

Utilizing Frequent Pattern Mining for Solving Cold-Start Problem in Recommender Systems

Eyad Kannout*, Michał Grodzki[†] and Marek Grzegorowski[‡]

Institute of Informatics, University of Warsaw

Banacha 2, Warsaw, Poland

Email: *eyad.kannout@mimuw.edu.pl, [†]m.grodzki@students.mimuw.edu.pl, [‡]m.grzegorowski@mimuw.edu.pl

Abstract—Although several approaches have been proposed throughout the last decade to build recommender systems (RS), most of them suffer from the cold-start problem. This problem occurs when a new item hits the system or a new user signs up. It is generally recognized that the ability to handle cold users and items is one of the key success factors of any new recommender algorithm. This paper introduces a frequent pattern mining framework for recommender systems (FPRS) - a novel approach to address this challenging task. FPRS is a hybrid RS that incorporates collaborative and content-based recommendation algorithms and employs a frequent pattern (FP) growth algorithm. The article proposes several strategies to combine the generated frequent itemsets with content-based methods to mitigate the cold-start problem for both new users and new items. The performed empirical evaluation confirmed its usefulness. Furthermore, the developed solution can be easily combined with any other approach to build a recommender system and can be further extended to make up a complete and standalone RS.

Index Terms—recommendation system, cold-start problem, frequent pattern mining, quality of recommendations.

I. INTRODUCTION

OVER the past few decades, alongside the explosion in the amount of data on the internet, the popularity of online streaming services, e-commerce, and social media has highlighted an important challenge to provide users with recommendations that match their preferences and interests. Therefore, the demand for finding more efficient techniques to generate recommendations has received more attention. Over the years, researchers have suggested various approaches for building recommender systems that leverage the rating history and possibly some other information, such as users' demographics and items' characteristics. The majority of these approaches can be classified into three main categories: (i) collaborative filtering (CF), (ii) content-based (CB) filtering, and (iii) hybrid filtering.

The basic idea behind collaborative filtering is that users with similar tastes or preferences tend to behave similarly in the future. This technique relies on historical transactions to compute similarities among users from which the recommendations are eventually generated. An analogous approach can be applied to create recommendations based on item similarities. On the other hand, content-based filtering tries to utilize items' characteristics, users' demographics, and

contextual information to recommend additional items similar to those preferred by the target user in the past. Finally, hybrid techniques cover the weaknesses and exploit the strengths of CF and CB models by combining them to provide more relevant results.

The aforementioned techniques are highly appreciated by practitioners and businesses. However, they also encounter significant difficulties in terms of data characteristics. One of the issues is related to the sparsity of data. The discussed methods rely on modeling the user-item interactions, and hence, the quality of such may be impacted by an insufficient number of movies rated by each user. Another challenge is related to the so-called cold-start problem. This phenomenon is particularly inconvenient and occurs whenever recommendations are generated for a new item or user that does not have any interaction or rating in the history. In fact, many state-of-the-art recommendation algorithms may generate unreliable recommendations for such cases since they cannot learn the preference embedding of these new users/items [1], [2], [3].

In this study, we presented a particular take on the challenge of devising more effective and efficient recommendation techniques. We put special attention to properly handling the new users and items. We propose several methods to overcome the cold-start problem and the sparsity nature of the datasets by utilizing the FP-growth algorithm to generate frequent patterns based on items' characteristics and users' demographics. The main contributions of this paper are as follows:

- 1) We introduce *frequent pattern mining framework for recommender systems* (FPRS) - a novel hybrid recommender system that utilizes the FP-growth algorithm to produce frequent itemsets based on the ratings in the user-item matrix.
- 2) We utilize the items' and users' features to extract particular patterns based on the features selected.
- 3) We propose several techniques to mitigate the cold-start problem by using the discovered patterns to provide recommendations for new users and involve new items into the recommendations generated for the users.
- 4) We conduct the empirical evaluation of the proposed approach on well established benchmark data (MovieLens 100K and MovieLens 1M), showing its effectiveness in the presence of new entities (users and items, i.e., movies) in test data.

Research co-funded by Polish National Science Centre (NCN) grant no. 2018/31/N/ST6/00610.

The remainder of this paper is organized as follows. Section II describes and reviews important research efforts that addressing the cold-start problem in the domain of Recommender systems. In Section III, we provide background information for collaborative filtering and frequent pattern mining. In Section IV, we present a novel frequent pattern mining model (FPRS) that utilizes the ratings in user-item rating matrix to discover the frequent itemsets associated with selected users/items features. Section V evaluates and compares the proposed model with a baseline recommender system. Finally, in Section VI, we draw conclusions and suggest possible future work.

II. RELATED WORKS

Recommender systems (RS) predict the utility of an item to a user and suggest the best items concerning the user's preferences, where the items may represent movies, books, restaurants, or any other things [4], [5]. The aforementioned capability of RSs makes those techniques especially useful, and indeed, there are many areas of their successful applications like eCommerce, online marketing, or social networks [6], [7]. The scientific literature provides several taxonomies for RS [8]. However, the most common approaches refer to content-based or collaboration-based techniques and their various hybridizations [9]. Collaborative Filtering (CF) is one of the most widely used and successful techniques, with excellent results in a wide range of applications in many fields [8], hence is particularly interesting in our research and further reviewed in detail in Section III-A. Despite the noticeable decline in their popularity in favor of collaborative systems, content-based techniques are still widely used because of handling the so-called cold-start problem [10]. Because of the significantly different characteristics of those approaches, it is advisable to construct hybridizations of both [11], as further discussed in our study.

A typical RS consists of the three main elements: a user model (established by analyzing the users' interests and preferences), an item model (based on its characteristics), and the recommendation algorithm that is a key constituent. There are many reported approaches to implementing the recommendation algorithm by the specific adoption of machine learning (ML) models like matrix factorization, deep neural networks, or factorization machines (FM) [12], [13], [14]. Building RS on top of the state-of-the-art ML models leveraged the quality of recommendation results, improving user satisfaction and profits in e-commerce [15], [6], [16]. At the same time, however, we may observe the known problems with ML related to the data sparsity, the latency of prediction returned by complex models, and foremost, the unfairness of recommendations for new users or items that is often referred to as the *cold-start* problem [17], [18].

Solving scalability issues is one of the most common tasks when deploying big-scale recommender systems. Especially as the number of users and items significantly grows over time, it is essential for RSs to handle requests without appreciable

latency. This problem is particularly challenging for memory-based methods like k-nearest neighbors. However, in the case of web-scale recommendation tasks like social media, the Internet of Things (IoT), or various e-commerce applications, it is a hot topic also for model-based techniques, especially considering more complex and deep models [1], [19]. Another aspect that is particularly noticeable for collaborative filtering is related to the sparsity of user-item interactions [20]. Here, the quality of CF-based methods may be impacted by an insufficient number of items rated by each user [21]. Some recommender systems suffer from their over-specialization (sometimes referred to as a serendipity problem). It is observed when the RS produces recommendations with minimal novelty, i.e., all of the same kind [22]. Recently, there is also an increasing interest in privacy awareness when handling user data and explainability of recommendations [23], [24].

Regardless of recent achievements in RS, the cold-start problem is still one of the most prevailing topics deserving further attention and is particularly interesting in the context of our study [3], [21]. The difficulty arises due to the deficient information about new entities. Therefore it has a particularly strong negative impact on collaborative methods, heavily impacting the fairness of recommendations for new users, often passing over new items [18]. Most of the attempts to deal with such a problem consider enhancing the collaborative-based methods with content-based approaches that leverage the intrinsic characteristics of the analyzed entities. For example, in [2], the authors propose hybrid recommender models that use content-based filtering and latent Dirichlet allocation (LDA)-based models. Whereas in [9], we may find a hybrid RS that combines the singular-value decomposition-based collaborative filtering with content-based and fuzzy expert systems.

There are many more techniques to dealing with the cold-start problem by combining collaborative filtering with a content-based methods, including using simultaneous co-clustering [25], self-organizing maps, or Siamese neural networks [3]. There are also some attempts to combine RSs with various dimensionality reduction techniques [26]. Considering the discussed problem of missing or insufficient information, it seems interesting to refer to the dimensionality reduction methods based on the granularization of the attribute space [27], and particularly on resilient techniques [28], [29] - i.e., resistant to data deficiencies. The hybridization of soft computing techniques with collaborative and content-based methods is a wide-ranging field of research and an interesting area for the further development of recommendation systems [30], particularly interesting for context-aware RSs [31], [4].

Some approaches to dealing with cold-start refer to popularity measures, e.g., on the recent trend in users' preferences or always returning the most popular items [14], [10]. However, these may be very misleading and result in so-called popularity bias since users often differ in their preferences, which may also vary between types of products and their characteristics [18]. Hence, an additional effort to deal with biases in data is required [32]. Another interesting approach to dealing with insufficient or missing historical transactions avail additional

sources of information to enhance the data representation. In particular, in [21], the authors train RSs with the Linked Open Data model based on DBpedia to find enough information about new entities. When dealing with the cold-start problem, some researchers rely on directly inquiring the users about their preferences. Such information may be collected, e.g., via survey or by asking users to select the most relevant picture related to the desired item [33]. Combining community-based knowledge with association rule mining to alleviate the cold-start problem is also bringing very promising results [31]. Referring to association rule mining (cf. [34]) and frequent pattern mining (cf. [35]) techniques to address the cold-start problem is interesting also from the perspective of speeding up the recommender systems. For this reason, frequent patterns mining is particularly interesting in our research, and we review this field in detail in Section III-B.

III. PRELIMINARIES

In this section, we briefly summarize the academic knowledge of collaborative filtering and frequent pattern mining techniques. Then, we review some of the research literature related to addressing the cold-start problem.

A. Collaborative Filtering

The basic idea behind collaborative filtering (CF) is that the users who have similar preferences in the past tend to behave similarly in the future. Basically, CF-based methods rely only on users rating history to generate recommendations, meaning that the more ratings the users provide, the more accurate the recommendation become [4]. Usually, the historical ratings or preferences can be acquired explicitly or implicitly. So, the CF-based methods are often distinguished by whether they operate over explicit ratings, where the user explicitly rate particular items, or implicit ratings, where the ratings are inferred from observable user activity, such as products bought, songs heard, visited pages, or any other types of information access patterns [4]. In the literature, collaborative filtering methods can be classified into two main categories: (i) memory-based techniques, and (ii) model-based techniques.

The memory-based technique uses directly the rating history, which is stored in memory, to predict the rating of items that the user has not seen before. However, the memory-based techniques can be grouped into two different classes: (i) user-based collaborative filtering, and (ii) item-based collaborative filtering. The user-based collaborative filtering, also known as k-NN collaborative filtering, works by finding the other users (neighbors) whose historical rating behavior is similar to that of the target user and then using their top-rated products to predict what the target user will like [36]. To mathematically formulate the problem, let us assume there is a list of users $U = \{u_1, u_2, \dots, u_m\}$ and a list of items $I = \{i_1, i_2, \dots, i_n\}$. Then, the user item rating matrix consists of a set of ratings $v_{i,j}$ corresponding to the rating for user i on item j . If I_i is

the set of items on which user i has rated in the past, then we can define the average rating for user i as follows [36]:

$$\bar{v}_i = \frac{1}{|I_i|} \sum_{j \in I_i} v_{i,j} \quad (1)$$

In user-based collaborative filtering, we estimate the rating of item j that has not yet rated by the target user a as follows [36] [37]:

$$p_{a,j} = \bar{v}_a + \frac{\sum_{i=1}^k s(a,i)(v_{i,j} - \bar{v}_i)}{\sum_{i=1}^k |s(a,i)|} \quad (2)$$

where k is the number of most similar users (nearest neighbors) to a . The weights $s(a,i)$ can reflect the degree of similarity between each neighbor i and the target user a . On the other hand, item-based collaborative filtering is just an analogous procedure to the previous method. The similarity scores can also be used to generate predictions using a weighted average, similar to the procedure used in user-based collaborative filtering. Mathematically, we can predict the rating of item j that has not yet been rated by the target user a as follows [36] [37]:

$$p_{a,j} = \frac{\sum_{i=1}^k s(j,i)(v_{a,i})}{\sum_{i=1}^k |s(j,i)|} \quad (3)$$

where k is the number of most similar items (nearest neighbors) to j that the target user a has rated in the past. However, the most popular metrics used to calculate the similarity between different users, or items, are cosine similarity and Pearson correlation. Finally, the recommendations are generated by selecting the candidate items with the highest predictions.

On the other hand, the model-based technique works by learning a predictive model using the rating history. Basically, it is based on matrix factorization which uses the rating history to learn the latent preferences of users and items. Matrix factorization is an unsupervised learning method that is used for dimensionality reduction. One of the most popular techniques applied for dimensionality reduction is Singular Value Decomposition (SVD). Mathematically, let us assume M is the user item rating matrix. The SVD of M is the factorization of M into three constituent matrices such that [37]:

$$M = U\Sigma V^T \quad (4)$$

where U is an orthogonal matrix representing left singular vectors of M . V is an orthogonal matrix representing right singular vectors of M . Σ is a diagonal matrix whose values σ_i are the singular values of M [37].

B. Frequent Pattern Mining

The basic idea of frequent pattern mining, also known as association rule mining, is to search for all relationships between elements in a given massive dataset. It helps us to discover the associations among items using every distinct transaction in large databases. The key difference between association rules mining and collaborative filtering is that in

association rules mining we aim to find global or shared preferences across all users rather than finding an individual's preference like in collaborative filtering-based techniques [38] [39] [40].

At a basic level, association rule mining analyzes the dataset searching for frequent patterns (itemsets) using machine learning models. To define the previous problem mathematically, let $I = \{i_1, i_2, \dots, i_m\}$ be an itemset and let D be a set of transactions where each transaction T is a nonempty itemset such that $T \subseteq I$. An association rule is an implication of the form $A \Rightarrow B$, where $A \subset I$, $B \subset I$, $A \neq \emptyset$, $B \neq \emptyset$, $A \cap B = \emptyset$. In the rule $A \Rightarrow B$, A is called the antecedent and B is called the consequent. Various metrics are used to identify the most important itemset and calculate their strength, such as support, confidence, and lift. Support metric [40] is the measure that gives an idea of how frequent an itemset is in all transactions. In other words, the support metric represents the number of transactions that contain the itemset. The Equation 5 shows how we calculate the support for an association rule.

$$\text{support}(A \Rightarrow B) = P(A \cup B) \quad (5)$$

On the other hand, the confidence [40] indicates how often the rule is true. It defines the percentage of transactions containing the antecedent A that also contain the consequent B . It can be taken as the conditional probability as shown in Equation 6.

$$\text{confidence}(A \Rightarrow B) = P(B|A) = \frac{\text{support}(A \cup B)}{\text{support}(A)} \quad (6)$$

Finally, the lift is a correlation measure used to discover and exclude the weak rules that have high confidence. The Equation 7 shows that the lift measure is calculated by dividing the confidence by the unconditional probability of the consequent [40] [38].

$$\text{lift}(A \Rightarrow B) = \frac{P(A \cup B)}{P(A)P(B)} = \frac{\text{support}(A \cup B)}{\text{support}(A)\text{support}(B)} \quad (7)$$

If the lift value is equal to 1, then A and B are independent and there is no correlation between them. If the lift value is greater than 1, then A and B are positively correlated. If the lift value is less than 1, then A and B are negatively correlated.

Various algorithms exist for mining frequent itemsets, such as Apriori [39] [41], AprioriTID [39] [41], Apriori Hybrid [39] [41], and FP-growth (Frequent pattern) [39] [42]. In this paper, we employ FP-growth algorithm to generate frequent itemsets. What makes FP-growth better than other algorithms is the fact that FP-growth algorithm relies on FP-tree (frequent pattern tree) data structure to store all data concisely and compactly which greatly helps to avoid the candidate generation step. Moreover, once the FP-tree is constructed, we can directly use a recursive divide-and-conquer approach to efficiently mine the frequent itemsets without any need to scan the database over and over again like in other algorithms [42].

IV. FREQUENT PATTERN MINING FRAMEWORK FOR RECOMMENDER SYSTEMS (FPRS)

The main problem we address in this paper is to alleviate the impact of new users and new items cold-start in recommender systems based on collaborative filtering techniques. In theory, collaborative filtering methods can be grouped into two general categories (i) memory-based techniques and (ii) model-based techniques. In memory-based techniques, we calculate the similarities between users/items based on the rating history and then generate recommendations based on the most similar users/items. In model-based techniques, we rely, e.g., on matrix factorization methods to learn the latent factors of users and items and then decompose the user-item interaction (rating) matrix into the product of two lower dimensionality matrices. Collaborative filtering methods are strictly relying on user ratings or user interactions. For that reason, these methods suffer from the cold-start problem whenever a new user joins the system or when a new item is added. In practice, both situations often lead to the inability to provide accurate or meaningful recommendations.

To tackle the cold-start problem, we implement the Frequent Pattern mining framework for Recommender Systems (FPRS). This framework extends the popularity-based approach by employing frequent pattern mining techniques to learn the user preferences depending on users' and items' characteristics. Fig 1 shows the high-level design which is used to develop the FPRS framework. The process of generating the recommendations consists of four stages: (i) Data Input, (ii) Data Preparation, (iii) Frequent Pattern Mining, and (iv) Recommendation Generation. In the first stage, we enrich the user-item rating matrix by users' demographics and items' characteristics. The data preparation stage consists of three steps. In the first one, we store only the favorable reviews by filtering out every review/rating below a determined threshold. In the second step, we perform attributes analysis and check the validity of using them for generating the recommendation. In the last step, we split the dataset for each selected attribute. It is important to note that we follow multiple strategies to perform the attribute selection. More details about these strategies will be provided later in this section. Then, in the third stage, we generate frequent itemsets using FP-Growth algorithm. Finally, we produce the recommendations in the last stage for user cold-start and item cold-start. However, the FPRS framework consists of two main modules: (i) user cold-start module, and (ii) item cold-start module. Each of these modules has dedicated strategies that are used to select the features and produce the recommendations.

Strategy 1 User Cold-Start Module

- 1: Use entire training set (no records splitting)
 - 2: Generate frequent 1-itemsets {support > min_support}
 - 3: Recommend all these frequent 1-itemsets to any new user
-

a) user cold-start module: In this module, we focus on generating recommendations for new users who signed

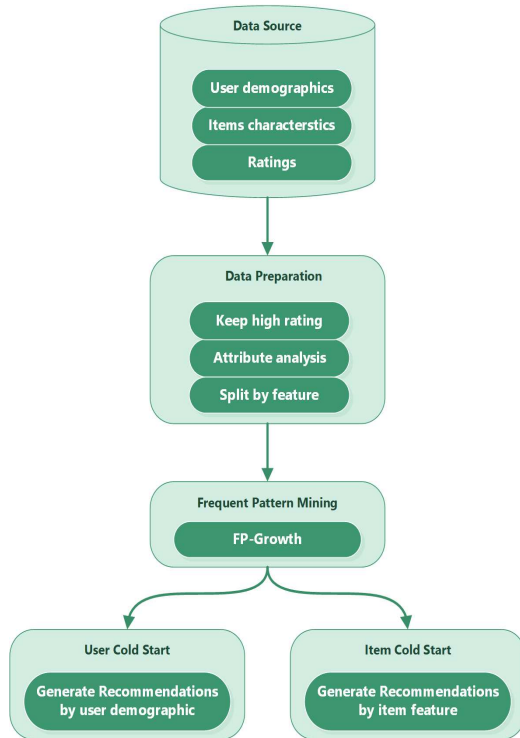


Fig. 1: Frequent Pattern Mining Framework For Recommender Systems

up recently to the system and most likely do not have, or have very few, ratings in the user-item rating matrix. We follow two strategies to generate such recommendations. These strategies differ in two main factors: (i) features selected to split the data, and (ii) the way how the frequent patterns are utilized to generate the recommendations. More details about the strategies followed in the user cold-start module are introduced in Strategy 1 and Strategy 2.

Strategy 2 User Cold-Start Module

- 1: Split the records based on users demographics (i.e. gender)
 - 2: Generate frequent 1-itemsets $\{\text{support} > \text{min_support}\}$
 - 3: Recommend to the new user all frequent 1-itemsets which are generated based on the demographics of new user
-

b) item cold-start module: In this module, we focus on generating recommendations for new items which are recently added to the system and most likely do not have, or have very few, ratings in the past. We follow multiple strategies to generate such recommendations. More details about the strategies followed in the item cold-start module are provided in Strategy 3, Strategy 4, and Strategy 5.

Finally, it is worth mentioning that the threshold values used in the above strategies are selected carefully by objectively searching for a good set of values that achieves the best performance on a given dataset. More details on how we choose these values are provided in Section V.

Strategy 3 Item Cold-Start Module

- 1: Split the records based on items characteristics (i.e. genre)
 - 2: Generate frequent itemsets $\{\text{support} > \text{min_support}\}$
 - 3: Set the participation percentage threshold
 - 4: **for each** value in genre **do**
 - 5: Find all users who involved in creating larger than the participation threshold of frequent itemsets
 - 6: **end for**
 - 7: Recommend the new item based on its genre to all users found in previous step
-

Strategy 4 Item Cold-Start Module

- 1: Split the records based on users demographics and items characteristics (i.e. gender and genre)
 - 2: Generate frequent itemsets $\{\text{support} > \text{min_support}\}$
 - 3: Set the participation percentage threshold
 - 4: **for each** value in genre **do**
 - 5: Find the dominant gender by counting how many frequent itemsets are generated by male and female
 - 6: **end for**
 - 7: Recommend the new item based on its genre to all users belong to dominant gender who involved in creating larger than the participation threshold of frequent itemsets
-

Strategy 5 Item Cold-Start Module

- 1: Split the records based on users demographics and items characteristics (i.e. gender and genre)
 - 2: Generate frequent 1-itemsets $\{\text{support} > \text{min_support}\}$
 - 3: Set the participation percentage threshold
 - 4: Assign frequent 1-itemsets created by male to one cluster, and frequent 1-itemsets created by female to another cluster
 - 5: Find the center of each cluster
 - 6: Calculate the distance between the new item and the center of each cluster
 - 7: Recommend the new item to all users who involved in creating larger than the participation threshold of frequent itemsets in the closest cluster
-

V. EVALUATION METHODOLOGY

In this section, we conduct comprehensive experiments to evaluate the performance of the FPRS recommender system.

A. Dataset and Evaluation Measures

In our experiments, we used two datasets (MovieLens 100K and MovieLens 1M)¹ which were collected by the GroupLens research project at the University of Minnesota. MovieLens 100K contains 100,000 ratings given by 943 users on 1682 movies on a scale from 1 to 5. While MovieLens 1M contains 1,000,000 ratings of approximately 3,900 movies made by 6,040 users on a scale from 1 to 5. In both datasets, we

¹<https://grouplens.org/datasets/movielens/>

combine three files (users.data, items.data, ratings.data) in order to join users' demographics, items' characteristics, and ratings in one dataset. The final/joined dataset contains userId, itemId, rating, gender, age, occupation, and genres attributes (cf. Table I). Moreover, we performed further analysis of the features we used in our experiments (gender, genre) to understand the interrelation between these features and obtained results. Figures 2 show the most popular movie genres among males and females for both datasets (MovieLens 100K and MovieLens 1M).

TABLE I: Selected data characteristics.

Attribute Name	Data Type	Value Range (MovieLens 100K)	Value Range (MovieLens 1M)
gender	Character	M-F	M-F
age	Number	Under 18-73	Under 18-56
occupation	Text	21 occupation	21 occupation
genres	Text	19 genres	19 genres

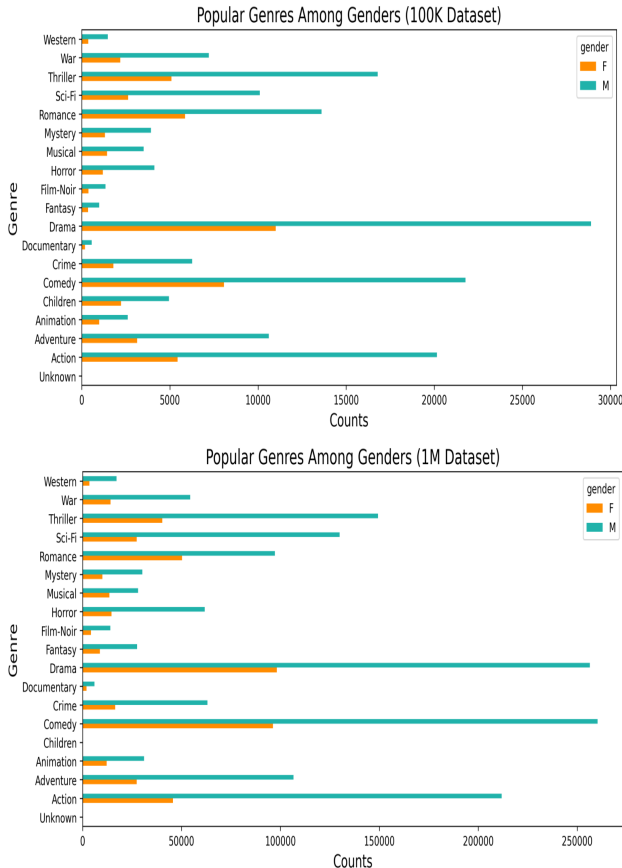


Fig. 2: Histogram of the variables (rating, genre and gender) in the MovieLens 100K and 1M datasets.

After clearing the data from invalid records, we split it into training and testing datasets according to FPRS module as follows:

- For the user cold-start module testing, the 20 users with the highest number of ratings on the 50 most popular movies were selected. The ratings given by all those 20 users (9052 records in MovieLens 100K and 25533 records in MovieLens 1M) are considered a testing set, keeping the rest of the records in the training set. This way, all the record related to the selected users were removed from the test set, which simulates the cold-start problem associated with new users.
- To properly evaluate the item cold-start module, the testing data is chosen similarly. We firstly find the 50 most active users. Then, we select the 20 most rated movies by those 50 users. The ratings of all those 20 movies by all the users in our data (7320 records in MovieLens 100K and 41105 records in MovieLens 1M) are considered as a testing set, keeping the rest of the records in the training set. Note that all the ratings for the selected movies are removed from the training data set, which corresponds to the item cold-start.

In our study, we consider a binary decision task whether a given item (i.e., movie) is appropriate for the user. To correctly model this situation for the MovieLens data, we assume that films rated by users 4 or 5 are preferred by them (belong to the positive class). In contrast, those ranked lower are poorly matched to the users. Therefore, the FPRS recommender system feedback for each new user or item is binary information: recommend or not recommend. Following that, in order to assess the quality of the prediction, the F1 measure is used [43].

$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \quad (8)$$

where precision quantifies the number of correct positive recommendations made (see Equation 9). While recall quantifies the number of correct positive recommendations made out of all positive predictions that could have been made (see Equation 10).

$$\text{Precision} = \frac{TP}{TP + FP} \quad (9)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (10)$$

Moreover, we use the accuracy metric to measure all the correctly identified cases. This measure is mostly used when all the classes are equally important.

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \quad (11)$$

B. Baseline Recommender System

To showcase the strengths of frequent pattern mining in RS, we build a baseline model in a similar way to the strategies described previously, but the FP-growth algorithm was omitted. In order to evaluate both modules of FPRS, two versions of baseline RS are developed as follows:

- New user baseline model: for each movie in the training set, the most common gender and age group of the user was found. When a new user comes, they get recommended all movies that were assigned to their age group and gender.
- New item baseline model: it works similarly. We find the most popular (watched) genre for each user. Then each new movie is recommended to all users whose favorite genre was the same as the new movie's.

C. Performance Comparison and Analysis

In order to provide a fair comparison, we use precision, recall, F1, and accuracy measures to compare the performance of FPRS against the baseline RS. After splitting the dataset into the training and testing sets and training both baseline and FPRS recommendation systems, we run two experiments to evaluate the user cold-start module and item cold-start module.

a) *User cold-start module*: In the performed experiment, we evaluated the user cold-start module in FPRS. We calculated precision, recall, F1, and accuracy measures for the results generated by the baseline RS and FPRS following the two strategies, which we have described in Section IV. The comparison results of this evaluation method are shown in Tables II and III. The results show that all developed strategies outperformed the baseline RS for precision, F1, and accuracy on both data-sets. However, the baseline solution reported higher recall, which is quite natural since the developed methods are more selective, providing more apt results with a tradeoff that some potentially relevant movies may be omitted. For the user-cold start problem, both strategies were evaluated at *min_support* value of 0.08 and performed similarly. The second one was just slightly better.

TABLE II: Evaluation for user cold-start (MovieLens 100k).

Strategy	Precision	Recall	F1-score	Accuracy
Baseline	0.24	0.72	0.33	0.44
Strategy 1	0.58	0.56	0.57	0.80
Strategy 2	0.59	0.58	0.58	0.80

TABLE III: Evaluation for user cold-start (MovieLens 1M).

Strategy	Precision	Recall	F1-score	Accuracy
Baseline	0.33	0.82	0.44	0.47
Strategy 1	0.63	0.63	0.62	0.77
Strategy 2	0.64	0.64	0.63	0.78

b) *Item cold-start module*: In the second experiment, we evaluated the item cold-start module in FPRS. We calculated precision, recall, F1, and accuracy measures for the results generated by the baseline RS and FPRS following all the strategies described in Section IV. The comparative summary of this evaluation is shown in Tables II and III. The results show that the performance of FPRS, using all strategies, is superior to the baseline solution. However, the results differ

slightly between datasets. For MovieLens 100K, all strategies reported similar recall. Regarding precision and F1 measures, the most successful in dealing with new items in this data appeared to be strategy no. 4, which is based on both items' and users' characteristics. However, for the applications that do require high accuracy, it would be better to apply strategy no. 5, which was also superior in terms of recall, F1, and accuracy on the second data-set (MovieLens 1M). All strategies were evaluated at the participation threshold value of 30% and *min_support* value of 0.2.

TABLE IV: Evaluation for item cold-start (MovieLens 100k).

Strategy	Precision	Recall	F1-score	Accuracy
Baseline	0.38	0.3	0.32	0.59
Strategy 3	0.69	0.64	0.66	0.75
Strategy 4	0.79	0.62	0.69	0.84
Strategy 5	0.67	0.63	0.63	0.86

TABLE V: Evaluation for item cold-start (MovieLens 1M).

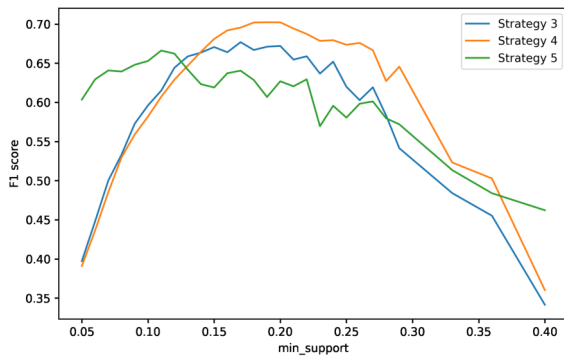
Strategy	Precision	Recall	F1-score	Accuracy
Baseline	0.49	0.43	0.43	0.64
Strategy 3	0.65	0.71	0.64	0.76
Strategy 4	0.64	0.73	0.66	0.83
Strategy 5	0.6	0.79	0.66	0.86

D. Thresholds Sensitivity Analysis

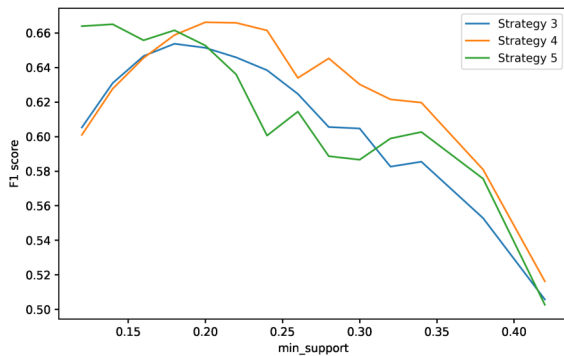
In FPRS model, we use some threshold values, such as *min_support* and participation percentage, in order to extract frequent itemsets and produce relevant recommendations for new users and new items. In this section, we conduct some experiments to show how changing those values may impact performance of FPRS. Moreover, the output of this experiment helps to find the optimal values of these thresholds, and hence to conduct fair and reliable experiments.

In the first experiment, we aim to find the optimal value of *min_sup* threshold by evaluating FPRS (item cold-start module) using different *min_sup* values. Fig 3a shows how the F1-score of FPRS is impacted by applying different values for MovieLena 100K data. Observably, the best *min_sup* values for all strategies used in FPRS (item cold-start module) are between 0.1 and 0.2. The similar observations, regarding MovieLens 1M data, we may find in Figure 3b.

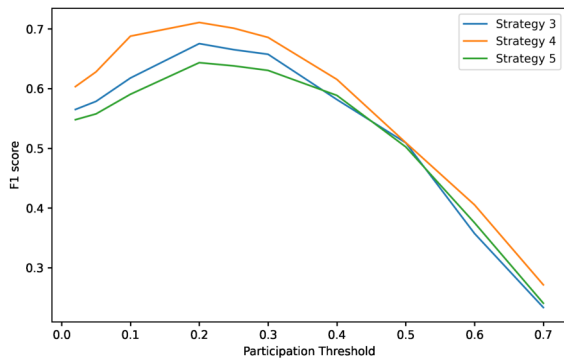
In the second experiment, we search for the optimal value of the participation threshold in FPRS (item cold-start module). Figures 3c and 3d show how F1-score of FPRS is impacted by applying different values for both investigated datasets. Observably, the best participation threshold values for all strategies used in FPRS (item cold-start module) are between 15% and 30%. Finally, it is worth noting that when we run this experiment, we use the optimal value of *min_sup* we found in the previous experiment.



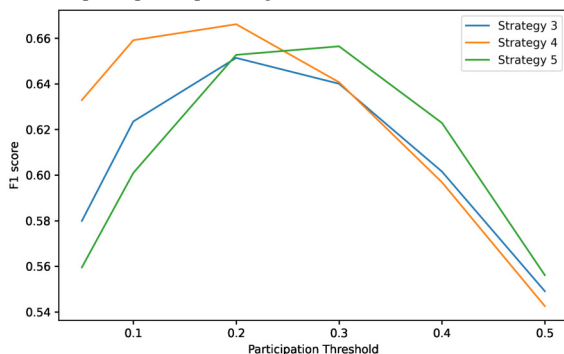
(a) min_supp threshold (MovieLens 100k)



(b) min_supp threshold (MovieLens 1M)



(c) participation percentage threshold (MovieLens 100k)



(d) participation percentage threshold (MovieLens 1M)

Fig. 3: Sensitivity analysis for min_supp and participation percentage thresholds for MovieLens 100k and 1M.

VI. CONCLUSIONS AND FUTURE WORKS

This article introduces FPRS, a novel recommender system, which methodically utilizes the ratings to discover frequent itemsets associated with selected users/items features and then incorporates these frequent itemsets in generating recommendations for new users and items. Our study evaluates multiple strategies for creating frequent itemsets to produce meaningful and relevant recommendations.

To evaluate FPRS, we conducted comprehensive experiments on MovieLens 100K and 1M datasets using the FP-growth algorithm to generate the frequent itemsets. The experimental results show that FPRS has outperformed the baseline recommender system in terms of the precision, recall, F1, and accuracy measures.

In the future work, we plan to incorporate additional contextual information and evaluate more advanced algorithms, such as AprioriTID and Apriori Hybrid, in the process of producing frequent patterns. Another important aspect to consider is to evaluate our method against more state-of-the-art recommender systems on various datasets. Furthermore, we plan to consider changes in users' behavior and preferences by periodically updating frequent itemsets based on recent changes in rating history. It would also be of value to extend the users' and items' data representation by applying a more advanced feature extraction to model the similarities among them more effectively [44], [45], [46].

ACKNOWLEDGMENT

Research co-funded by Polish National Science Centre (NCN) grant no. 2018/31/N/ST6/00610.

REFERENCES

- [1] Z. Batmaz, A. Yürekli, A. Bilge, and C. Kaleli, "A review on deep learning for recommender systems: challenges and remedies," *Artif. Intell. Rev.*, vol. 52, no. 1, pp. 1–37, 2019. doi: 10.1007/s10462-018-9654-y
- [2] M. Kawai, H. Sato, and T. Shiohama, "Topic model-based recommender systems and their applications to cold-start problems," *Expert Systems with Applications*, vol. 202, p. 117129, 2022. doi: 10.1016/j.eswa.2022.117129
- [3] M. Pulis and J. Bajada, *Siamese Neural Networks for Content-Based Cold-Start Music Recommendation*. New York, NY, USA: Association for Computing Machinery, 2021, p. 719–723. ISBN 9781450384582
- [4] E. Kannout, "Context clustering-based recommender systems," in *2020 15th Conference on Computer Science and Information Systems (FedCSIS)*, 2020. doi: 10.15439/2020F54 pp. 85–91.
- [5] M. Pondel and J. Korczak, "Collective clustering of marketing data - recommendation system upsally," in *Proceedings of the 2018 Federated Conference on Computer Science and Information Systems, FedCSIS 2018, Poznań, Poland, September 9-12, 2018*, ser. Annals of Computer Science and Information Systems, M. Ganzha, L. A. Maciaszek, and M. Paprzycki, Eds., vol. 15, 2018. doi: 10.15439/2018F217 pp. 801–810.
- [6] B. Yi, X. Shen, H. Liu, Z. Zhang, W. Zhang, S. Liu, and N. Xiong, "Deep Matrix Factorization With Implicit Feedback Embedding for Recommendation System," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 8, pp. 4591–4601, 2019. doi: 10.1109/TII.2019.2893714
- [7] M. Jalili, S. Ahmadian, M. Izadi, P. Moradi, and M. Salehi, "Evaluating Collaborative Filtering Recommender Algorithms: A Survey," *IEEE Access*, vol. 6, pp. 74 003–74 024, 2018. doi: 10.1109/ACCESS.2018.2883742

- [8] R. Chen, Q. Hua, Y.-S. Chang, B. Wang, L. Zhang, and X. Kong, "A Survey of Collaborative Filtering-Based Recommender Systems: From Traditional Methods to Hybrid Methods Based on Social Networks," *IEEE Access*, vol. 6, pp. 64 301–64 320, 2018. doi: 10.1109/ACCESS.2018.2877208
- [9] B. Walek and V. Fojtik, "A hybrid recommender system for recommending relevant movies using an expert system," *Expert Systems with Applications*, vol. 158, p. 113452, 2020. doi: 10.1016/j.eswa.2020.113452
- [10] Y. Pérez-Almaguer, R. Yera, A. A. Alzahrani, and L. Martínez, "Content-based group recommender systems: A general taxonomy and further improvements," *Expert Systems with Applications*, vol. 184, p. 115444, 2021. doi: 10.1016/j.eswa.2021.115444
- [11] Y. Afoudi, M. Lazaar, and M. Al Achhab, "Hybrid recommendation system combined content-based filtering and collaborative prediction using artificial neural network," *Simulation Modelling Practice and Theory*, vol. 113, p. 102375, 2021. doi: 10.1016/j.simpat.2021.102375
- [12] R. Pasricha and J. McAuley, "Translation-Based Factorization Machines for Sequential Recommendation," in *Proceedings of the 12th ACM Conference on Recommender Systems*, ser. RecSys'18. New York, NY, USA: Association for Computing Machinery, 2018. doi: 10.1145/3240323.3240356. ISBN 9781450359016 p. 63–71.
- [13] H. Wu, Z. Zhang, K. Yue, B. Zhang, J. He, and L. Sun, "Dual-regularized matrix factorization with deep neural networks for recommender systems," *Knowledge-Based Systems*, vol. 145, pp. 46–58, 2018. doi: 10.1016/j.knsys.2018.01.003
- [14] F. Liu, R. Tang, X. Li, W. Zhang, Y. Ye, H. Chen, H. Guo, and Y. Zhang, "Deep Reinforcement Learning based Recommendation with Explicit User-Item Interactions Modeling," 2018.
- [15] M. K. Najafabadi, A. H. Mohamed, and M. N. Mahrin, "A survey on data mining techniques in recommender systems," *Soft Comput.*, vol. 23, no. 2, pp. 627–654, 2019. doi: 10.1007/s00500-017-2918-7
- [16] M. Grzegorowski, A. Janusz, S. Lazewski, M. Swiechowski, and M. Jankowska, "Prescriptive analytics for optimization of FMCG delivery plans," in *Information Processing and Management of Uncertainty in Knowledge-Based Systems - 19th International Conference, IPMU 2022, Milan, Italy, July 11-15, 2022, Proceedings, Part II*, ser. Communications in Computer and Information Science, D. Ciucci, I. Couso, J. Medina, D. Ślęzak, D. Petturiti, B. Bouchon-Meunier, and R. R. Yager, Eds., vol. 1602. Springer, 2022. doi: 10.1007/978-3-031-08974-9_4 pp. 44–53.
- [17] A. Janusz, M. Grzegorowski, M. Michalak, E. Wróbel, M. Sikora, and D. Ślęzak, "Predicting Seismic Events in Coal Mines Based on Underground Sensor Measurements," *Engineering Applications of Artificial Intelligence*, vol. 64, pp. 83–94, 2017.
- [18] Z. Zhu, J. Kim, T. Nguyen, A. Fenton, and J. Caverlee, *Fairness among New Items in Cold Start Recommender Systems*. New York, NY, USA: Association for Computing Machinery, 2021, p. 767–776. ISBN 9781450380379
- [19] Z. Cui, X. Xu, F. XUE, X. Cai, Y. Cao, W. Zhang, and J. Chen, "Personalized Recommendation System Based on Collaborative Filtering for IoT Scenarios," *IEEE Transactions on Services Computing*, vol. 13, no. 4, pp. 685–695, 2020. doi: 10.1109/TSC.2020.2964552
- [20] A. Karpus, I. Vagliano, K. Goczyla, and M. Morisio, "An ontology-based contextual pre-filtering technique for recommender systems," in *Proceedings of the 2016 Federated Conference on Computer Science and Information Systems, FedCSIS 2016, Gdańsk, Poland, September 11-14, 2016*, ser. Annals of Computer Science and Information Systems, M. Ganzha, L. A. Maciaszek, and M. Paprzycki, Eds., vol. 8. IEEE, 2016. doi: 10.15439/2016F428 pp. 411–420.
- [21] S. Natarajan, S. Vairavasundaram, S. Natarajan, and A. H. Gandomi, "Resolving data sparsity and cold start problem in collaborative filtering recommender system using linked open data," *Expert Systems with Applications*, vol. 149, p. 113248, 2020. doi: 10.1016/j.eswa.2020.113248
- [22] A. Karpus, I. Vagliano, and K. Goczyla, "Serendipitous recommendations through ontology-based contextual pre-filtering," in *Beyond Databases, Architectures and Structures. Towards Efficient Solutions for Data Analysis and Knowledge Representation - 13th International Conference, BDAS 2017, Ustroń, Poland, May 30 - June 2, 2017, Proceedings*, ser. Communications in Computer and Information Science, S. Kozielski, D. Mrozek, P. Kasprowski, B. Malysiak-Mrozek, and D. Kostrzewa, Eds., vol. 716, 2017. doi: 10.1007/978-3-319-58274-0_21 pp. 246–259.
- [23] A. Pawlicka, M. Pawlicki, R. Kozik, and R. S. Choraś, "A systematic review of recommender systems and their applications in cybersecurity," *Sensors*, vol. 21, no. 15, 2021. doi: 10.3390/s21155248. [Online]. Available: <https://www.mdpi.com/1424-8220/21/15/5248>
- [24] K. Ghazinour, S. Matwin, and M. Sokolova, "Monitoring and recommending privacy settings in social networks," in *Joint 2013 EDBT/ICDT Conferences, EDBT/ICDT '13, Genoa, Italy, March 22, 2013, Workshop Proceedings*, G. Guerrini, Ed. ACM, 2013. doi: 10.1145/2457317.2457344 pp. 164–168.
- [25] A. L. Vazine Pereira and E. R. Hruschka, "Simultaneous co-clustering and learning to address the cold start problem in recommender systems," *Knowledge-Based Systems*, vol. 82, pp. 11–19, 2015. doi: 10.1016/j.knsys.2015.02.016
- [26] M. Nilashi, O. Ibrahim, and K. Bagherifard, "A recommender system based on collaborative filtering using ontology and dimensionality reduction techniques," *Expert Systems with Applications*, vol. 92, pp. 507–520, 2018. doi: 10.1016/j.eswa.2017.09.058
- [27] M. Grzegorowski, A. Janusz, D. Ślęzak, and M. S. Szczuka, "On the role of feature space granulation in feature selection processes," in *2017 IEEE International Conference on Big Data (IEEE BigData 2017), Boston, MA, USA, December 11-14, 2017*, J. Nie, Z. Obradovic, T. Suzumura, R. Ghosh, R. Nambiar, C. Wang, H. Zang, R. Baeza-Yates, X. Hu, J. Kepner, A. Cuzzocrea, J. Tang, and M. Toyoda, Eds. IEEE Computer Society, 2017. doi: 10.1109/BigData.2017.8258124 pp. 1806–1815.
- [28] M. Grzegorowski and D. Ślęzak, "On resilient feature selection: Computational foundations of r-C-reducts," *Inf. Sci.*, vol. 499, pp. 25–44, 2019. doi: 10.1016/j.ins.2019.05.041
- [29] M. Grzegorowski, "Governance of the Redundancy in the Feature Selection Based on Rough Sets' Reducts," in *Rough Sets - International Joint Conference, IJCRS 2016, Santiago de Chile, Chile, October 7-11, 2016, Proceedings*, ser. Lecture Notes in Computer Science, V. Flores, F. A. C. Gomide, A. Janusz, C. Meneses, D. Miao, G. Peters, D. Ślęzak, G. Wang, R. Weber, and Y. Yao, Eds., vol. 9920, 2016. doi: 10.1007/978-3-319-47160-0_50 pp. 548–557. [Online]. Available: https://doi.org/10.1007/978-3-319-47160-0_50
- [30] M. asid and R. Ali, *Use of Soft Computing Techniques for Recommender Systems: An Overview*. Singapore: Springer Singapore, 2017, pp. 61–80. ISBN 978-981-10-7098-3
- [31] I. Viktoratos, A. Tsadiras, and N. Bassiliades, "Combining community-based knowledge with association rule mining to alleviate the cold start problem in context-aware recommender systems," *Expert Systems with Applications*, vol. 101, pp. 78–90, 2018. doi: 10.1016/j.eswa.2018.01.044
- [32] H. Steck, "Collaborative filtering via high-dimensional regression," *CoRR*, vol. abs/1904.13033, 2019. doi: 10.48550/ARXIV.1904.13033
- [33] H. Kwasknicka and T. Ovedenski, "Pix2trips - a system supporting small groups of urban tourists," in *Proceedings of the 16th Conference on Computer Science and Intelligence Systems, Online, September 2-5, 2021*, ser. Annals of Computer Science and Information Systems, M. Ganzha, L. A. Maciaszek, M. Paprzycki, and D. Ślęzak, Eds., vol. 25, 2021. doi: 10.15439/2021F130 pp. 141–145.
- [34] H. Sobhanan and A. K. Mariappan, "Addressing cold start problem in recommender systems using association rules and clustering technique," in *2013 International Conference on Computer Communication and Informatics*, 2013. doi: 10.1109/ICCCI.2013.6466121 pp. 1–5.
- [35] W. Feng, Q. Zhu, J. Zhuang, and S. Yu, "An expert recommendation algorithm based on pearson correlation coefficient and fp-growth," *Clust. Comput.*, vol. 22, no. Supplement, pp. 7401–7412, 2019. doi: 10.1007/s10586-017-1576-y
- [36] J. S. Breese, D. Heckerman, and C. Kadie, "Empirical analysis of predictive algorithms for collaborative filtering," in *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, ser. UAI'98. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1998. ISBN 155860555X p. 43–52.
- [37] M. D. Ekstrand, J. T. Riedl, and J. A. Konstan, "Collaborative filtering recommender systems," *Found. Trends Hum.-Comput. Interact.*, vol. 4, no. 2, p. 81–173, feb 2011. doi: 10.1561/1100000009. [Online]. Available: <https://doi.org/10.1561/1100000009>
- [38] J. Han, M. Kamber, and J. Pei, "6 - mining frequent patterns, associations, and correlations: Basic concepts and methods," in *Data Mining (Third Edition)*, third edition ed., ser. The Morgan Kaufmann Series in Data Management Systems, J. Han, M. Kamber, and J. Pei, Eds. Boston: Morgan Kaufmann, 2012, pp. 243–278. ISBN 978-0-12-381479-1. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/B978012381479100006X>

- [39] J. Hipp, U. Güntzer, and G. Nakhaeizadeh, "Algorithms for association rule mining — a general survey and comparison," *SIGKDD Explor. Newsl.*, vol. 2, no. 1, p. 58–64, jun 2000. doi: 10.1145/360402.360421. [Online]. Available: <https://doi.org/10.1145/360402.360421>
- [40] U. Fayyad, *Knowledge Discovery in Databases: An Overview*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 28–47. ISBN 978-3-662-04599-2. [Online]. Available: https://doi.org/10.1007/978-3-662-04599-2_2
- [41] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules in large databases," in *Proceedings of the 20th International Conference on Very Large Data Bases*, ser. VLDB '94. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1994. ISBN 1558601538 p. 487–499.
- [42] J. Han, J. Pei, and Y. Yin, "Mining frequent patterns without candidate generation," *SIGMOD Rec.*, vol. 29, no. 2, p. 1–12, may 2000. doi: 10.1145/335191.335372. [Online]. Available: <https://doi.org/10.1145/335191.335372>
- [43] T. Silveira, M. Zhang, X. Lin, Y. Liu, and S. Ma, "How good your recommender system is? A survey on evaluations in recommendation," *International Journal of Machine Learning and Cybernetics*, vol. 10, pp. 813–831, 2019. doi: 10.1007/s13042-017-0762-9
- [44] M. Grzegorowski, "Massively Parallel Feature Extraction Framework Application in Predicting Dangerous Seismic Events," in *Proceedings of the 2016 Federated Conference on Computer Science and Information Systems, FedCSIS 2016, Gdańsk, Poland, September 11-14, 2016*, ser. Annals of Computer Science and Information Systems, M. Ganzha, L. A. Maciaszek, and M. Paprzycki, Eds., vol. 8. IEEE, 2016. doi: 10.15439/2016F90 pp. 225–229. [Online]. Available: <https://doi.org/10.15439/2016F90>
- [45] E. Zdravevski, P. Lameski, R. Mingov, A. Kulakov, and D. Gjorgievikj, "Robust histogram-based feature engineering of time series data," in *2015 Federated Conference on Computer Science and Information Systems, FedCSIS 2015, Łódź, Poland, September 13-16, 2015*, ser. Annals of Computer Science and Information Systems, M. Ganzha, L. A. Maciaszek, and M. Paprzycki, Eds., vol. 5. IEEE, 2015. doi: 10.15439/2015F420 pp. 381–388. [Online]. Available: <https://doi.org/10.15439/2015F420>
- [46] M. Grzegorowski and S. Stawicki, "Window-based feature extraction framework for multi-sensor data: A posture recognition case study," in *2015 Federated Conference on Computer Science and Information Systems, FedCSIS 2015, Łódź, Poland, September 13-16, 2015*, ser. Annals of Computer Science and Information Systems, M. Ganzha, L. A. Maciaszek, and M. Paprzycki, Eds., vol. 5. IEEE, 2015. doi: 10.15439/2015F425 pp. 397–405.