Computer Science and Information Systems 20(4):1343–1366 https://doi.org/10.2298/CSIS221116052K

Towards Addressing Item Cold-Start Problem in Collaborative Filtering by Embedding Agglomerative Clustering and FP-Growth into the Recommendation System ⋆

Eyad Kannout, Michał Grodzki, and Marek Grzegorowski

Institute of Informatics, University of Warsaw
Banacha 2, Warsaw, Poland
{eyad.kannout,m.grzegorowski}@mimuw.edu.pl,
m.grodzki@students.mimuw.edu.pl

Abstract. This paper introduces a frequent pattern mining framework for recommender systems (FPRS) - a novel approach to address the items' cold-start problem. This difficulty occurs when a new item hits the system, and properly handling such a situation is one of the key success factors of any deployment. The article proposes several strategies to combine collaborative and content-based filtering methods with frequent items mining and agglomerative clustering techniques to mitigate the cold-start problem in recommender systems. The experiments evaluated the developed methods against several quality metrics on three benchmark datasets. The conducted study confirmed usefulness of FPRS in providing apt outcomes even for cold items. The presented solution can be integrated with many different approaches and further extended to make up a complete and standalone RS.

Keywords: recommendation system, cold-start problem, frequent pattern mining, quality of recommendations.

1. Introduction

Many modern businesses undergo digital transformation, moving their offerings online, which allows for providing broadly available and more advanced products and services to customers or users [53, 74]. This trend has led to an overwhelming amount of content and the high velocity of new items reaching the systems daily. Finding the object of interest has become very time-consuming for many people, especially after moving most e-commerce to automated remote channels without qualified human advisors [2, 22, 78]. Viable solutions to this problem are recommender systems (RS), which leverage the rating history and possibly some other information, such as users' demographics or items' characteristics [57, 59].

Recommender systems are indispensable in allowing customers to find a desired product or service [69]. The quality of RS is mainly impacted by the density of the historical user-item interactions and may encounter some significant difficulties due to particular data characteristics related to volume, limited content analysis, sparsity of data, or cold

⋆ This is an extended version of the paper "Utilizing Frequent Pattern Mining for Solving Cold-Start Problem in Recommender Systems" [35] published at the FedCSIS'22 conference.

items [5, 73]. The last one is particularly challenging, having broad interest among researchers and practitioners [79, 86]. This difficulty occurs when a new item hits the system, and an RS attempts to generate recommendations with very scarce and insufficient historical ratings available [63]. Many state-of-the-art recommendation algorithms may generate unreliable recommendations for such cases since they cannot learn the preference embedding of these new items [42].

In this study, we present a particular take on the challenge of devising more effective and efficient recommendation techniques with specific attention to the problems of the sparsity of interactions and cold items. The developed Frequent Pattern mining framework for Recommender Systems (FPRS) is based on the popularity approaches extended by the FP-growth algorithm to generate frequent patterns based on items' characteristics and by adding an agglomerative clustering step into the developed pipeline [6, 38]. This way, we better reflect and leverage content-based similarities between new items, even for the partially incomplete data. The agglomerative clustering methods allow us to tune the number and size of clusters dynamically.

The developed method creates a kind of platform incorporating several strategies, which is a distinguishing feature of this approach compared to the methods reported in the literature, typically reporting one universal system. We believe that, in practical applications, the observed differences between the analyzed problems and datasets are significant. A single solution may be ineffective depending on the quality measure of choice or data characteristics. Hence, data scientists need to operate with a whole toolset and adjust it to the particular case.

Compared to our former research on utilizing frequent pattern mining for solving the cold-start problem in recommender systems [35], this study is focused on the problems of the sparsity of interactions and cold items. In this regard, we significantly extended the formerly developed strategies to better reflect and leverage content-based similarities between new items, even for the partially incomplete data. We also included new datasets for the evaluation procedure to give a more versatile assessment of our method and conducted a broad review of research efforts related to the discussed problem. The main contributions of this paper are as follows:

1. An extended version of *frequent pattern mining framework for recommender systems* (FPRS) - a hybrid recommender system that utilizes the FP-growth algorithm to produce frequent itemsets based on the ratings in the user-item matrix.
2. A novel approach to utilize the items' features to extract particular patterns based on the features selected and agglomerative clustering to mitigate the sparsity issue.
3. New strategies to mitigate the cold-start problem by using the discovered patterns to properly assess users' interest in new items.
4. An empirical evaluation of the proposed approach against two state-of-the-art models that are designed for the cold-start recommender system. The experiments are conducted on well-established benchmark datasets. Namely, MovieLens 100K, MovieLens 1M, and LDOS-CoMoDa.

The remainder of this paper is organized as follows. Section 2 describes and reviews major research efforts on the cold-start problem in the domain of recommender systems. In Section 3, we provide background information for collaborative filtering and frequent pattern mining. In Section 4, 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 5 evaluates and compares the proposed model with a baseline solution. In Section 6, we discuss the limitations of FPRS and possible future research directions. Finally, in Section 7, we summarise the study.

2. 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 [33, 62]. The aforementioned capability of RSs makes those techniques especially useful in broad areas of applications like eCommerce, online marketing, social networks, price-comparison services, or even energy market [29, 32, 48, 83]. RS may also incorporate several extensions, like context awareness, action recommendations, prescriptions, or some techniques derived from game theory [20, 65, 69].

There are many taxonomies for RS [8]. The most common approaches refer to content-based or collaboration-based techniques, and their various hybridizations [30,37,78]. 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 3.1. 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 [35, 64]. Because of the significantly different characteristics of those approaches, it is advisable to construct hybridizations of both [1], as further discussed in our study.

A typical RS consists of 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 deep neural networks or factorization machines (FM) [50, 60, 80]. 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 [46, 54, 83]. 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 scalability and unfairness of recommendations for new users or items that is often referred to as the *cold-start* problem [31, 35, 87].

Solving scalability issues is one of the most common tasks when deploying big-scale recommender systems [5,16,67]. Especially as the number of users and items significantly grows over time, it is essential for RS 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 [10, 75]. Some RS suffer from 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 [39]. Recently, there is also an increasing interest in privacy awareness and explainability of recommendations [4, 14, 61]. Another aspect that is particularly noticeable for collaborative filtering is related to the sparsity of user-item interactions [40].

Together with the growing amount of items available for the recommendations, the quality of CF-based methods may be impacted by an insufficient number of items rated by each user [55]. One of the possibilities to address this issue is to rally on auxiliary data or additional information sources such as user/item profiles or user reviews on items [28]. Some approaches aim to resolve the data sparsity problem by generating data (e.g., purchasees) from machine learning models of auxiliary feedback, or from the nearest neighbors with a set of purchased items in multiple dimensions [25]. Other popular approaches apply selected clustering methods, often referring to notions of similarity [19,82]. In [85], the authors use user clustering to reconstruct the user-item bipartite network such that the network density is significantly improved. The recommendations on this dense network thus can achieve much higher accuracy than on the original sparse one. In [44], the density-based clustering algorithm is used for coping with the sparsity problem. In [81], the authors employ a granular computing model to realize the nearest neighbor clustering and a covering rough granular computing model for the collaborative filtering recommendation algorithm. The application of granular methods [21], selected approaches to clustering, and various similarity measures seem to be an exciting research direction, yielding promising results [41]. From our perspective, agglomerative clustering methods are particularly interesting. They allow us to utilize various notions of similarity (e.g., the Jaccard coefficient) and manage the number and size of clusters [71].

Agglomerative clustering algorithms create a hierarchy of data clusters by starting from singleton groupings (clusters containing a single element) and iteratively merging the closest groups into a bigger cluster [15]. This process ends when all data instances are merged into a single cluster, hence, is often referred to as a bottom-up approach. To measure the proximity (dissimilarity) between groups, agglomerative clustering algorithms employ so-called linkage functions like *single linkage* or *complete linkage*. The first one defines the dissimilarity between two groups as the smallest distance between any two instances from those groups. Analogically, the second function asserts the proximity of groups as the largest distance between any two instances [71].

Cold-start problem occurs whenever a RS tries to generate recommendations for either a new user who signed up recently to the system without having any rating records available yet or when a new item is added to the system without any rating given to that item so far. Most state-of-the-art recommendation algorithms generate unreliable recommendations for such cases since they cannot learn the preference embedding of these new users/items [49, 72]. In content-based filtering (CBF), it is necessary to learn user preferences in order to provide reliable recommendations. Therefore, CBF suffers from the user cold-start problem when new users who signed up recently do not have, or have very few, ratings. Hence, the quality of recommendation will be impacted by an insufficient number of rated items [13, 61]. Many studies recognize the challenge of fairness among new items' recommendations in cold systems, [79, 86, 87].

The difficulty arises due to the deficient information about new entities [76]. 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 [87]. 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 [42], the authors propose hybrid recommender models that use content-based filtering and latent Dirichlet allocation (LDA)-based models. In

[78], we may find a hybrid RS that combines the singular-value decomposition-based collaborative filtering with content-based and fuzzy expert systems.

In literature, we may find many efforts to resolve the cold-start problem [11,47]. In [7], the authors aim to address the cold-start problem by extending the matrix-factorization-based methods, namely SVD, SVD++, and the NMF models, using three simple regularization differentiating functions (RDF) so that the regularization weights on different items and users are set based on their popularity. In [9], the item-side cold-start problem is addressed with the concept of weak supervision. The authors introduced a new process for identifying representative reviewers and developed a method to predict the expected preferences for new items by combining content-based filtering and the preferences of representative users. In [87], the authors formalize fairness among new items with the concepts of equal opportunity and Rawlsian Max-Min fairness and present a learnable post-processing framework with score scaling and joint-learning generative models. Zhu et al. propose a novel model designed to overcome cold start by (i) a combined separate-training and joint-training framework to overcome the error superimposition issue and improve model quality; (ii) a Randomized Training mechanism to promote the effectiveness of model learning; and (iii) a Mixture-of-Experts Transformation mechanism to provide personalized transformation functions.

There are many more techniques to dealing with the cold-start problem by combining collaborative filtering with content-based methods, including using simultaneous co-clustering [77], self-organizing maps, meta-learning [51], or Siamese neural networks [63]. There are also attempts to combine RS with various dimensionality reduction techniques [56]. 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 [21], and particularly on resilient ML techniques [17, 23] - 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 [3], particularly interesting for context-aware RSs [33, 43, 58].

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 [50, 64]. 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 [87]. Hence, an additional effort to deal with biases in data is required [70]. Another interesting approach to dealing with insufficient or missing historical transactions avail additional sources of information to enhance the data representation. In particular, in [55], 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 [45]. Combining community-based knowledge with association rule mining to alleviate the cold-start problem is also bringing very promising results [76]. Referring to association rule mining (cf. [68]) and frequent pattern mining (cf. [12]) techniques to address the cold-start problem is interesting also from the perspective of speeding up the recommender systems [36]. For this reason, frequent pattern mining is particularly interesting in our research, and we review this field in detail in Section 3.2.

Most reported cases focus on alleviating cold users [11, 47]. Scenarios related to new items - without any feedback history - are investigated far less often. Whereas, having in mind the still-emerging new products and services, such approaches are in high demand and require further research attention. Additionally, we did not find in the literature any attempt to address the cold-start problem using frequent pattern mining methods. 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 [55, 63].

3. 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.

3.1. Collaborative Filtering

The basic idea behind collaborative filtering (CF) is that 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 recommendations become [33]. Usually, 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 users 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 [33]. 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 directly uses 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. To mathematically formulate the problem, let us assume there is a list of users $U = \{u_1, u_2, ..., u_m\}$ and a list of items $I = \{i_i, i_2, ..., i_n\}$. 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:

$$\overline{v}_i = \frac{1}{|I_i|} \sum_{j \in I_i} v_{i,j} \tag{1}$$

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

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

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 [37]:

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

where k is the number of most similar items (nearest neighbors) to j that the target user a has rated in the past. Among other popular metrics, which are often used to calculate the similarity between users, or items, we may mention cosine similarity or Pearson correlation [26]. 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]:

$$M = U \Sigma V^T \tag{4}$$

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].

3.2. 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 [27].

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 $I = \{i_1, i_2, ..., i_m\}$ 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 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. Equation 5 shows how we calculate the support for an association rule.

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

On the other hand, confidence 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.

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

Finally, the lift is a correlation measure used to discover and exclude the weak rules that have high confidence. Equation 7 shows that the lift measure is calculated by dividing the confidence by the unconditional probability of the consequent [27].

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

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, AprioriTID, Apriori Hybrid, and FP-growth (Frequent pattern) [52]. In this paper, we employ the 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.

4. Frequent Pattern Mining Framework For Recommender Systems

The main problem we address in this paper is to alleviate the impact of new items cold-start in recommender systems based on collaborative filtering techniques. These methods suffer from the cold-start problem whenever a new user joins the system or 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).

The FPRS 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) Data Preprocessing, (iv) Frequent Pattern Mining, and (v) 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 two 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 their validity for generating the recommendation. In the third stage, we apply cluster attributes with more than one value associated with the object's key. The main objective of this step is to convert multi-valued attributes into single-value ones with the cluster id so that the dataset can be split based on their values. In the analyzed case study, we applied agglomerative clustering techniques on the 'genre' since each movie can be assigned to more than one genre. After that, we split the dataset for each selected attribute. In the fourth stage, we generate frequent itemsets using the FP-Growth algorithm. Finally, we produce the recommendations in the last stage. The developed item cold-start module in the FPRS framework contains several strategies to select the features and produce recommendations. More details about these strategies will be provided later in the next section.

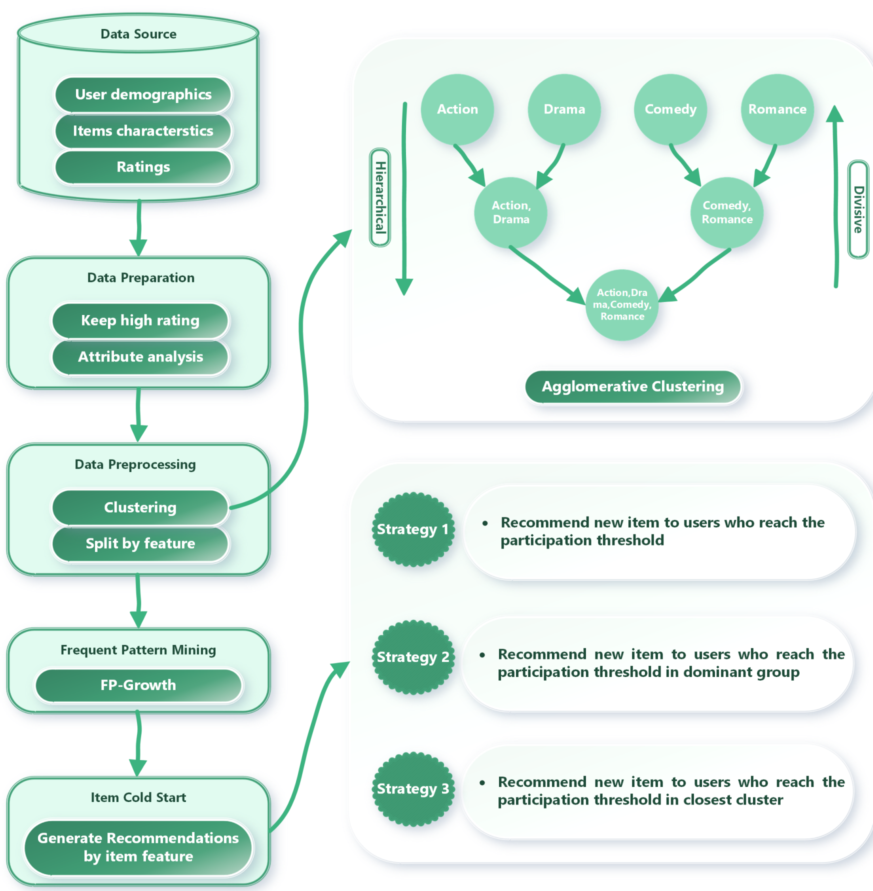


Fig. 1. Frequent Pattern Mining Framework For Recommender Systems

In the item cold-start 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. 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 item cold-start module are provided in Strategy 1, Strategy 2, and Strategy 3.

In Strategy 1, we split the data only by items' features, while in Strategy 2. we utilize both items' and users' characteristics to split the data (cf. line 1 in both strategies). Moreover, in Strategy 1, we select the users that compose the recommendation group based on their engagement in creating the frequent itemsets (cf. line 5 in Strategy 1), whereas in Strategy 2, the representatives of recommendation group are selected based on items' features, to find the so-called dominant group, in addition to the percentage of users' engagement in creating the frequent itemsets (cf. lines 5 and 7 in Strategy 2). On the other hand, in Strategy 3, we form clusters based on the frequent 1-itemsets created by users' and items' features (cf. line 4 in Strategy 3). Then, the recommendation group for the new item is created based on the percentage of users' engagement in creating the frequent itemsets in the closest cluster (cf. line 7 in Strategy 3).

All the thresholds used in the above-described strategies are selected by objectively searching for an optimal set of values, i.e., that achieves the best performance on a given dataset. More details on how we choose these values are provided in Section 5.

Strategy 1 Item Cold-Start Module (Items Characteristics Split)

1: Split the records based on items characteristics (i.e., genre)
2: Generate frequent itemsets {support > 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 2 Item Cold-Start Module (Users/Items Characteristics Split)

1: Split the records based on users/items characteristics (i.e., gender and genre)
2: Generate frequent itemsets {support > 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 belonging to the dominant gender who involved in creating larger than the participation threshold of frequent itemsets

Strategy 3 Item Cold-Start Module (Clustering-based)

1: Split the records based on users/items characteristics (i.e., gender and genre)
2: Generate frequent 1-itemsets {support > min_support}
3: Set the participation percentage threshold
4: For each genre: 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

5. Evaluation Methodology

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

5.1. Dataset and Evaluation Measures

In our experiments, we used three datasets (MovieLens 100K, MovieLens 1M [1] and LDOS-CoMoDa [2]). MovieLens datasets were collected by the GroupLens research project at the University of Minnesota. MovieLens 100K contains 100,000 ratings given by 943 users on 1,682 movies on a scale from 1 to 5. In comparison, MovieLens 1M contains 1,000,000 ratings of approximately 3,900 movies made by 6,040 users on a scale from 1 to 5. On the other hand, LDOS-CoMoDa is a context-rich movie recommender dataset that consists of 200 users who gave 2,296 ratings for 4,138 movies in twelve pieces of contextual information describing the situation in which the user consumed the item. This dataset is collected from real user-item interactions and not from the hypothetical situation or user's memory of past interactions.

In all datasets, we combine three files (*users.data*, *items.data*, *ratings.data*) in order to join users' characteristics (e.g., demographics), items' attributes, and ratings in one dataset. The final/joined dataset contains userId, itemId, rating, gender, age, occupation, and genre attributes (cf. Table 1). Moreover, we performed further analysis of some features we used in our experiments (i.e., gender and genre) to understand the interrelation between these features and their potential impact on the obtained results. Figures 2 show the most popular movie genres among males and females for all datasets.

After preprocessing the data and removing invalid records, we split it into training and testing datasets. To properly evaluate the item cold-start module, we first find the 50 most active users. Then, we select some of the most-rated movies by those 50 users. The ratings of all those movies by all users (7,320 records in MovieLens 100K, 41,105 records in MovieLens 1M, and 126 records in CoMoDa) are considered as testing set, keeping the rest of the records in the training set. This way, we may ensure enough interactions to assess quality reliably in the testing phase. Note that all the ratings for the selected movies are removed from the training dataset, which corresponds to the item cold-start.

[1] https://grouplens.org/datasets/movielens/
[2] https://www.lucami.org/en/research/ldos-comoda-dataset/

Table 1. Selected data characteristics.

Attribute Name	Data Type	Value Range (MovieLens 100K)	Value Range (MovieLens 1M)	Value Range (LDOS-CoMoDa)
gender	Character	M-F	M-F	M-F
age	Number	Under 18-73	Under 18-56	15-63
occupation	Text	21 occupation	21 occupation	NA
genres	Text	19 genres	19 genres	25 genres

In our study, we consider a binary decision task whether a given item (i.e., a movie) is appropriate for the user. To correctly model this situation, we assume that films rated by users 3 or more 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 item is binary information: to recommend or not. Following that, in order to assess the quality of the prediction, the F_1 measure is used [66].

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{8}$$

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

$$Precision = \frac{TP}{TP + FP} \tag{9}$$

$$Recall = \frac{TP}{TP + FN} \tag{10}$$

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.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{11}$$

5.2. Baseline Methods

To showcase the strengths of FPRS, we evaluate it against two baseline models designed for the cold-start recommender system. The first one [7], called regularization differentiating functions (RDF), was proposed recently to address the cold-start problem by extending the matrix factorization-based models using three simple regularization differentiating functions (RDF). In particular, these functions assign lower regularization weights to the latent factors associated with popular items and active users, and set higher regularization weights on long tail items, that were rated/viewed/purchased by few users, and less active users. The goal of this method is to enhance the MF-based models by utilizing more information revealed by popular items and active users, and make conservative predictions on long tail items and less active users. In the evaluation, we utilized the publicly available

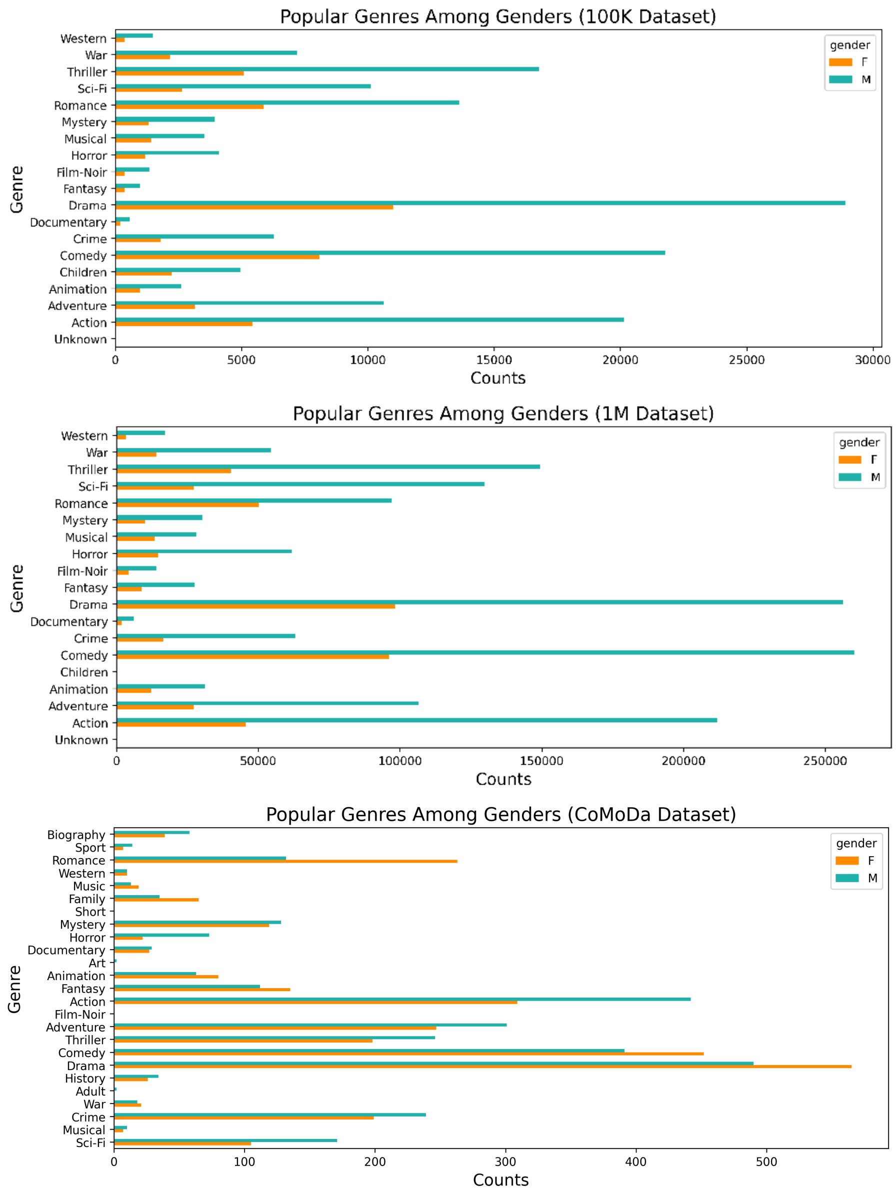


Fig. 2. Histogram of the variables (rating, genre, and gender) in MovieLens (100K and 1M) and LDOS-CoMoDa datasets.

implementation [1], provided by the authors, which integrates the proposed regularization technique with the SVD, SVD++, and the NMF models. On the other hand, in order to

[1] https://github.com/ncu-dart/rdf

demonstrate the efficiency of frequent pattern mining in RS, we build a baseline model in a similar way to the FPRS strategies, yet the FP-growth algorithm was omitted. In order to evaluate the item module of FPRS, we find the most popular (watched) genre for each user. Then each new movie is recommended to all users whose favorite genre is the same as the new movie's.

5.3. Performance Comparison and Analysis

In order to provide a fair comparison, we use precision, recall, F_1-score, and accuracy measures to compare the performance of FPRS against two RSs, namely Baseline and RDF, which are described in Section 5.2. After splitting the dataset into the training and testing sets and training the baseline and FPRS recommendation systems, we run multiple experiments to evaluate the item cold-start module.

In these experiments, we evaluated the item cold-start module in FPRS. We calculated precision, recall, F_1, and accuracy measures for the results generated by Baseline, RDF, and FPRS following all the strategies described in Section 4. The comparative summary of this evaluation is shown in Tables 2, 3, and 4. The results show that the performance of FPRS, using all strategies, is superior to Baseline and RDF solutions. However, the results differ slightly between datasets.

For MovieLens 100K, all strategies reported similar recall. Regarding precision and F_1 measures, the most successful in dealing with new items in this data appeared to be Strategy 2, 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 3, which was also superior in terms of recall, F_1, and accuracy on the second dataset (i.e., MovieLens 1M). When it comes to the LDOS-CoMoDa dataset, we notice that Strategy 1 is the most successful in terms of recall and F_1 measures. Whereas, Strategy 2 achieved the highest accuracy, while strategy 3 reported the best precision. When it comes to Baseline and RDF models, they reported relatively similar F_1 scores. However, RDF outperformed Baseline in terms of recall, while Baseline achieved higher accuracy than RDF.

It is important to emphasize that there is no absolute superior strategy, but the appropriate approach should be selected based on the application's requirements. For example, when we deal with small datasets, like LDOS-CoMoDa, it is recommended to use strategies that keep enough records in the partitioned datasets to extract frequent itemsets. This is observed while evaluating FPRS using the LDOS-CoMoDa dataset (cf. Table 4). The results show that Strategy 1, which splits the data only based on items' features, is superior in terms of recall and F_1 measures.

However, selecting the appropriate evaluation measure plays an important role while choosing the proper strategy. The precision measure is focusing on the number of correct recommendations considering the mistakes made. On the other hand, the recall measure does not take into account the mistakes made since it only considers the number of correct recommendations made out of all positive predictions that could have been made (cf. Equations 9 and 10). According to the previous, the proper strategy can be selected relying on the evaluation measure that best matches the target of our application.

Finally, all strategies were evaluated at the participation threshold value of 30% and $min\_support$ value of 0.2 for MovieLens 1M and 100K. Regarding LDOS-CoMoDa, the strategies were evaluated at the participation threshold value of 20% and $min\_support$

value of 0.07. The values of the thresholds are determined based on sensitivity analysis experiments which we will be presented in Section 5.4.

Table 2. Evaluation for item cold-start (MovieLens 100k).

Strategy	Precision	Recall	F1-score	Accuracy
Baseline	0.38	0.3	0.32	0.59
RDF	0.35	0.51	0.41	0.48
Strategy 1	0.69	0.64	0.66	0.75
Strategy 2	0.79	0.62	0.69	0.84
Strategy 3	0.67	0.63	0.63	0.86

Table 3. Evaluation for item cold-start (MovieLens 1M).

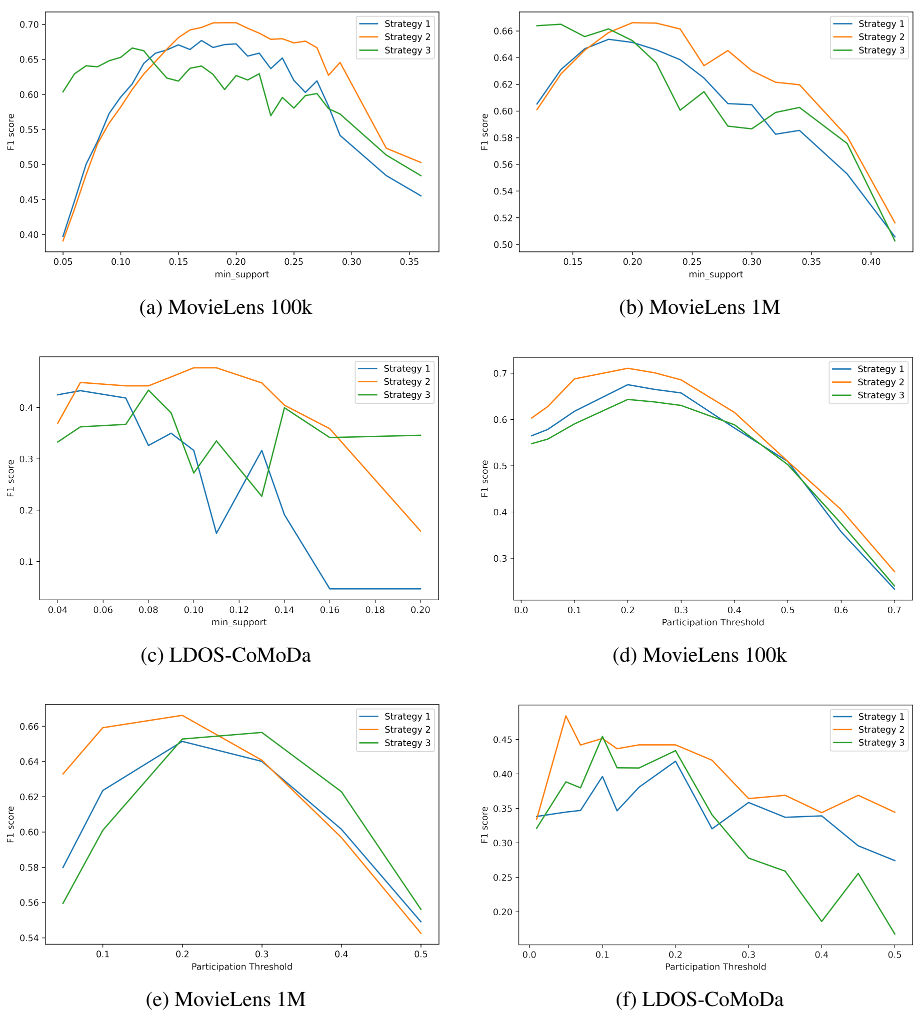
Strategy	Precision	Recall	F1-score	Accuracy
Baseline	0.49	0.43	0.43	0.64
RDF	0.34	0.52	0.39	0.51
Strategy 1	0.65	0.71	0.64	0.76
Strategy 2	0.64	0.73	0.66	0.83
Strategy 3	0.6	0.79	0.66	0.86

Table 4. Evaluation for item cold-start (LDOS-CoMoDa).

Strategy	Precision	Recall	F1-score	Accuracy
Baseline	0.02	0.21	0.10	0.73
RDF	0.07	0.59	0.13	0.42
Strategy 1	0.36	0.62	0.40	0.81
Strategy 2	0.36	0.55	0.38	0.96
Strategy 3	0.4	0.21	0.26	0.86

5.4. Thresholds Sensitivity Analysis

In the 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 items. In this section, we conduct some experiments to show how changing those values may impact the 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.



(a) MovieLens 100k (b) MovieLens 1M

(c) LDOS-CoMoDa (d) MovieLens 100k

(e) MovieLens 1M (f) LDOS-CoMoDa

Fig. 3. Sensitivity analysis for min_supp and participation percentage thresholds in MovieLens (100k and 1M) and LDOS-CoMoDa.

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 F_1-score of FPRS is impacted by applying different values for MovieLens 100K data. Observably, the best min_supp values for all strategies used in FPRS (item cold-start module) are between 0.1 and 0.2. A similar analysis, yet regarding MovieLens 1M and LDOS-CoMoDa data, we may find in Figures 3b and 3c respectively.

In the second experiment, we search for the optimal value of the participation threshold in FPRS (item cold-start module). Figures 3d, 3e and 3f show how the F_1-score of FPRS is impacted by applying different values for all investigated datasets. Observably, the best participation threshold values for all strategies used in FPRS (item cold-start module) are between 15% and 30% for MovieLens 1M and 100K data, and it is between 5% and 20% for LDOS-CoMoDa data. Finally, it is worth noting that when we run this experiment, we use the optimal value of min_supp we found in the previous experiment.

6. Challenges, Limitations and Future Works

This section is dedicated to discussing the challenges we faced while evaluating FPRS and suggesting the best practices to overcome the revealed limitations.

One of these challenges appears when we deal with small datasets. In such scenarios, we might not be able to generate frequent patterns after splitting out the dataset. To address this issue, we have to carefully select appropriate values for $min\_support$ and $participation$ percentage thresholds by analyzing the results of the experiments presented in Section 5.4.

Another limitation that FPRS may suffer from is the lack of users' and items' characteristics. For future research, we find it very promising to investigate further the application of granular methods in this respect [19, 21]. Here, it would be advisable to incorporate the methods to improve the resilience of the FPRS framework to data discrepancies or loss [23]. The alternative approach could investigate the applicability of dimensionality reduction methods, like PCA, in data modeling [34].

In the future, we plan to investigate more algorithms for association rule and frequent pattern mining, e.g., AprioriTID or Apriori Hybrid. It is also interesting to incorporate more contextual information. We also plan to respond better to changes in users' behavior and preferences, addressing the possible drifts and shifts in data. One viable option is to periodically update 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 [18, 24, 84].

7. Conclusions

This article presents FPRS, a 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 items. Our study evaluates multiple strategies for creating frequent itemsets to produce meaningful and relevant recommendations. The presented pipeline also includes the clustering-based feature extraction phase, which aims at modeling similarities between investigated entities. This way, we not only extend the data representation but also increase their density, allowing us to alleviate the omnipresent problem of too few interactions in historical data, especially severe for new products or services.

To evaluate FPRS, we conducted experiments on MovieLens (100K and 1M) and LDOS-CoMoDa datasets with the FP-growth algorithm to generate the frequent itemsets. The experimental results show that FPRS has outperformed two state-of-the-art models, designed to address the cold-start problem, in terms of precision, recall, F_1, and accuracy measures.

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

References

1. Afoudi, Y., Lazaar, M., Al Achhab, M.: Hybrid recommendation system combined content-based filtering and collaborative prediction using artificial neural network. Simulation Modelling Practice and Theory 113, 102375 (2021)
2. Alamdari, P.M., Navimipour, N.J., Hosseinzadeh, M., Safaei, A.A., Darwesh, A.: A Systematic Study on the Recommender Systems in the E-Commerce. IEEE Access 8, 115694–115716 (2020)
3. Asid, M., Ali, R.: Use of Soft Computing Techniques for Recommender Systems: An Overview, pp. 61–80. Springer Singapore, Singapore (2017)
4. Barria-Pineda, J., Akhuseyinoglu, K., Zelem-Celap, S., Brusilovsky, P., Klasnja-Milicevic, A., Ivanovic, M.: Explainable recommendations in a personalized programming practice system. In: Roll, I., McNamara, D.S., Sosnovsky, S.A., Luckin, R., Dimitrova, V. (eds.) Artificial Intelligence in Education - 22nd International Conference, AIED 2021, Utrecht, The Netherlands, June 14-18, 2021, Proceedings, Part I. Lecture Notes in Computer Science, vol. 12748, pp. 64–76. Springer (2021)
5. Batmaz, Z., Yürekli, A., Bilge, A., Kaleli, C.: A review on deep learning for recommender systems: challenges and remedies. Artif. Intell. Rev. 52(1), 1–37 (2019)
6. Baydogmus, G.K.: Solution for TSP/MTSP with an improved parallel clustering and elitist ACO. Computer Science and Information Systems 20(1), 195—-214 (2023), https://doi.org/10.2298/CSIS220820053B
7. Chen, H.H., Chen, P.: Differentiating Regularization Weights – A Simple Mechanism to Alleviate Cold Start in Recommender Systems. ACM Trans. Knowl. Discov. Data 13(1) (jan 2019), https://doi.org/10.1145/3285954
8. Chen, R., Hua, Q., Chang, Y.S., Wang, B., Zhang, L., Kong, X.: A Survey of Collaborative Filtering-Based Recommender Systems: From Traditional Methods to Hybrid Methods Based on Social Networks. IEEE Access 6, 64301–64320 (2018)
9. Choi, S.M., Jang, K., Lee, T.D., Khreishah, A., Noh, W.: Alleviating Item-Side Cold-Start Problems in Recommender Systems Using Weak Supervision. IEEE Access 8, 167747–167756 (2020)
10. Cui, Z., Xu, X., XUE, F., Cai, X., Cao, Y., Zhang, W., Chen, J.: Personalized Recommendation System Based on Collaborative Filtering for IoT Scenarios. IEEE Transactions on Services Computing 13(4), 685–695 (2020)
11. Feng, J., Xia, Z., Feng, X., Peng, J.: RBPR: A hybrid model for the new user cold start problem in recommender systems. Knowledge-Based Systems 214, 106732 (2021)
12. Feng, W., Zhu, Q., Zhuang, J., Yu, S.: An expert recommendation algorithm based on Pearson correlation coefficient and FP-growth. Clust. Comput. 22(Supplement), 7401–7412 (2019)

13. de Gemmis, M., Lops, P., Musto, C., Narducci, F., Semeraro, G.: Semantics-Aware Content-Based Recommender Systems, pp. 119–159. Springer US, Boston, MA (2015), https://doi.org/10.1007/978-1-4899-7637-6_4
14. Ghazinour, K., Matwin, S., Sokolova, M.: Monitoring and recommending privacy settings in social networks. In: Guerrini, G. (ed.) Joint 2013 EDBT/ICDT Conferences, EDBT/ICDT '13, Genoa, Italy, March 22, 2013, Workshop Proceedings. pp. 164–168. ACM (2013)
15. Goy, S., Coors, V., Finn, D.: Grouping techniques for building stock analysis: A comparative case study. Energy and Buildings 236, 110754 (2021)
16. Grzegorowski, M.: Scaling of complex calculations over big data-sets. In: Slezak, D., Schaefer, G., Vuong, S.T., Kim, Y. (eds.) Active Media Technology - 10th International Conference, AMT 2014, Warsaw, Poland, August 11-14, 2014. Proceedings. Lecture Notes in Computer Science, vol. 8610, pp. 73–84. Springer (2014), https://doi.org/10.1007/978-3-319-09912-5_7
17. Grzegorowski, M.: Governance of the Redundancy in the Feature Selection Based on Rough Sets' Reducts. In: Flores, V., Gomide, F.A.C., Janusz, A., Meneses, C., Miao, D., Peters, G., Ślęzak, D., Wang, G., Weber, R., Yao, Y. (eds.) Rough Sets - International Joint Conference, IJCRS 2016, Santiago de Chile, Chile, October 7-11, 2016, Proceedings. Lecture Notes in Computer Science, vol. 9920, pp. 548–557 (2016), https://doi.org/10.1007/978-3-319-47160-0_50
18. Grzegorowski, M.: Massively Parallel Feature Extraction Framework Application in Predicting Dangerous Seismic Events. In: Ganzha, M., Maciaszek, L.A., Paprzycki, M. (eds.) Proceedings of the 2016 Federated Conference on Computer Science and Information Systems, FedCSIS 2016, Gdańsk, Poland, September 11-14, 2016. Annals of Computer Science and Information Systems, vol. 8, pp. 225–229. IEEE (2016), https://doi.org/10.15439/2016F90
19. Grzegorowski, M.: Selected aspects of interactive feature extraction. Trans. Rough Sets 23, 121–287 (2023), https://doi.org/10.1007/978-3-662-66544-2_8
20. Grzegorowski, M., Janusz, A., Lazewski, S., Swiechowski, M., Jankowska, M.: Prescriptive analytics for optimization of FMCG delivery plans. In: Ciucci, D., Couso, I., Medina, J., Ślęzak, D., Petturiti, D., Bouchon-Meunier, B., Yager, R.R. (eds.) Information Processing and Management of Uncertainty in Knowledge-Based Systems - 19th International Conference, IPMU 2022, Milan, Italy, July 11-15, 2022, Proceedings, Part II. Communications in Computer and Information Science, vol. 1602, pp. 44–53. Springer (2022)
21. Grzegorowski, M., Janusz, A., Ślęzak, D., Szczuka, M.S.: On the role of feature space granulation in feature selection processes. In: Nie, J., Obradovic, Z., Suzumura, T., Ghosh, R., Nambiar, R., Wang, C., Zang, H., Baeza-Yates, R., Hu, X., Kepner, J., Cuzzocrea, A., Tang, J., Toyoda, M. (eds.) 2017 IEEE International Conference on Big Data (IEEE BigData 2017), Boston, MA, USA, December 11-14, 2017. pp. 1806–1815. IEEE Computer Society (2017)
22. Grzegorowski, M., Litwin, J., Wnuk, M., Pabis, M., Marcinowski, L.: Survival-based feature extraction - application in supply management for dispersed vending machines. IEEE Transactions on Industrial Informatics 19(3), 3331–3340 (2023)
23. Grzegorowski, M., Ślęzak, D.: On resilient feature selection: Computational foundations of r-C-reducts. Inf. Sci. 499, 25–44 (2019)
24. Grzegorowski, M., Stawicki, S.: Window-based feature extraction framework for multi-sensor data: A posture recognition case study. In: Ganzha, M., Maciaszek, L.A., Paprzycki, M. (eds.) 2015 Federated Conference on Computer Science and Information Systems, FedCSIS 2015, Lódz, Poland, September 13-16, 2015. Annals of Computer Science and Information Systems, vol. 5, pp. 397–405. IEEE (2015)
25. Guo, G., Qiu, H., Tan, Z., Liu, Y., Ma, J., Wang, X.: Resolving data sparsity by multi-type auxiliary implicit feedback for recommender systems. Knowl. Based Syst. 138, 202–207 (2017)
26. Hamedani, M.R., Ali, I., Hong, J., Kim, S.W.: Trustrec: An effective approach to exploit implicit trust and distrust relationships along with explicitones for accurate recommendations. Comput. Sci. Inf. Syst. 18, 93–114 (2021)

27. Han, J., Kamber, M., Pei, J.: 6 - mining frequent patterns, associations, and correlations: Basic concepts and methods. In: Han, J., Kamber, M., Pei, J. (eds.) Data Mining (Third Edition), pp. 243–278. The Morgan Kaufmann Series in Data Management Systems, Morgan Kaufmann, Boston, third edition edn. (2012), https://www.sciencedirect.com/science/article/pii/B978012381479100006X
28. Heidari, N., Moradi, P., Koochari, A.: An attention-based deep learning method for solving the cold-start and sparsity issues of recommender systems. Knowl. Based Syst. 256, 109835 (2022)
29. Himeur, Y., Alsalemi, A., Al-Kababji, A., Bensaali, F., Amira, A., Sardianos, C., Dimitrakopoulos, G., Varlamis, I.: A survey of recommender systems for energy efficiency in buildings: Principles, challenges and prospects. Information Fusion 72, 1–21 (2021)
30. Jalili, M., Ahmadian, S., Izadi, M., Moradi, P., Salehi, M.: Evaluating Collaborative Filtering Recommender Algorithms: A Survey. IEEE Access 6, 74003–74024 (2018)
31. Janusz, A., Grzegorowski, M., Michalak, M., Wróbel, Ł., Sikora, M., Ślęzak, D.: Predicting Seismic Events in Coal Mines Based on Underground Sensor Measurements. Engineering Applications of Artificial Intelligence 64, 83–94 (2017)
32. Kajan, E., Faci, N., Maamar, Z., Sellami, M., Ugljanin, E., Kheddouci, H., Stojanovic, D., Benslimane, D.: Real-time tracking and mining of users' actions over social media. Comput. Sci. Inf. Syst. 17(2), 403–426 (2020)
33. Kannout, E.: Context clustering-based recommender systems. In: 2020 15th Conference on Computer Science and Information Systems (FedCSIS). pp. 85–91 (2020)
34. Kannout, E., Grodzki, M., Grzegorowski, M.: Considering various aspects of models' quality in the quality pipeline - application in the logistics sector. In: Ganzha, M., Maciaszek, L.A., Paprzycki, M., Slezak, D. (eds.) Proceedings of the 17th Conference on Computer Science and Intelligence Systems, FedCSIS 2022, Sofia, Bulgaria, September 4-7, 2022. Annals of Computer Science and Information Systems, vol. 30, pp. 403–412 (2022)
35. Kannout, E., Grodzki, M., Grzegorowski, M.: Utilizing Frequent Pattern Mining for Solving Cold-Start Problem in Recommender Systems. In: Ganzha, M., Maciaszek, L.A., Paprzycki, M., Ślęzak, D. (eds.) Proceedings of the 17th Conference on Computer Science and Intelligence Systems, FedCSIS 2022, Sofia, Bulgaria, September 4-7, 2022. Annals of Computer Science and Information Systems, vol. 30, pp. 217–226 (2022)
36. Kannout, E., Nguyen, H.S., Grzegorowski, M.: Speeding up recommender systems using association rules. In: Intelligent Information and Database Systems, pp. 167–179. Springer Nature Switzerland (2022), https://doi.org/10.1007/978-3-031-21967-2_14
37. Kannout, E., Nguyen, H.S., Grzegorowski, M.: Efficient Techniques to Recommender System. Springer Cham (2023), revised Selected Papers from the 29th International Workshop on Concurrency, Specification and Programming (CS&P'21), Berlin, Germany
38. Karpus, A., Raczynska, M., Przybylek, A.: Things You Might Not Know about the k-Nearest Neighbors Algorithm. In: Fred, A.L.N., Filipe, J. (eds.) Proceedings of the 11th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management, IC3K 2019, Volume 1: KDIR, Vienna, Austria, September 17-19, 2019. pp. 539–547. ScitePress (2019), https://doi.org/10.5220/0008365005390547
39. Karpus, A., Vagliano, I., Goczyla, K.: Serendipitous recommendations through ontology-based contextual pre-filtering. In: Kozielski, S., Mrozek, D., Kasprowski, P., Malysiak-Mrozek, B., Kostrzewa, D. (eds.) 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. Communications in Computer and Information Science, vol. 716, pp. 246–259 (2017)
40. Karpus, A., Vagliano, I., Goczyla, K., Morisio, M.: An ontology-based contextual pre-filtering technique for recommender systems. In: Ganzha, M., Maciaszek, L.A., Paprzycki, M. (eds.) Proceedings of the 2016 Federated Conference on Computer Science and Information Systems, FedCSIS 2016, Gdańsk, Poland, September 11-14, 2016. Annals of Computer Science and Information Systems, vol. 8, pp. 411–420. IEEE (2016)

41. Kashef, R.: Enhancing the Role of Large-Scale Recommendation Systems in the IoT Context. IEEE Access 8, 178248–178257 (2020)
42. Kawai, M., Sato, H., Shiohama, T.: Topic model-based recommender systems and their applications to cold-start problems. Expert Systems with Applications 202, 117129 (2022)
43. Khan, Z., Hussain, M.I., Iltaf, N., Kim, J., Jeon, M.: Contextual recommender system for e-commerce applications. Applied Soft Computing 109, 107552 (2021)
44. Kolahkaj, M., Harounabadi, A., Nikravanshalmani, A., Chinipardaz, R.: Incorporating multidimensional information into dynamic recommendation process to cope with cold start and data sparsity problems. J. Ambient Intell. Humaniz. Comput. 12(10), 9535–9554 (2021)
45. Kwasnicka, H., Ovedenski, T.: Pix2trips - a system supporting small groups of urban tourists. In: Ganzha, M., Maciaszek, L.A., Paprzycki, M., Ślęzak, D. (eds.) Proceedings of the 16th Conference on Computer Science and Intelligence Systems, Online, September 2-5, 2021. Annals of Computer Science and Information Systems, vol. 25, pp. 141–145 (2021)
46. Kwiecinski, R., Gorecki, T., Filipowska, A.: Learning edge importance in bipartite graph-based recommendations. In: 2022 17th Conference on Computer Science and Intelligence Systems (FedCSIS). pp. 227–233 (2022)
47. Lee, H., Im, J., Jang, S., Cho, H., Chung, S.: MeLU: Meta-Learned User Preference Estimator for Cold-Start Recommendation. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. p. 1073–1082. KDD '19, Association for Computing Machinery, New York, NY, USA (2019)
48. Lee, S., Kim, S., Park, S., Chae, D.: A tripartite-graph based recommendation framework for price-comparison services. Comput. Sci. Inf. Syst. 16(2), 333–357 (2019)
49. Lika, B., Kolomvatsos, K., Hadjiefthymiades, S.: Facing the cold start problem in recommender systems. Expert Systems with Applications 41(4, Part 2), 2065–2073 (2014)
50. Liu, F., Tang, R., Li, X., Zhang, W., Ye, Y., Chen, H., Guo, H., Zhang, Y.: Deep Reinforcement Learning based Recommendation with Explicit User-Item Interactions Modeling (2018)
51. Lu, Y., Fang, Y., Shi, C.: Meta-learning on heterogeneous information networks for cold-start recommendation. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery &; Data Mining. p. 1563–1573. KDD '20, Association for Computing Machinery, New York, NY, USA (2020), https://doi.org/10.1145/3394486.3403207
52. Luna, J.M., Fournier-Viger, P., Ventura, S.: Frequent itemset mining: A 25 years review. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 9(6), e1329 (2019)
53. Merabet, F.Z., Benmerzoug, D.: QoS prediction for service selection and recommendation with a deep latent features autoencoder. Comput. Sci. Inf. Syst. 19(2), 709–733 (2022), https://doi.org/10.2298/csis210518054m
54. Najafabadi, M.K., Mohamed, A.H., Mahrin, M.N.: A survey on data mining techniques in recommender systems. Soft Comput. 23(2), 627–654 (2019)
55. Natarajan, S., Vairavasundaram, S., Natarajan, S., Gandomi, A.H.: Resolving data sparsity and cold start problem in collaborative filtering recommender system using linked open data. Expert Systems with Applications 149, 113248 (2020)
56. Nilashi, M., Ibrahim, O., Bagherifard, K.: A recommender system based on collaborative filtering using ontology and dimensionality reduction techniques. Expert Systems with Applications 92, 507–520 (2018)
57. Obeid, C., Lahoud, C., El, K.H., Champin, P.A.: A novel hybrid recommender system approach for student academic advising named COHRS, supported by case-based reasoning and ontology. Computer Science and Information Systems 19(2), 979–1005 (2022), https://doi.org/10.2298/CSIS220215011O
58. Odic, A., Tkalcic, M., Tasic, J.F., Kosir, A.: Predicting and detecting the relevant contextual information in a movie-recommender system. Interact. Comput. 25(1), 74–90 (2013)
59. Papadakis, H., Papagrigoriou, A., Panagiotakis, C., Kosmas, E., Fragopoulou, P.: Collaborative filtering recommender systems taxonomy. Knowl. Inf. Syst. 64(1), 35–74 (2022)

60. Pasricha, R., McAuley, J.: Translation-Based Factorization Machines for Sequential Recommendation. In: Proceedings of the 12th ACM Conference on Recommender Systems. p. 63–71. RecSys'18, Association for Computing Machinery, New York, NY, USA (2018)
61. Pawlicka, A., Pawlicki, M., Kozik, R., Choraś, R.S.: A systematic review of recommender systems and their applications in cybersecurity. Sensors 21(15) (2021), `https://www.mdpi.com/1424-8220/21/15/5248`
62. Pondel, M., Korczak, J.: Collective clustering of marketing data - recommendation system upsaily. In: Ganzha, M., Maciaszek, L.A., Paprzycki, M. (eds.) Proceedings of the 2018 Federated Conference on Computer Science and Information Systems, FedCSIS 2018, Poznań, Poland, September 9-12, 2018. Annals of Computer Science and Information Systems, vol. 15, pp. 801–810 (2018)
63. Pulis, M., Bajada, J.: Siamese Neural Networks for Content-Based Cold-Start Music Recommendation., p. 719–723. Association for Computing Machinery, New York, NY, USA (2021)
64. Pérez-Almaguer, Y., Yera, R., Alzahrani, A.A., Martínez, L.: Content-based group recommender systems: A general taxonomy and further improvements. Expert Systems with Applications 184, 115444 (2021)
65. Sikora, M., Matyszok, P., Wróbel, L.: SCARI: separate and conquer algorithm for action rules and recommendations induction. Inf. Sci. 607, 849–868 (2022)
66. Silveira, T., Zhang, M., Lin, X., Liu, Y., Ma, S.: How good your recommender system is? A survey on evaluations in recommendation. International Journal of Machine Learning and Cybernetics 10, 813–831 (2019)
67. Singh, M.: Scalability and sparsity issues in recommender datasets: a survey. Knowl. Inf. Syst. 62(1), 1–43 (2020)
68. Sobhanam, H., Mariappan, A.K.: Addressing cold start problem in recommender systems using association rules and clustering technique. In: 2013 International Conference on Computer Communication and Informatics. pp. 1–5 (2013)
69. Sofikitis, E., Makris, C.: Development of recommendation systems using game theoretic techniques. Computer Science and Information Systems 19(00), 1133–1154 (2022)
70. Steck, H.: Collaborative filtering via high-dimensional regression. CoRR abs/1904.13033 (2019)
71. Sulc, Z., Rezanková, H.: Comparison of similarity measures for categorical data in hierarchical clustering. J. Classif. 36(1), 58–72 (2019), `https://doi.org/10.1007/s00357-019-09317-5`
72. Tilahun, Z., Jun, H., Oad, A.: Solving cold-start problem by combining personality traits and demographic attributes in a user based recommender system. International Journal of Advanced Research in Computer Science and Software Engineering 7(5), 231–239 (may 2017)
73. Tsagkias, M., King, T.H., Kallumadi, S., Murdock, V., de Rijke, M.: Challenges and Research Opportunities in ECommerce Search and Recommendations. SIGIR Forum 54(1) (feb 2021)
74. Valiente, J.A.R., Merino, P.J.M., Díaz, H.J.P., Ruiz, J.S., Kloos, C.D.: Evaluation of a learning analytics application for open edX platform. Comput. Sci. Inf. Syst. 14(1), 51–73 (2017), `https://doi.org/10.2298/CSIS160331043R`
75. Vančura, V., Alves, R., Kasalický, P., Kordík, P.: Scalable Linear Shallow Autoencoder for Collaborative Filtering. In: Proceedings of the 16th ACM Conference on Recommender Systems. p. 604–609. RecSys '22, Association for Computing Machinery, New York, NY, USA (2022)
76. Viktoratos, I., Tsadiras, A., Bassiliades, N.: Combining community-based knowledge with association rule mining to alleviate the cold start problem in context-aware recommender systems. Expert Systems with Applications 101, 78–90 (2018)
77. Vizine Pereira, A.L., Hruschka, E.R.: Simultaneous co-clustering and learning to address the cold start problem in recommender systems. Knowledge-Based Systems 82, 11–19 (2015)
78. Walek, B., Fojtik, V.: A hybrid recommender system for recommending relevant movies using an expert system. Expert Systems with Applications 158, 113452 (2020)

79. Wei, Y., Wang, X., Li, Q., Nie, L., Li, Y., Li, X., Chua, T.S.: Contrastive Learning for Cold-Start Recommendation. In: Proceedings of the 29th ACM International Conference on Multimedia. p. 5382–5390. MM '21, Association for Computing Machinery, New York, NY, USA (2021), https://doi.org/10.1145/3474085.3475665
80. Wu, H., Zhang, Z., Yue, K., Zhang, B., He, J., Sun, L.: Dual-regularized matrix factorization with deep neural networks for recommender systems. Knowledge-Based Systems 145, 46–58 (2018)
81. Yan, H.C., Wang, Z.R., Niu, J.Y., Xue, T.: Application of covering rough granular computing model in collaborative filtering recommendation algorithm optimization. Advanced Engineering Informatics 51, 101485 (2022)
82. Yeh, J.Y., Tsai, C.J.: A graph-based feature selection method for learning to rank using spectral clustering for redundancy minimization and biased PageRank for relevance analysis. Computer Science and Information Systems 19(1), 141–164 (2022)
83. Yi, B., Shen, X., Liu, H., Zhang, Z., Zhang, W., Liu, S., Xiong, N.: Deep Matrix Factorization With Implicit Feedback Embedding for Recommendation System. IEEE Transactions on Industrial Informatics 15(8), 4591–4601 (2019)
84. Zdravevski, E., Lameski, P., Mingov, R., Kulakov, A., Gjorgjevikj, D.: Robust histogram-based feature engineering of time series data. In: Ganzha, M., Maciaszek, L.A., Paprzycki, M. (eds.) 2015 Federated Conference on Computer Science and Information Systems, FedCSIS 2015, Lódz, Poland, September 13-16, 2015. Annals of Computer Science and Information Systems, vol. 5, pp. 381–388. IEEE (2015), https://doi.org/10.15439/2015F420
85. Zhang, F., Qi, S., Liu, Q., Mao, M., Zeng, A.: Alleviating the data sparsity problem of recommender systems by clustering nodes in bipartite networks. Expert Syst. Appl. 149, 113346 (2020), https://doi.org/10.1016/j.eswa.2020.113346
86. Zhu, Y., Xie, R., Zhuang, F., Ge, K., Sun, Y., Zhang, X., Lin, L., Cao, J.: Learning to Warm Up Cold Item Embeddings for Cold-Start Recommendation with Meta Scaling and Shifting Networks. In: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval. p. 1167–1176. SIGIR '21, Association for Computing Machinery, New York, NY, USA (2021), https://doi.org/10.1145/3404835.3462843
87. Zhu, Z., Kim, J., Nguyen, T., Fenton, A., Caverlee, J.: Fairness among New Items in Cold Start Recommender Systems, p. 767–776. Association for Computing Machinery, New York, NY, USA (2021)

Eyad Kannout is currently pursuing a Ph.D. in the Institute of Informatics at the University of Warsaw, wherein he is also delivering the labs of Machine Learning course for many years. His research activities are mainly focused on Recommendation Systems and ML/AI. He obtained his Master in Computer Science at Warsaw University of Technology, Poland in 2016, and received his Bachelor in Informatics Engineering from University of Aleppo, Syria in 2012. In addition to his academic career, Eyad Kannout held several software engineering positions in the financial sector.

Michał Grodzki is student at faculty of Mathematics, Informatics and Mechanics at the Warsaw University. He received the Bachelor's degree in 2020 in the field of ordinary differential equations. Currently, he is pursuing his Master and mainly interested in Machine Learning and Recommender Systems research.

Marek Grzegorowski is for many years, associated with the Institute of Informatics at the University of Warsaw, wherein, in 2021, received Ph.D. in computer science. He is an

active and experienced researcher in the fields of science related to data exploration, machine learning, artificial intelligence, and recommender systems. He authored a number of scientific articles. In his career, conducted several research and development projects related to the application of ML/AI in academic and industry collaboration.

Received: April 11, 2022; Accepted: November 16, 2022.