer is	Positive	Positive	Dojin Otc	Role-Play	https://s	https://v
46	Slay the	https://s	2019-1-23	Positive	66	94.9	28.9	Deck-Buil	We combin	Positive	Positive	Mega Crit	Strategy	https://s	https://v
47	Astral Pa	https://s	2024-2-29	Mixed	0	0	0	Adult Cor	Star Engi	Mixed	Mixed	STAR ENGI	Indie	https://s	https://v
48	Satisfact	https://s	2024-9-10	Positive	30	125.2	95.2	Base Buil	Satisfact	Positive	Positive	Coffee St	Indie	https://s	https://v
49	Escape th	https://s	2022-8-12	Positive	0	37	37	Horror|Mu	Escape th	Positive	Positive	Fancy Gam	Early Acc	https://s	https://v

Figure 8. Data Crawling Result Chart

4. The function implementation and optimization effect analysis of weighted recommendation algorithm

To realize the game recommendation function, the first step is the content filtering algorithm, which reads the current user's data, then acquires all the game data and converts them into DataFrame format, extracts the features of the games purchased by the user, including the game type and the manufacturer, and converts the features into vectors using the TF-IDF vectorization method. Then we select a representative game from each cluster, merge the features of the representative game, and calculate its cosine similarity with all the games, and finally sort according to the similarity scores, and select the top 10 similar games as the recommendation results; followed by collaborative filtering algorithms to obtain the purchase records of all the users and construct the user-game matrix, and then calculate the similarity to find the other users who are the most similar to the current user, and recommend the similar games that the current user has purchased but the current user has not purchased. The next step is the collaborative filtering algorithm, which obtains the purchase records of all users and constructs the user-game matrix, calculates the similarity to find other users who are most similar to the current user, and recommends the games that similar users have purchased, but the current user has not. Finally, weighted mixing is performed to get the final recommendation result.

Taking user 1 as the user who is logged in at the moment, the information of the serial number of the purchased game for this user is (1, 2, 3, 4, 5, 6, 7, 9, 10, 15, 19, 21, 22, 27, 29, 30, 34, 55, 68, 71, 76, 90, 95, 110, 112, 115, 120), and that of the other users are respectively:

User 2: (1, 3, 4, 5, 6, 12, 14, 17, 22, 30, 84, 85, 150, 191, 193, 195, 199, 205, 208);

User 3: (3, 9, 12, 15, 21, 29, 50, 53, 54, 56, 58, 88, 90).

Through the clustering algorithm and similarity calculation and weighted hybrid calculation can get the final recommendation results, click to get the recommendation, its recommended 10 games as shown in Figure 9:

Get the full 10 game recommendations, including Overcooked! 2, Nuclear Age, Summer Rhapsody: Memories of the Countryside, Roller Coaster Star 2, Rainy Adventure 2, Devil May Cry HD Collection, DevilMay Cry 4 Special Edition, Hello! We've got a love affair ahead of us, Devil May Cry 5 - playable character "Vergil", Jurassic World Evolution 2: Camp Cretaceous Dinosaur Pack, as well as detailed information about the corresponding games, so that users can learn more about these games. Click Get Recommendations again, and the result is shown in Figure 10:

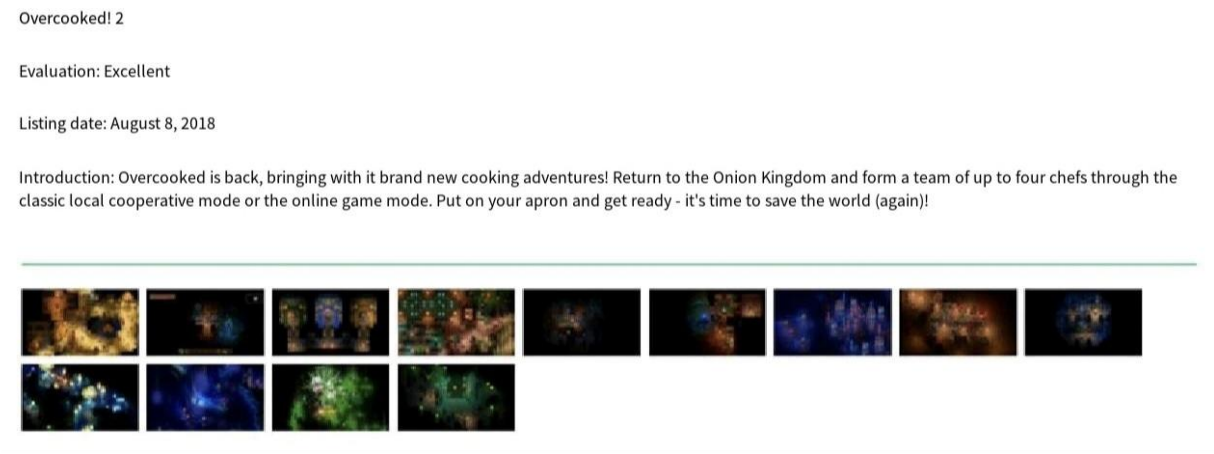

Figure 9. Graph of first game recommendation results

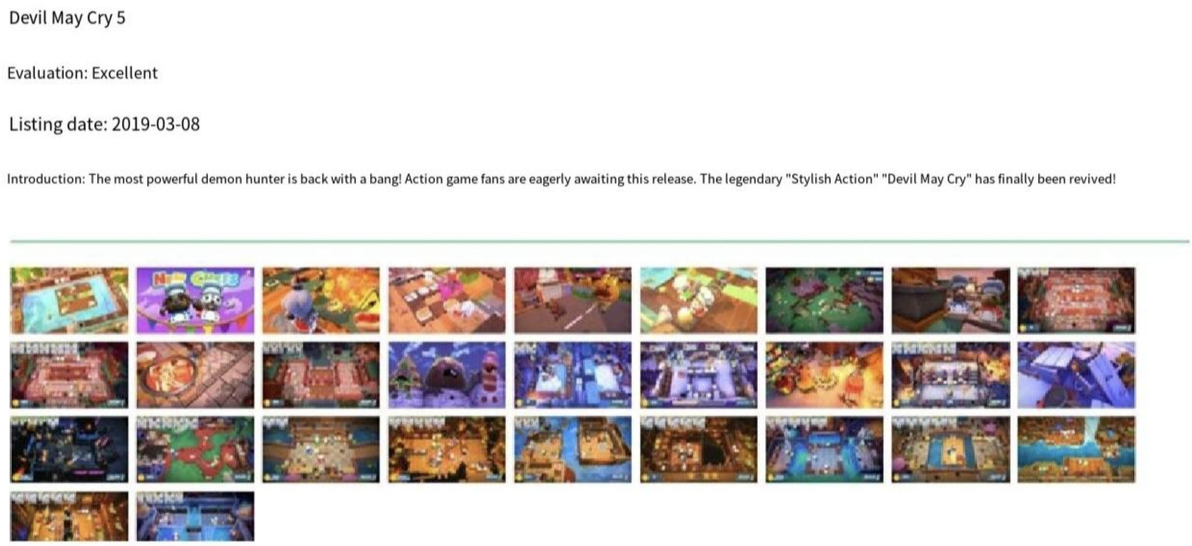

Figure 10. Graph of second game recommendation results

Get the full 10 game recommendation results including Devil May Cry 5, Overcooked! 2, Protector Core Chronicles, Summer Rhapsody: Unforgettable Memories in the Countryside, Adventure in the Rain 2, Devil May Cry 4 Special Edition, Hello! We've Got a Love Affair Left, Void Survivor, Overcooked! 2 - Surf 'n' Turf, Devil May Cry 5-Playable Characters "Vergil". So when the party gets the recommendation for the first time and the user is not very satisfied, they can click to get it again, thus satisfying the user and making them have more choices.

It is not difficult to find that, due to the clustering algorithm to select the k value of the randomness of the problem, resulting in each time to obtain the recommendation of the results are not exactly the same, although the randomness of this helps to avoid falling into the local optimal solution, at the same time help to prevent the model overfitting, and from a practical point of view, because the data tend to change dynamically, so if the game data will be updated, new games will continue to be shelves, clustering algorithm of the Randomness allows the algorithm to better adapt to this dynamic change, and each time the algorithm is run, due to the different initial state, the algorithm is able to re-explore and clustering of the new data distribution, so as to provide recommendation results that are more in line with the current data situation. Ultimately, this can make our recommendation results more diverse and thus more responsive to the personalized needs of users.

5. Summary

This study focuses on solving the core problem of the mismatch between information overload and personalized needs in Steam platform game recommendation, and constructs a set of optimized game recommendation models through a weighted hybrid algorithm that integrates collaborative filtering and content filtering. In this study, we first use Python crawler technology to collect real-time game data from Steam platform, covering multi-dimensional features such as genres, ratings, vendors, etc., and construct a high-quality raw dataset through preprocessing steps such as missing value filling, duplicate value processing, and type conversion, which lays a solid data foundation for the algorithmic modeling.

At the algorithm design level, the study innovatively combines the advantages of content filtering and collaborative filtering: the content filtering algorithm generates similar game recommendations by extracting the characteristic attributes (e.g., genres, vendors) of the games that users have already purchased, which effectively captures the users' explicit interests; the collaborative filtering algorithm mines similar user groups based on users' purchasing behaviors, and pushes the potentially interesting games that the target users haven't been exposed to, which solves the limitations of a single algorithm under the data This solves the limitations of a single algorithm in data sparsity and cold-start scenarios. By introducing the K-Means clustering algorithm to model users' historical behaviors, the accuracy of user clustering is further optimized, and the computational methods such as cosine similarity and Pearson's correlation coefficient are used to improve the computational accuracy of feature vectors and user similarity. Ultimately, the diversity and accuracy of recommendations are significantly improved by integrating the results of the two types of algorithms through a weighted hybrid strategy.

Although the research has achieved some results, there is still room for improvement. The current data only focuses on structured data (e.g., genre, evaluation), and in the future, unstructured data such as user comments, game screenshots, etc. can be introduced to deepen feature extraction through natural language processing and computer vision technology; the weighting strategy adopts fixed weights, and a dynamic weight adjustment mechanism based on factors such as user activity and game popularity can be explored in the future; in addition, the algorithm can be further extended to other gaming platforms such as Epic and GOG to realize cross-ecosystem personalized recommendation. In addition, the algorithm can be further extended to other game platforms such as Epic, GOG, etc. to realize cross-ecological personalized recommendation.

This study provides a reusable technical path for the game recommendation field and verifies the effectiveness of hybrid algorithms in improving the performance of recommendation systems. Future research can further explore the dynamic weight adjustment mechanism, optimize the feature representation by combining deep learning techniques, and expand to cross-platform user behavior analysis to further enhance the intelligence level and generalization ability of recommendation.

Acknowledgements

The authors gratefully acknowledge the financial support from Innovation and Entrepreneurship Training Program of Wuhan Business University (202311654193), Academic team for big data analysis, mining and security(2023TD008)

References

[1] GUAN Fei,ZHOU Yi,ZHANG Han. Optimization study of collaborative filtering recommendation algorithm in personalized recommendation system[J]. Operations Research and Management,2022,31(11):9-14.

[2] Lu Hao-Yin. Research on the integration of collaborative filtering and content-based recommendation algorithms in movie recommendation system[J]. Computer Knowledge and Technology,2025,21(01):79-81.

[3] Bian Hongsheng. Research on travel recommendation algorithm based on hybrid content and collaborative filtering [D]. Shenyang: Shenyang University of Architecture,2024.

[4] WANG Miao,LI Dawei. Application of hybrid recommendation algorithm based on content and collaborative filtering in digital science and technology museum[J]. Network Security Technology and Application,2023,(08):37-39.

[5] Zhao Yuanyuan,Zhang Xiaolei. Weighted fusion of content-based and collaborative filtering music recommendation algorithms[J]. Journal of Fuyang Institute of Vocational Technology,2023,34(03):51-55+69.

[6] Pan Yue. Research on movie recommendation system based on content and collaborative filtering algorithm [D]. Harbin: Heilongjiang University,2023.

[7] LIU Xu,ZHAO Shu-chang,SHAO Mingzhu. Application of hybrid recommendation algorithm based on collaborative filtering and content for teaching cloud vocal music[J]. Industrial Control Computer,2022,35(10):141-142+145.

[8] LI Xueding,YANG Lyric,SAIYAJE DILICHATI,et al. Research on the application of hybrid recommendation algorithms incorporating content and collaborative filtering[J]. Computer Technology and Development,2021,31(10):24-29+37.

[9] Gao Fei,Chen Deli,Yan Tao. Personalized assessment using content recommendation and collaborative filtering algorithms[J]. Journal of Anhui University (Natural Science Edition),2022,46(02):22-29.

[10] Lei F,XiaoMing M.An Improved Recommendation Method Based on Content Filtering and Collaborative Filtering[J].Complexity,2021,2021

[11] Geetha G,Safa M,Fancy C, et al. A Hybrid Approach using Collaborative filtering and Content based Filtering for Recommender System[J]. Physics: Conference Series,2018,1000(1):012101-012101.