Apriori Algorithm in The Context of Movie Recommendation Systems

Fedor Khomchenko

Received August 05, 2024
Accepted March 30, 2025
Electronic access April 30, 2025

Streaming services increasingly depend on effective recommendation systems to boost user engagement and profitability. This paper investigates the adaptation of the Apriori algorithm—traditionally used in market basket analysis for e-commerce—to the domain of movie recommendations. Utilizing the MovieLens 20M dataset, user ratings are transformed into transactional "baskets" by selecting movies rated 4 or above, thereby treating positive ratings as implicit purchases. The Apriori algorithm is then applied with a minimum support threshold of 30%, a maximal rule length of 2, and a confidence level of 100% to generate association rules linking movies. Experimental results reveal that while the approach successfully generates rules that connect popular films with niche titles, a majority of the rules exhibit a lift value of 1, indicating statistical independence between the associated movies. Furthermore, the study discusses challenges such as rule explosions and reviewer bias, which affect the overall predictive power and computational efficiency. Despite these limitations, the method consistently produces relevant associations, underscoring its potential in recommendation systems. The paper concludes by suggesting that integrating hybrid approaches and advanced data preprocessing techniques could mitigate current shortcomings and further enhance recommendation quality.

Keywords: Apriori algorithm, movie recommendations, recommender systems, MovieLens 20M

Introduction

In recent years streaming services have experienced a big rise in popularity due to the pandemic and a variety of other reasons. That, in turn, has led to an increased demand for accurate movie recommendation systems. Accurate recommendation systems are able to both improve user experience and business profits by suggesting movies that the user will want to watch. For these reasons there is an influx in works exploring different approaches to creating an efficient movie recommendation system.

Of course, streaming services are not the only ones to utilize recommendation systems. Other huge industries such as e-commerce and retail rely on recommendation systems to increase their sales. We believe that we can apply common practices used in creation of different types of recommendation systems to creating a movie recommendation system. To be more specific we are going to use the Apriori algorithm and Market basket analysis, which are most commonly used in creating recommendation systems for e-commerce.

Since the Apriori algorithm is not usually used in the context of movie recommendations. Therefore we cannot simply apply Apriori in the same way we would in e-commerce or any similar field and must come up with a way that would produce meaningful results.

Thus, the question we are going to be answering is: "What are the advantages and disadvantages of implementing market basket analysis and the Apriori algorithm to create an effective movie recommendation system?"

In this paper we will propose and test a way to apply Apriori Algorithm to create a movie recommendation system. By doing so we hope to advance the understanding of versatility of the Apriori algorithm and its usefulness in creating movie recommendation systems.

Literature Review

The Apriori algorithm, introduced by Agrawal and Srikant (1994)[1], is a foundational method for association rule mining in transactional databases. It identifies frequent itemsets and derives association rules that satisfy predefined support and confidence thresholds, enabling the discovery of co-occurrence patterns in large datasets. The algorithm operates iteratively using a breadth-first search strategy, leveraging the downward closure property (also known as the Apriori principle), which posits that all subsets of a frequent itemset must themselves be frequent (Han et al., 2011)[2]. This property allows the algorithm to prune the search space efficiently by eliminating candidate itemsets that cannot meet the minimum support threshold, thereby reducing computational complexity.

The Apriori algorithm has been widely adopted in recommendation systems due to its ability to identify frequent itemsets and generate meaningful association rules. This literature review explores the application of the Apriori algorithm in recommendation systems, focusing on its strengths, limitations, and hybrid approaches that enhance its effectiveness.

The Apriori Algorithm in Recommendation Systems

The Apriori algorithm is primarily used to uncover patterns in transactional data, making it particularly effective for market basket analysis and product recommendation systems. By identifying frequently co-occurring items, the algorithm generates association rules that can predict user preferences based on historical data. For instance, Talwar et al. (2015)[3] demonstrated the use of the Apriori algorithm in recommending products by analyzing frequent itemsets and association rules, achieving precise recommendations with a confidence value of 76.92% (Talwar et al., 2015)[3]. Similarly, a hybrid recommender system proposed in 2019 combined the Apriori algorithm with collaborative filtering to address the shortcomings of traditional recommendation methods, emphasizing its utility in handling large information catalogues and user preferences (Gupta et al., 2019)[4].

Hybrid Approaches and Enhancements

To overcome the limitations of the Apriori algorithm, such as high computational complexity and memory consumption, researchers have proposed hybrid and enhanced models. For example, Song and He (2023)[5] developed an intelligent tourism recommendation system that integrated the Apriori algorithm with artificial intelligence (AI) and the Internet of Things (IoT) to provide personalized travel itineraries. This system achieved a 94.3% improvement in accuracy over traditional recommendation algorithms, showcasing the potential of combining the Apriori algorithm with modern technologies (Song & He, 2023)[5]. Additionally, an improved version of the Apriori algorithm reduced computational time by 67.38% by scanning only a subset of transactions, making it more efficient for large datasets (Liu et al., 2023)[6].

Applications in Diverse Domains

The versatility of the Apriori algorithm is evident in its application across various domains. In e-commerce, it has been used to recommend products based on user purchase history, significantly enhancing user engagement and satisfaction (Talwar et al., 2015)[3]. In the context of college libraries, an Apriori-based recommendation system was designed to suggest books to students, leveraging transaction data to identify reading patterns (Xueyuan & Bo, 2018)[7]. Furthermore, the algorithm has been applied in news recommendation systems, where it helps filter and prioritize content based on user preferences and association rules (Atmadja et al., 2024)[8].

Challenges and Future Directions

Despite its widespread adoption, the Apriori algorithm faces challenges such as scalability issues with large datasets and the need for frequent database scans. Researchers have proposed solutions like parallel processing and integration with other machine learning techniques to mitigate these limitations (Bhareti et al., 2020)[9]. Future research could explore the integration of the Apriori algorithm with deep learning models to further enhance recommendation accuracy and scalability.

Conclusion

The Apriori algorithm remains a powerful tool for developing recommendation systems, particularly in domains requiring frequent itemset mining and association rule generation. While challenges such as computational complexity persist, hybrid approaches and technological advancements continue to expand its applicability and effectiveness.

Dataset

In our experiment we are going to use a dataset provided by MovieLens (GroupLens, 2016)[10]. This section's aim is to provide an overview of the dataset, as well as to point out some patterns and relations in data. The tables that we will utilise in our experiment are "rating.csv" and "movie.csv" which we will use to generate "baskets". We believe that the understanding of the dataset is important for interpreting the results of the experiment.

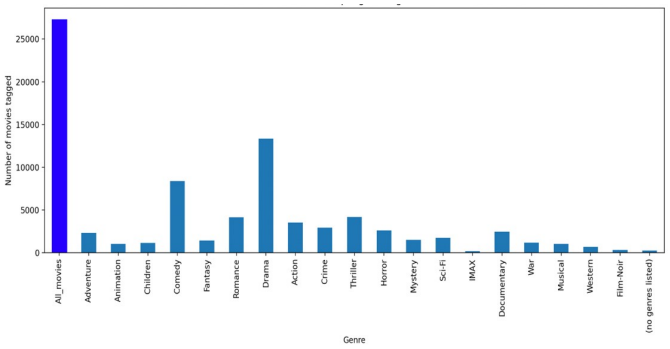

Fig. 1 (Genre distribution in "movie.csv" table)

The "movie.csv" table has information about the movies that are referenced within the dataset including titles, release dates (included in the title field), movieIds (internal dataset identifier), marked genres. We have explored this table for genre distribution and found that the most prevalent genres are drama and comedy.

Table "rating.csv" contains records of reviews left by the users. Data includes userIds (internal user Identifier), movieIds, ratings (on a 0 to 5 scale) and timestamps. This table is of the most interest to us as this is the data we will extract users "baskets" from.

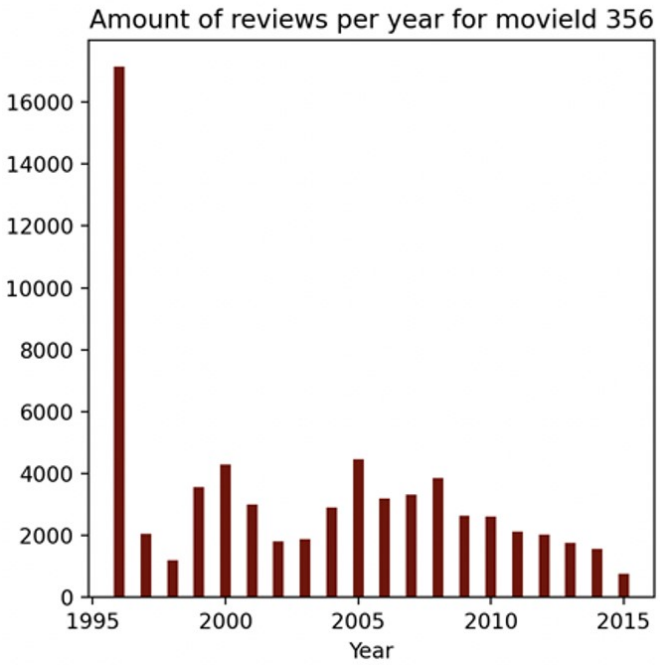

Fig. 2 (Distribution of reviews over the years for two films)

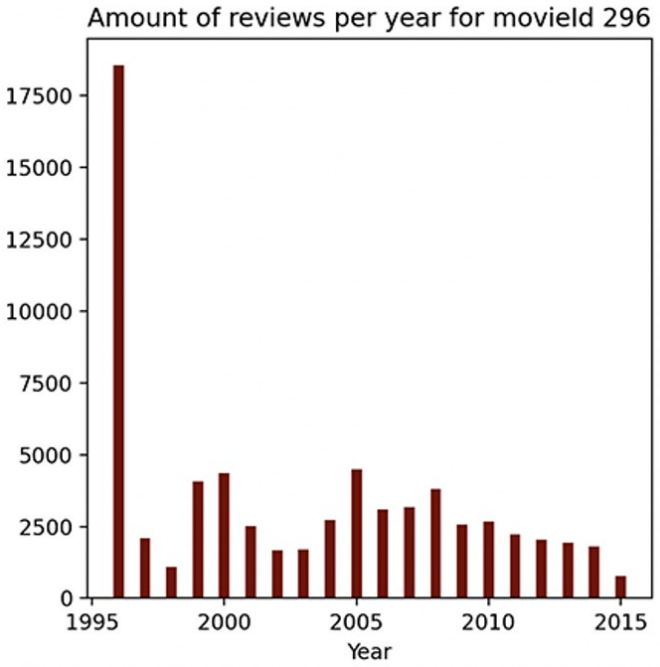

Fig. 3 (Distribution of reviews over the years for two films)

An interesting pattern that we have uncovered is the distribution of reviews in the premiere year vs post-primer years.

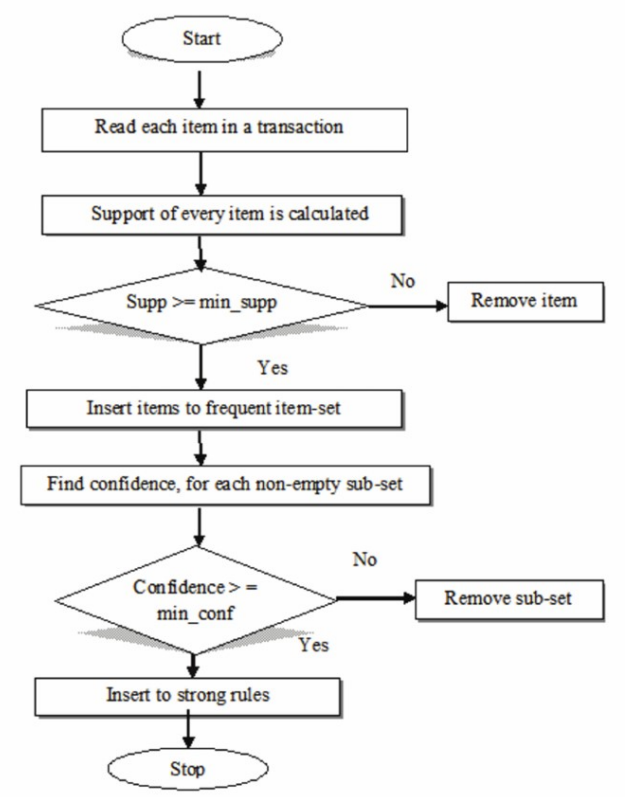

Fig. 4 Apriori algorithm flowchart (Mittal, 2014) [11]

To provide some evidence we present the distribution for top 2 movies with highest rating numbers. We find these examples representative of the overall trend present in films from the dataset. We can see most reviews were left on the premiere year. The distribution is also very similar between the two examples, although the values differ.

Methodology

1. The Apriori algorithm, introduced by Agrawal and Srikant (1994) [1], is a foundational method for association rule mining. It identifies frequent itemsets (groups of items that co-occur) and derives rules that capture relationships between variables (association rules), which could be formally expressed as an implication of the form $X \geq Y$, where X and Y are disjoint itemsets. For example "customers who buy X also buy Y." The algorithm relies on the Apriori property: All subsets of a frequent itemset must also be frequent.

   Let us explore the algorithmic workflow (Figure 4) of the Apriori algorithm.

   (a) Itemsets of length 1 are generated and support is calculated for each one. Support as defined on Figure

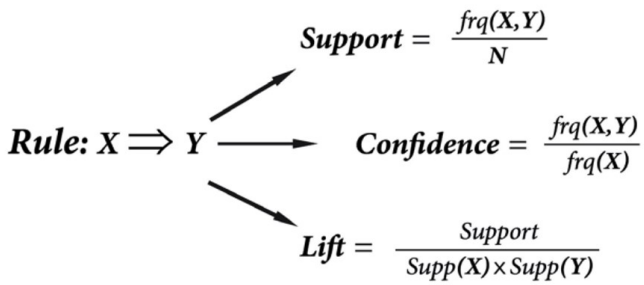

Fig. 5 Association rule terminology definitions (Tangpong et al., 2021)

5 is a measure of frequency of the itemset. Apriori algorithm discards all itemsets that do not meet the minimum support.

(b) After infrequent itemsets are discarded, the rest are gathered into an itemset. For each subset present in this newly created itemset confidence value is found. Confidence as defined on Figure 5 is a measure of the reliability of an association rule X $\geq$ Y, where a larger percentage of itemsets containing both X and Y as opposed to itemsets containing only X signals higher reliability, while the opposite signals lower reliability. In our experiment a variable "maximal length" is used. This variable regulates the maximal size of rules that are generated. Where maximal length can be calculated as the length of itemset X added to the length of itemset Y in a rule of form X $\geq$ Y. Alternatively it can be seen as the length of the subset from which the rule was found. When the itemset of frequent items is sufficiently large, it is pragmatic to limit the maximal size of subsets in order to decrease computation time.

(c) Each subset that meets confidence is designated as a strong rule and outputted. An additional metric is usually recorded called lift. Lift as defined on Figure 5 is a measure of dependence between X and Y. Lift $>$ 1 implies positive correlation, lift = 1 implies independence and lift $<$ 1 pimples negative correlation.

2. Market basket analysis is a data mining technique used by retailers to increase sales by better understanding customer purchasing patterns. It involves analyzing large data sets, such as purchase history, to reveal product groupings, as well as products that are likely to be purchased together.

   In our experiment we are going to approach creating a movie recommendation system from a commercial perspective using Apriori algorithm and market basket analysis. We are going to perceive a user giving a movie a good rating as a customer buying a product. This approach limits our investigation as a single user can only review a movie once, however the nature of film consumption is different to the consumption of other goods like groceries. Most people do not consume the same films daily, but rather watch them once and then move on to consume new films (Motion Picture Association, 2020)[12] Because of this we cannot fully simulate the retail conditions using only review data. Therefore we consider not a single reviewer, but rather all the reviewers as a collective to be the subject, where an individual user is a single "purchase". To achieve that, we find a list of movies that a user rated above a certain threshold for every user, this list is treated as the users' "basket". After that, we run this data through the Apriori algorithm to get association rules between movies, from these rules we will be able to create recommendations. There are some apparent issues with this approach. One of them is that we cannot reliably address the bias of reviewers. Bias can be film-specific or more general. For example some users might like or dislike certain franchises, like Star Wars fans rating movies belonging to the franchise unreasonably high or low. Rating scales in general are rarely completely consistent between reviewers, with some people having higher expectations, while others hold films to less scrutiny. To reliably determine bias and account for it sentiment analysis could be employed, but this problem is beyond the scope of this paper.

Experiment Design

In order to create relevant rules, we propose that we only consider preferences of users that have rated a certain movie that we are interested in generating recommendations for highly. Our program takes in the dataset of "transactions", confidence and support threshold and outputs rules generated by applying the thresholds to the dataset. The Apriori algorithm and data cleaning tools were all created using Python.

Data Pre-processing

We begin by identifying the film we will base our set of rules from. First, we want to get a list of users that have rated our movie above a certain threshold. Since the Apriori algorithm is used primarily for e-commerce we need to adapt our data to look like a set of transactions. So as the next step form a list of all movies that our users "bought" in order to form a "basket" for every user. Once we have all the baskets we can proceed further.

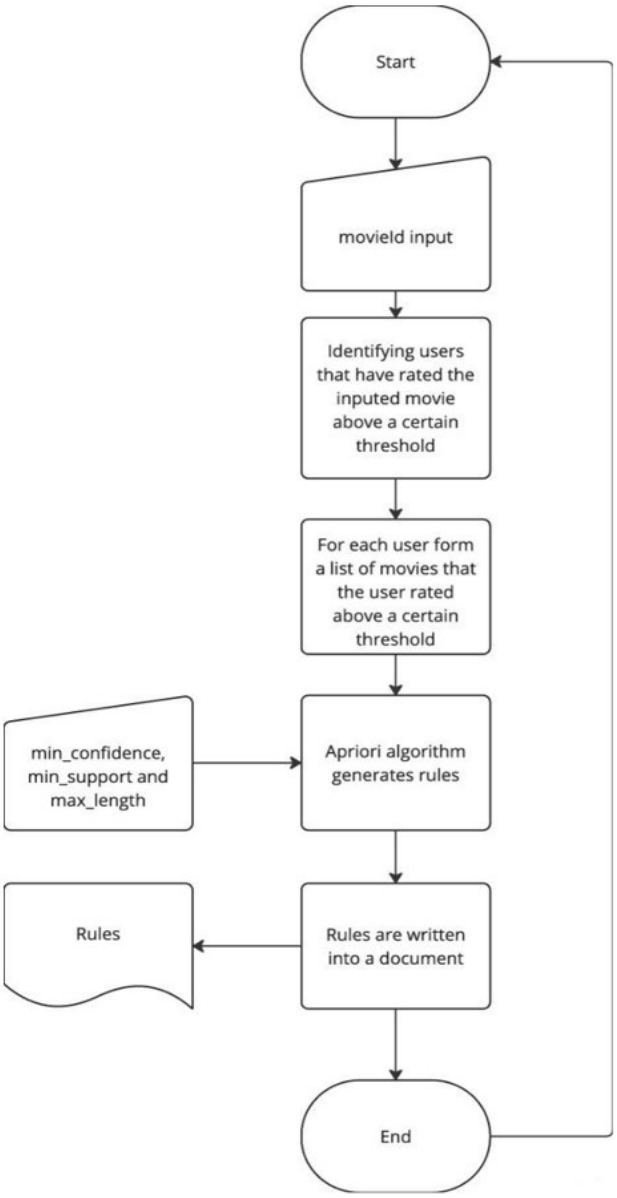

Fig. 6 Flowchart of the experiment

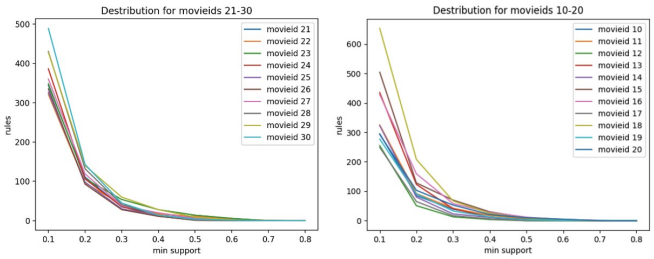

Fig. 7 Graph of parameter selection trials for movies with movieIds from 10 to 30

Results

Parameter Selection

After analysing the size of "baskets" that come out of the dataset we have found that the average size of one "basket" is approximately 72 items. Considering that multiple "baskets" will be drawn each time, the number of subsets to be explored in each pass of Apriori rises dramatically when we increase the maximal length of the subsets explored. Due to the resource constraints we are faced with, it is practical to keep the maximal length of the subsets at 2, to preserve computational efficiency.

When picking the rating threshold, we wanted to strike a balance between the amount of data points that are let through and be as close to 5 as possible. It has been observed that the MovieLens dataset that we are using is skewed towards the rating of 4 (Ghazanfar et al., 2013)[13]. Meaning that setting the threshold at the rating of 4 will ensure that a significant amount of data points will be let through. Thus, we set our rating threshold at 4, since raising it any higher would most likely impede the generation of rules.

Due to how we pick our "baskets" if we set the minimum confidence threshold to 100, then we are guaranteed to generate rules that relate the movieId that we are currently exploring to some other movieIds if they exist, since the aforementioned movieId must be a part of the intersection of all relevant baskets. Rules that relate items from the intersection of all relevant baskets to any other item will necessarily be supported by a 100

Therefore, the parameter that we should mainly manipulate in our experiment is minimum support. To find the optimal value we conduct trials where we run the experiment with different minimum support levels and record the amount of rules. Our goal is to find a minimum support level that is both as close to 100% as possible and steadily generates more than 0 rules.

As we can see on Figure 7 the number of rules decreases dramatically as we increase the minimum support levels. We can observe that at around 40 or 50 percent a significant portion of graphs reach 0. Similar trend can be observed throughout the majority of the dataset. Therefore, for this experiment we will use the minimum support level of 30%, to ensure that in most

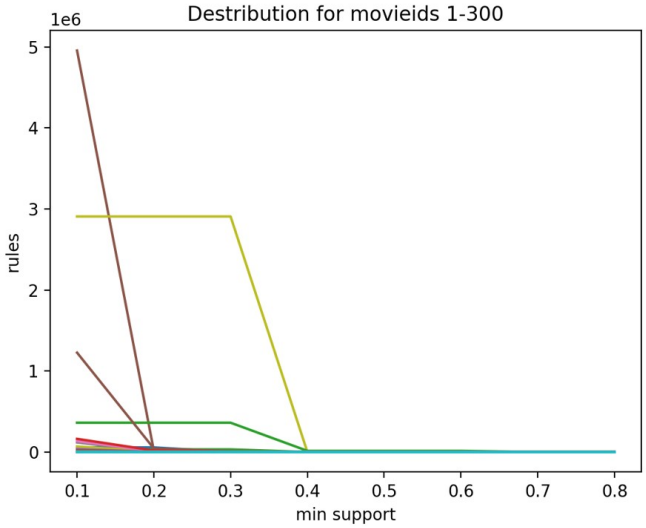

Fig. 8 Graph of parameter selection trials for movies with movieIds from 1 to 300

cases at least some rules are guaranteed to be generated.

While conducting the trials we came across some anomalies that can be seen on Figure 8. Certain graphs generated significantly more rules and did not follow the general trend when increasing the minimum support level. We have found that these "explosions" in the amount of rules occur when a certain film has a relatively low amount of relevant "baskets". Most notably films with movieIds 56, 109, 130, 133, 139, 142, 143, 226, 286 (the most noticeable explosions on the graph) all had the amount of relevant "baskets" in the range from 3 to 17. It was also observed that the intersections between the relevant baskets were extremely small, mostly in the range from 1 to 7, with one outlier which had an intersection of length 27 (movieId 143). The combination of these factors leads to almost every rule possible being boosted on lower minimum support thresholds, while the amount of meaningful rules is very limited, hence the steep drop at a certain point (the less total relevant baskets, the later the is the drop).

Results

The parameters we use to gather the following results are: minimum confidence - 100%, minimum support - 30%, maximal length - 2.

By running the experiment for movies with movieIds from 1 to 300 we have observed that rules generated can be split into 3 general groups based on the amount of rules generated: no rules generated, moderate amount of rules generated and excessive amount (more than 500) of rules generated.

From the sample of 300 only 2 runs did not yield any rules, which is an acceptably low rate. Similarly, only 12 out of 300 runs generated an excessive amount of rules and we have already covered why they appear in the previous section.

The remaining 284 of the runs have produced between 1 and 499 rules. To understand the nature of the rules generated we will explore several data points that represent the general trends.

Exploring data points

movieId 33

Film with movieId 33 is the Wings of Courage (1995), which is not a very popular or highly acclaimed movie. Out of 331 rules generated, 116 include the film itself. Other rules relate popular movies to each other, for example Aladdin (1992) to Toy Story (1995). This shows that our approach successfully generates rules for less famous movies, however rules that relate more popular movies take up a significant portion of total rules generated.

movieId 12

Film with movieId 12 is Dracula: Dead and Loving It (1995), which is a relatively popular movie with mediocre ratings. All 13 of the generated rules relate Dracula: Dead and Loving It to popular movies such as Toy Story, Forrest Gump and Jurassic Park. We can see that our approach can generate relevant rules for less popular movies.

movieId 1

Film with movieId 1 is Toy Story (1995), which is a very popular and highly acclaimed movie. All 32 of the generated rules relate Toy Story to other popular movies such as Apollo 13, Lion King and Godfather. Our approach works well on popular movies, generating relevant rules.

General trends

It is worth noting that absolute majority rules that relate the movieId we are interested in had lift = 1, implying independence of the movies involved.

Analysis

The rules generated are dominated by popular franchises and films such as "Star Wars", "Toy Story", "Jurassic Park" and "Lion King". Not only are most rules related to them, but sometimes rules relating popular movies appear in seemingly unrelated to these movies sets of rules. This behaviour is expected, as apriori algorithm strives to find the strongest connections and popular movies are the items that appear more frequently in the dataset than other movies. Using our approach we have

consistently generated strong rules that connect almost any film to a number of popular movies.

Independence of variables implied by the value of lift is expected within the context of our application, as an increase in the number of positive reviews of one film should not lead to an increase in the number of positive reviews of some other film. Additionally, this value of lift can be explained by our approach to picking "baskets". Since every "basket" contains X (in a rule of form $X \geq Y$), then it is logical to assume that every "basket" that contains X also contains Y. Therefore, the frequency of X and Y can be equated to just the frequency of Y, and since the frequency of X is 100%, then, by definition, when calculating lift we divide the frequency of Y by itself. Association rules with lift of 1 have no predictive power and therefore are often discarded when encountered in the context of retail for example. Within the context of our experiment, where predictive power of rules matters little, as predictions are oftentimes generated in real time, rules with lift value of 1 are still valuable to us and should not be discarded. These results indicate the usefulness of the proposed algorithm in identifying popular movies and generating rules that connect appropriate popular films to the majority of films in the database. From these rules recommendations could be formed.

Results and Code Availability

All the code used in this study is openly available to the scientific community. The scripts written for this research can be accessed freely at https://github.com/NeoMCHS/AprioriMovieResearch. We encourage fellow researchers to take advantage of this availability to reproduce our findings and facilitate further investigations in this field.

Conclusion

In this study we have looked into the advantages and disadvantages of implementing market basket analysis and the Apriori algorithm to create an effective movie recommendation system. We have proposed an approach to applying Apriori algorithm to generate rules, upon which movie-specific recommendations could be made. We have found the hyperparameters best suited for our approach and using them have found that our approach succeeds in finding popular movies within the dataset and relating them to any film.

Our findings suggest that the proposed approach is successful in generating rules that could be used to provide movie recommendations. Thus, our study provides an approach that other researchers interested in using Apriori algorithm to create movie recommendations could evaluate and improve on further.

We have encountered and outlined several limitations of our approach: rule explosions, the underlying problem of reviewer bias, computational difficulty of our method. No in- depth data pre-processing and exploration was performed, which could mean that certain undetected biases or inconsistencies in data affected the results.

To future researchers we recommend adopting a hybrid approach, fusing apriori algorithm with other technologies such as artificial intelligence, or other algorithms, to increase computational efficiency of the approach and allow for larger rules to be generated. Additionally, we recommend future researchers explore ways of normalizing the amount of rules generated for any given movie.

Acknowledgment

Thank you for the guidance of Dilina Dehigama Mentor from University of Edinburgh in the development of this research paper.

References

1 R. Agrawal and R. Srikant, *Fast algorithms for mining association rules*.

2 J. Han, J. Pei and M. Kamber, *Data mining: Concepts and techniques*.

3 K. Talwar, A. Oraganti, N. Mahajan and P. Narsale, *Recommendation system using Apriori algorithm*, https://www.semanticscholar.org/paper/Recommendation-System-Using-Apriori-Algorithm-Talwar-Oraganti/ea9c6f9ec8421ef536b9171171a1aaf2cb9e3fde.

4 M. Gupta, S. Kochhar, P. Jain and P. Nagrath, *Hybrid recommender system using A-priori algorithm*, https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3349290.

5 Y. Song and Y. He, *Toward an intelligent tourism recommendation system based on artificial intelligence and IoT using Apriori algorithm*, https://doi.org/10.1007/s00500-023-09330-2.

6 H. Liu, A. Liu, R. Wang and J. Tang, *Analysis of Apriori algorithm and its application*, https://ieeexplore.ieee.org/document/10339418.

7 W. Xueyuan and Y. Bo, *Design and implementation of an Apriori-based recommendation system for college libraries*, https://ieeexplore.ieee.org/document/8530437.

8 A. Atmadja, S. Rahmawati, Y. Gerhana, M. Firdaus and I. Budiman, *News recommendation system using Apriori algorithm and association rules*, https://ieeexplore.ieee.org/document/10775302.

9 K. Bhareti, S. Perera, S. Jamal, M. Pallege, V. Akash and S. Wieweera, *A literature review of recommendation systems*, https://ieeexplore.ieee.org/document/9298450.

10 GroupLens, *MovieLens 20M dataset*, https://www.kaggle.com/datasets/grouplens/movielens-20m-dataset?select=movie.csv.

11 M. Mittal, *Efficient ordering policy for imperfect quality items using association rule mining*, https://doi.org/10.4018/978-1-4666-5888-2.ch074.

12 M. P. Association, https://www.motionpictures.org/research/theme-report/.

13 M. Ghazanfar, A. Ghazanfar and A. Prugel-Bennett, *The advantage of careful imputation sources in sparse data-environment of recommender systems: Generating improved SVD-based recommendations*.