Royal University Of Phnom Penh
Faculty of Engineering
Information Technology of Engineering

Steam Game Recommendation System: A Report

Group 4:
Cheam Norakpanha
Kong Pov
Keo Vuththana

Supervisor: Lecturer Chanpiseth Chap

Submission date: December 25th, 2024

Table of Content

Steam Game Recommendation System: A Report

1. Introduction

The Steam platform hosts thousands of games, catering to diverse gaming preferences. With the abundance of choices available, it can be challenging for users to discover new games tailored to their interests. To address this, a game recommendation system was developed using the Apriori algorithm. This report outlines the steps taken to design, implement, and evaluate this recommendation system.

1.1. What is the Apriori Algorithm?

The Apriori algorithm is one of the most widely used algorithms for association rule mining and data mining. It operates on the principle that subsets of frequent itemsets must also be frequent. This algorithm helps identify frequent patterns, associations, or relationships among data items in large datasets.

1.2. Components of the Apriori Algorithm

The Apriori algorithm is built on three key components: support, confidence, and lift.

- **Support:** Support measures how frequently an itemset appears in the dataset. It is calculated as the number of transactions containing the itemset divided by the total number of transactions. In this context, support helps identify itemsets that occur often enough to be considered significant.
- **Confidence:** Confidence indicates the likelihood of an item B being purchased when item A is purchased. It is calculated as the number of transactions containing both A and B divided by the number of transactions containing A. Therefore, confidence is crucial for understanding the strength of an association between items.
- **Lift:** Lift assesses the strength of a rule compared to random co-occurrence. It is calculated as the confidence of the rule divided by the support of the consequent (item B). Notably, lift values greater than 1 indicate a positive association, while values less than 1 suggest a negative or no association.

1.3. How the Apriori Algorithm Works in Data Mining

The Apriori algorithm follows a systematic process to identify meaningful patterns and associations within datasets. It begins by identifying all individual items in the dataset and calculating their support values, filtering out items that do not meet the minimum support threshold. This ensures that only frequently occurring items are considered for further analysis.

Next, the algorithm combines the remaining items into larger itemsets and repeats the process of calculating support and filtering. This iterative approach builds progressively larger combinations of items, identifying only those that occur frequently enough to be significant. Once the frequent itemsets are identified, the algorithm proceeds to generate association rules. These rules highlight the relationships between items and are evaluated against a minimum confidence threshold to ensure reliability.

Finally, the lift metric is used to assess the strength of each association rule, comparing its occurrence to what would be expected by chance. Rules with a lift value greater than one are retained, as they represent meaningful associations that can inform actionable insights, such as game recommendations for users. This structured approach enables the Apriori algorithm to uncover valuable patterns in large datasets effectively.

1.4. Advantages and Disadvantages of the Apriori Algorithm

The Apriori algorithm offers several advantages and disadvantages when applied to data mining tasks. One of its main advantages is its simplicity and ease of implementation, making it accessible for beginners and researchers alike. It is particularly effective for identifying frequent itemsets in small to medium-sized datasets and generates interpretable rules that aid decision-making processes.

However, the algorithm also has limitations. It can become computationally intensive for large datasets because it requires repeated scanning of the database, which increases processing time and memory usage. Additionally, selecting appropriate support and confidence thresholds is critical, as overly strict thresholds may result in too few rules, while lenient ones may lead to an overwhelming number of rules that lack relevance.

Despite these challenges, the Apriori algorithm is a valuable tool for uncovering patterns and relationships in datasets. In this project, it was leveraged to demonstrate how personalized game recommendations can enhance user experiences on platforms like Steam.

2. Objectives

The objectives of this project are aligned with building an effective and user-centric recommendation system for Steam games. We aim to uncover patterns in user behavior and leverage these insights to enhance user engagement. Specifically, the goals include:

- To identify frequent combinations of games played or purchased by users on Steam
- To generate association rules that provide personalized game recommendations
- To visualize the most popular game combinations based on user transactions

3. Methodology

The development of the Steam Game Recommendation System was structured into six distinct steps, ensuring a systematic approach to achieving the project's objectives.

3.1. Step 1: Data Collection

Simulated data was used for this project, representing the games played or purchased by individual users on Steam. Each user's game library was treated as a transaction, with duplicate entries removed to ensure data accuracy.

Below is an example of the dataset, showcasing a few transactions for illustrative purposes. The full dataset consists of 300 users with varying game combinations:

User	Games
1	["Counter-Strike", "Rainbow Six Siege", "Terraria", "Team Fortress 2", "Stardew Valley", "Rust"]
2	["Call of Duty: Modern Warfare", "Stardew Valley", "Terraria", "Left 4 Dead 2", "Among Us", "Dota 2", "Rust", "ARK: Survival Evolved", "Team Fortress 2", "Portal 2"]
3	["Left 4 Dead 2", "Hollow Knight", "Terraria", "ARK: Survival Evolved", "The Witcher 3", "Cyberpunk 2077", "Counter-Strike", "Fall Guys"]

4	["ARK: Survival Evolved", "Garry's Mod", "Hollow Knight", "Phasmophobia", "Stardew Valley", "Fall Guys", "Terraria"]
5	["Rust", "Left 4 Dead 2", "Terraria", "Hollow Knight", "ARK: Survival Evolved", "Garry's Mod", "Stardew Valley", "Among Us", "Cyberpunk 2077"]
…	…
…	…
…	…
296	["Cyberpunk 2077", "ARK: Survival Evolved", "Terraria", "Rainbow Six Siege", "The Witcher 3"]
297	["Fall Guys", "Rust", "Left 4 Dead 2", "Cyberpunk 2077", "Portal 2", "Apex Legends", "Hollow Knight", "Terraria"]
298	["Fall Guys", "PUBG", "Counter-Strike", "Portal 2", "Left 4 Dead 2"]
299	["Hollow Knight", "Phasmophobia", "ARK: Survival Evolved", "PUBG", "Portal 2", "Apex Legends", "Rust"]
300	["PUBG", "Call of Duty: Modern Warfare", "Dota 2", "Terraria", "Among Us", "Counter-Strike", "Team Fortress 2"]

3.2. Step 2: Data Preparation

The transactions were carefully processed to eliminate duplicate entries within individual game libraries to maintain accuracy. After cleaning, the dataset was restructured into a format suitable for analysis with the Apriori algorithm, where each user's game library represented a unique transaction.

The preparation phase also ensured the dataset adhered to the requirements of the algorithm by standardizing game names and reviewing the consistency of entries. This step was critical in enabling accurate identification of frequent game combinations and the effective generation of association rules in later stages.

3.3. Step 3: Applying the Apriori Algorithm

The Apriori algorithm was used to mine frequent itemsets and generate association rules. The following parameters were configured:

- Minimum Support: 0.01
- Minimum Confidence: 0.4
- Minimum Lift: 1.2

The algorithm examined the dataset to identify game combinations that met the specified thresholds. It first analyzed individual games, calculating their support values and filtering out those below the minimum support threshold. It then combined the remaining games into larger itemsets, repeating the process. From these frequent itemsets, association rules were generated, such as:

"If a user owns Counter-Strike and Dota 2, they are likely to own PUBG."

These rules helped identify relationships between games that users are likely to play together, allowing the recommendation system to suggest relevant games based on users' existing preferences, thus improving their experience on the platform.

3.4. Step 4: Frequent Itemset Analysis

Frequent game combinations were identified by analyzing the results of the Apriori algorithm. The algorithm examined various game pairings and groupings to determine which combinations appeared most often across users' libraries.

3.5. Step 5: Visualization

The top 10 frequent game combinations were visualized using a horizontal bar chart to display their support values. This visualization helped identify the most popular game combinations and provided insights into user behavior patterns on the Steam platform. By highlighting the most commonly played games together, the chart revealed trends in user preferences, offering a clearer understanding of which games are often enjoyed in combination. This visualization played a key role in recognizing patterns that could inform more personalized game recommendations.

3.6. Step 6: Game Recommendation Rules

The top 10 frequent game combinations were visualized using a horizontal bar chart to display their support values. This visualization helped identify the most popular game combinations and provided insights into user behavior patterns on the Steam platform. By highlighting the most commonly played games together, the chart revealed trends in user preferences, offering a clearer understanding of which games are often enjoyed in combination. This visualization played a key role in recognizing patterns that could inform more personalized game recommendations.

Example rule:

- If a user owns "Counter-Strike" and "Dota 2," recommend "PUBG."
- Support: 0.42, Confidence: 0.56, Lift: 1.35

4. Results & Challenges

The results validated the effectiveness of the Apriori algorithm in uncovering meaningful patterns in user behavior. The association rules provided actionable recommendations, such as suggesting multiplayer games based on similar preferences. The visualization of the top 10 frequent combinations highlighted popular games and game combinations, offering valuable insights into gaming trends.

Despite the successful implementation, challenges such as computational intensity for large datasets and the careful selection of thresholds remained. However, these challenges were mitigated by using a manageable dataset and fine-tuning the algorithm parameters.

5. Code Implementation (Jupyter Notebook)

```python
# Required Libraries
import pandas as pd
import matplotlib.pyplot as plt
from apyori import apriori
import json


# Step 1: Load Data
with open('data/steam.json', 'r') as file:
    transactions = json.load(file)
```

```python
# Step 2: Remove Duplicates Within Transactions
transactions = [list(set(transaction)) for transaction in transactions]


# Step 3: Display the Transactions (Optional)
print(f"======> Total Users: {len(transactions)} <======")
for transaction in transactions:
    print(transaction)


# Step 4: Generate Frequent Itemsets Using Apriori
min_confidence = 0.4
min_lift = 1.2
min_support = 0.01

# Apply the apriori algorithm on the transaction list
results = list(apriori(transactions, min_support=min_support, min_confidence=min_confidence,
min_lift=min_lift))


# Step 5: Display Graphic of the Top 10 Frequent Game Combinations
# Sort by 'Support' and get the top 10 frequent game combinations
results_df = pd.DataFrame([(tuple(result.items), result.support)
                           for result in results],
                          columns=['Itemset', 'Support'])

# Convert itemsets to string format for easier visualization
results_df['Itemset_str'] = results_df['Itemset'].apply(lambda x: ', '.join(x))

# Sort by 'Support' and get the top 10 frequent game combinations
top_10_results = results_df.sort_values(by='Support', ascending=False).head(10)

# Create a horizontal bar plot
plt.figure(figsize=(12, 6))
plt.barh(top_10_results['Itemset_str'], top_10_results['Support'], color=plt.cm.plasma(range(256)))

# Title and labels
plt.title('Top 10 Frequent Game Combinations and Their Support')
plt.xlabel('Support')
plt.ylabel('Game Combination')

# Invert y-axis to have the most frequent combination on top
plt.gca().invert_yaxis()

# Display the plot
plt.tight_layout()
plt.show()


# Step 6: Generate Association Rules for Game Recommendations
# Now, we extract the rules and provide recommendations based on the games a user plays
print("=====> Game Recommendation Rules <=====")
for result in results:
    items = list(result.items)
    support = result.support
    for ordered_stat in result.ordered_statistics:
        antecedent = list(ordered_stat.items_base)
        consequent = list(ordered_stat.items_add)
        confidence = ordered_stat.confidence
        lift = ordered_stat.lift
        if antecedent and consequent:
            print(f"Rule: {antecedent} -> {consequent}")
            print(f"Support: {support:.4f}\nConfidence: {confidence:.4f}\nLift: {lift:.4f}")
            print("-" * 30)
```

6. Conclusion & Future Work

The Steam Game Recommendation System successfully applied the Apriori algorithm to mine frequent game combinations and generate personalized game recommendations. The system demonstrated the potential of association rule mining in enhancing user experiences on platforms like Steam.

In the future, the system could be improved by using a larger, real-time dataset and experimenting with alternative algorithms, such as collaborative filtering or deep learning techniques, for better recommendation accuracy.

7. References

Agrawal, R., & Srikant, R. (1994). Fast algorithms for mining association rules. *Proceedings of the 20th International Conference on Very Large Data Bases (VLDB),* 487-499.

Tan, P.-N., Steinbach, M., & Kumar, V. (2005). Chapter 5: Association Analysis: Basic Concepts and Algorithms. *In Introduction to Data Mining.* (pp. 359-500). Pearson Education.