Mitigating the Cold-Start Problem by Leveraging Category Level Associations

Sandeep Chowdary Annabathuni

A Thesis

in

The Department

of

Computer Science and Software Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of

Master of Science (Computer Science) at

Concordia University

Montréal, Québec, Canada

March 2023

© Sandeep Chowdary Annabathuni, 2023

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the thesis prepared

By: **Sandeep Chowdary Annabathuni**

Entitled: **Mitigating the Cold-Start Problem by Leveraging Category Level Associations**

and submitted in partial fulfillment of the requirements for the degree of

Master of Science (Computer Science)

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the Final Examining Committee:

______________________________ Chair
Dr. Joey Paquet

______________________________ Examiner
Dr. Joey Paquet

______________________________ Examiner
Dr. Sabine Bergler

______________________________ Supervisor
Dr. Yuhong Yan

Approved by ______________________________
Leila Kosseim, Graduate Program Director, Chair
Department of Computer Science and Software Engineering

March 22, 2023

____
Mourad Debbabi, Dean
Faculty of Engineering and Computer Science

Abstract

Mitigating the Cold-Start Problem by Leveraging Category Level Associations

Sandeep Chowdary Annabathuni

Recommender systems model user preferences by exploiting their profiles, historical transactions, and ratings of the items. The quality of the recommendations heavily relies on the availability of the data. While typical recommendation methods such as collaborative and content-based filtering can be effective in a wide range of online shopping and e-commerce applications, they suffer from the cold-start problem in settings where new users enter the system and ratings are sparse for new or low-volume items. To this end, we present a pairwise association rule-based recommendation algorithm that builds a model of collective user preferences by utilizing mined associations at both the item and the category levels. In the meantime, the model allows an individual user's in-session activities to be integrated at the category level to further improve the recommendation quality. Experimental results show that the proposed method improves recommendation performance, as compared to similar approaches.

Acknowledgments

First, I would like to express my gratitude to my supervisors, Dr. Yan and Dr. Wang. Through their guidance, I learned a great deal about how to think, work, and write. My sincere gratitude is extended to Dr. Wang for his support throughout my thesis research. He provided me with valuable guidance and knowledge. Thanks to his supervision and the support he provided me, I would like to express my gratitude and appreciation. Without his continuous support and assistance, I would not have been able to complete my thesis. I would also want to express my gratitude to MITACS and Concordia University for their funding.

In addition, I would like to thank my senior, Mr. Hou, for always being available to aid me in my studies and personal life. It is Ms. Sibila who stands by me, and she gives me spiritual support. My future life will be influenced by all of them. The final thing I would like to say is thanks to my parents for their support and encouragement to keep studying

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Recommender systems model user preferences by exploiting their profiles, historical transactions, and ratings of the items. Based on the type of data being collected and the ways of using them in recommendation algorithms, the approaches for recommendation can be classified into three categories. These categories are content-based (CB), collaborative filtering (CF), and hybrid [Koren et al., 2009]. CB filtering recommends items to users by matching the features and attributes of items to user profiles. In this approach, it is essential to learn user preferences over available items. A typical approach is to compare potential candidates with previously liked items by the user, and the best-matched items are recommended. On the other hand, CF does not require analyzing the features and attributes of content items. Instead, it relies on the rating feedback matrix, which represent the preference relationship between users and items. This matrix contains user ratings on specific items which can be mined by data mining algorithms to identify similar user clusters (neighbors of the given user for recommendation). A CF approach make recommendations by first predicting the missing ratings for the given user based on the ratings from their neighborhood and, then, recommend items with high ratings. To obtain a useful rating feedback matrix, a large amount of data needs to be collected from user behavior in the past to predict which items users will like. The hybrid approach is one that combines CB filtering approach and CF approach attempting to further boost the quality of the recommendations in the case that content information or rating information in the feedback matrix is not sufficient for a single CB or CF approach to reach the recommendation quality requirement.

In general, the quality of the recommendations relies heavily on the availability of the data. In particular, CB and CF based approaches are very sensitive to data sparsity. While CB and CF can be effective in a wide range of online shopping and e-commerce applications, they produce poor recommendations in settings where new users enter the system and ratings are sparse for new or low-volume items. In these settings, the data of content features and attributes and the ratings in the feedback matrix are sparse. When the sparsity of content information or item ratings reach to a level that significantly degrades the recommendation quality provided by the recommender systems, we say the recommender system faces a cold-start problem. While there is no universally agreed data sparsity threshold to identify the cold-start situation for recommender systems, for CF based approaches, it is suggested in Zhang et al. [2014] that the sparsity of ratings for cold-start items is higher than 85% and the sparsity of ratings for complete cold-start items is 100%.

Cold-start problems in recommender systems can be seen as an extreme form of data sparseness which can derive from the lack of user information (user-only), the lack of item information, or the lack of both item and user information. When a new user is added, recommendations may be inaccurate due to the lack of user preferences or historical purchase data. Further, existing users may also encounter this problem if their interests change, their preferences have changed based on their mood or context, or if they are less active. Currently, due to the rapid expansion of e-commerce platforms, a wide variety of items are being added to the system and the majority have not been purchased and have no historical transactions. Item-based cold starts may result from sparsity in item interactions. Additionally, it is possible that the value of an item may change over time, which may lead to a decline in interest from users. It may be detrimental to the business to recommend these items. In some cases, both the user and the item may be newly created and contain no information. It is also called complete cold-start. This presents the same challenges as item-based and user-based cold starts. In the cold-start setting, collaborative and content-based filtering recommender systems do not have sufficient information to effectively compute the neighborhoods of similar items and like-minded users, which leads to low-quality recommendations or an inability to produce recommendations.

1.1 The Cold-Start Problem

The cold-start problem addressed in this thesis is motivated by an online artwork marketplace project for which a recommender system needs to be developed. The platform has a large number of visits and interactions, most of which came from anonymous users (users who did not register). In those situations, it was not possible to collect explicit user data, such as ratings or user profiles. This made it difficult to recommend artwork, as we could not track users and their interactions over time. However, we are able to track in-session activies of unique visits while the users are interacting with the platform. For example, the sequence of artworks that a particular user interacts and the time they spent on each of them. We try to provide relevant recommendations by leveraging the limited user-item in-session interaction information in a real time manner.

The characteristics of the cold-start problem can be summarized as follows:

- **Users interact anonymously**. This situation is typical in situations where applications require high-level of privacy or user-registration is not enforced. For example, most e-commerce websites allow anonymous users to interact with the platform. In these cases, it is no possible to build stable user profile since the platform does not have or cannot identify the user's historic transaction data.
- **The rating feedback matrix is not available**. Since the user interacts with the platform anonymously, their rating feedbacks cannot be linked to the user and used in their future visits. This makes the use of CF based approaches infeasible.
- **Content information of items is available**. The content information of the collection of artworks on the platform is available. In addition, artworks are assigned category labels by the artists and the domain experts.
- **In-session activities can be tracked in real time**. The platform can track a user's in-session activities when they are interacting with the platform. These in-session activities can involve

page views, time spent, shopping cart, and other viewing behaviors.

1.2 Association Rule-Based Recommendation

There are a variety of studies on mitigating the cold-start problem in recommender systems [Osadchiy et al., 2019, Viktoratos et al., 2018, Wei et al., 2017]. These studies have different assumptions and target different problem settings. Most of them require certain level of user-item history data, which are not suitable for our cold-start problem setting in which no history data is available for constructing user profiles. Association rule-based methods have been considered suitable in a number of application contexts where the construction of user preference models is not possible [Osadchiy et al., 2019], e.g., in some online shops or services where users interact anonymously. An important advantage of using association rule-based techniques is that they are independent of personal user models and do not require a complex system of ratings. They rely on a model consisting of a set of mined associations that represent the collective preferences of users. The recommendation for individual users is produced based on a subset of association rules fired by the user's in-session activities, such as a set of observed items selected or browsed by the user. This rather unique mechanism used by association rule-based recommendation methods makes them an interesting type of approach to the cold-start problem.

However, in applications involving a large number and variety of items, which is a typical case in many online shopping platforms, rule model building can be very expensive in terms of computation needed due to the requirement of mining association rules that match all possible combinations of the observed items. That is because it requires an exhaustive search of the item set to generate the necessary rules, and the number of possible combinations increases exponentially with the number of items. Therefore, the computational cost of rule model building increases significantly as the number of items increases. In addition, as a frequent issue in many applications, data sparseness could lead to low support and low confidence levels for many mined association rules, which prevent them from being included in the model. It is possible that a recommender system that uses association rules to produce recommendations may not be able to produce recommendations when given a combination of observed items that is less common in comparison to the most common combinations.

In a recent study [Osadchiy et al., 2019], a recommender system based on pairwise association rules is proposed. According to the study, pairwise association rule mining can be more efficient since it does not require the mining of all possible association rules, but only the frequencies of the item pairs. The performance of the proposed system is compared with three existing methods that are adapted to meet the constraints of the application task: recommending omitted foods in a real-world dietary recall system. The evaluation, performed on a large data set of real dietary recalls, has demonstrated that the pairwise association rule-based algorithm performs better for the defined task.

1.3 Contribution

In this thesis, we improve the performance of the pairwise association rule-based method described in Osadchiy et al. [2019] by proposing a Pairwise Multilevel Association Rule-based Recommendation (PMARR) algorithm that utilizes mined pairwise associations at both the item level and the category level. The PMARR algorithm is particularly suitable for data sets in which clear hierarchical relationships exist, i.e., items can be assigned to exclusive categories. In the proposed approach, associations at both item and category levels are mined from the data set. Data sparseness at item level can be mitigated by higher-level association rules since they usually have much higher support. In the situations where no association rules are available at item level given observed items from a user, by leveraging category level associations, the proposed algorithm can still recommend relevant items to the user. Furthermore, the fact that the item level associations are already implicit in the category level associations reduces the non-transitive association problem Kim et al. [2004] commonly observed in association rule-based recommender systems when facing sparse data sets.

1.4 Thesis Outline

In the following chapters, we will provide a background for our research. A description of the research and design problems with clear definitions and a limited scope was provided. The remainder of the thesis is organized as follows:

- In Chapter 2, we provide a summary of the literature on existing approaches to the cold-start recommendation problem.
- In Chapter 3, we present the proposed PMARR model building algorithm.
- In Chapter 4, we present the proposed algorithm for recommendations using the trained PMARR model.
- In Chapter 5, we describe the methodology we use to compare the performance of PMARR with that of existing related recommendation algorithms and presents the experimental results.
- In Chapter 6, we conclude the thesis and discuss future works

Chapter 2

Background and Related Work

A recommender system is a subclass of information retrieval systems which provides users with item suggestions they might be interested. The goal of developing recommender systems is to reduce information overload by retrieving the most relevant information and services from a large amount of data, thereby providing personalized service. At Xerox Palo in 1992, Tapestry [Goldberg et al., 1992] is the first recommender system developed to deal with the large volume of incoming documents via electronic mail. Tapestry enables users to filter emails according to topics that they select as relevant. Although the topic categories were not necessarily created by the program, they could be proposed by the users based on their own interpretation and classification of the contents of the emails. Other users can use those topic categories to specify their preferences. Besides filtering emails based on their content, Tapestry also determines the importance of a message by determining how ”popular” it was among other employees in the company who also received the message. As a result of these two principles, the recommendation of items based on their content and the popularity of those items among other people with similar interests, modern content-based and collaborative filtering recommender systems were created.

2.1 Summary of Recommender Systems

2.1.1 Collaborative Filtering

Collaborative filtering (CF), also known as social-based filtering, is likely the most widely used recommender system. It is primarily based on the assumption that people who agreed in their evaluations of particular items will likely agree again in the future [Resnick et al., 1994]. As an example, consider two users named X and Y who share similar preferences. The underlying algorithm will be able to identify the similarity between two ratings if they are very similar. An interpretation of this similarity can be drawn about values that have been incompletely defined. As a successful use case, Amazon now applies collaborative filtering, which provides real-time high-quality recommendations [1]. A Collaborative filtering system can be categorized into two types: user-based and item-based [Sarwar et al., 2001].

User-Based Collaborative Filtering

User-based method, also known as memory-based, predicts which items a particular individual will be interested in based upon the ratings provided by other individuals who share similar interests [Bellogín et al., 2014, Goldberg et al., 1992]. Among the most well-known user-based CF algorithms are nearest neighbor algorithms. By utilizing statistical similarity measures to compute user-item matrices, these algorithms discover similar users. Based on the user's behavior or preferences, this method attempts to find users who are similar to the active user. They serve as neighbors of the active user and can be used to generate predictions.

Item-Based Collaborative Filtering

Using the item-based method, also called model-based, predicts evaluations without taking account of similarities between users, but rather by identifying similarities between the items and other items associated with the user [Sarwar et al., 2001]. In order to calculate the similarity, we take note of how many users bought item X along with item Y. It can be assumed that there is a similarity between the two items, if the correlation is high enough, and that they are similar to one another.

[1] https://fortune.com/2012/07/30/amazons-recommendation-secret/

The subsequent purchase of item Y will then be recommended to users who purchased item X, and vice versa.

2.1.2 Content-Based Recommender Systems

Content-based (CBF) method works on the principle that if a user has purchased an item that has certain characteristics, chances are he or she will be interested in an item with the same characteristics [Chicaiza and Valdiviezo-Diaz, 2021]. The two main pieces of information necessary for content-based filtering are: the characteristics of items and user profiles. These recommendations do not necessarily require the presence of a large group of users or a detailed rating history.

2.1.3 Knowledge-Based Recommender Systems

Knowledge-based recommendations (KB) recommend items to users based on the item's domain knowledge and how it is appropriate for the user's preference [Colombo-Mendoza et al., 2015]. In order for a recommender system to be knowledge-based, it must make recommendations based not on a user's rating history, but on specific queries the user has submitted [Shokeen, 2018]. This can take two forms [Aggarwal et al., 2016]: Constraint-based recommender system and Case-based recommender system.

Constraint-based recommender system

In this scenario, users specify certain features or domain rules and items qualifying those features are selected for recommendation based on those specifications [Aggarwal et al., 2016].

Case-based recommender system

A case-based recommender utilizes similarity metrics to select items that are similar to the target item (or case). In order to retrieve similar items to the cases described here, similarity metrics are defined on the item attributes. Contrary to constraint-based systems, these attributes are not subject to strict constraints.

2.1.4 Demographics-Based Recommender Systems

A Demographics recommendation system generates recommendations based on the demographic characteristics of the user. According to the user's attributes, it categorises them and recommends items using their demographic information [Pazzani, 1999].

2.1.5 Context-Aware Recommender Systems

Context-aware recommendation system recommends the most similar and relevant items based on the user's profile, interests, preferences, and interactions. By incorporating contextual features into the model, prediction accuracy is improved [Sejwal et al., 2020]. Furthermore, context is highly dynamic and never explicitly specified by users [Woerndl and Schlichter, 2007]. Application domains also play a considerable role in the context attributes used.

2.1.6 Hybrid Recommender Systems

Hybrid recommender system employs two or more recommendation strategies and leverages their complementary strengths in different ways. An example of this is the application of a hybridization of demographics and collaborative filtering methods to deal with the cold start problem [Wang et al., 2012]. Currently, there are four major recommendation techniques used to construct hybrids: collaborative filtering, content-based recommendation, demographics, and knowledge-based recommendation. In contrast to the first three, which use learning algorithms, Knowledge-based relies upon domain knowledge and makes inferences about users' needs and preferences. Through the use of multiple techniques, hybrid recommendation systems are able to produce outputs that outperform standard single component systems.

2.2 Existing approaches to the cold-start problem

There are a variety of studies on mitigating the cold start problem in recommender system design. The existing approaches have different assumptions, target different problem settings and application domains. Most of them integrate filtering with machine learning, data mining methods. Commonly used categorization scheme such as collaborative filtering (CF), content-based (CB) filtering, and hybrid recommender systems that combine the first two does not reflect well in terms of how the approach can help mitigate the cold start problem. In this thesis, we adopt a different categorization scheme. We summarize the representative and state-of-the-art approaches to the cold start problem in recommender systems based on the types of auxiliary or side information that the approach provides to enrich the modeling of user preferences given sparse ratings, such that the recommendation quality can be improved.

2.2.1 Improved CF-Based Methods

Conventional CF-based methods use the ratings given to items by users as the sole source of information for learning to make recommendation. When ratings are sparse, auxiliary information such as item content data may be utilized to improve recommendation quality. A range of machine learning based methods have been proposed to retrieve content information which is then used by CF in recommender systems. In [Wang et al., 2015], a hierarchical Bayesian model called collaborative deep learning (CDL) is adopted to supplement item content information with CF ratings data. The proposed model jointly performs deep representation learning for the content information and collaborative filtering for the ratings matrix. In [Wei et al., 2017] the proposed CF algorithm is tightly coupled with a deep learning neural network. The deep learning neural network is used to extract the content features of the items. The CF model which is a modified TimeSVD++ [Koren, 2009] is used to take the content features into prediction of ratings for cold start items. In [Ocepek et al., 2015] local learning algorithms combined with attribute selection, and value aggregation algorithms are proposed to impute missing ratings into the input matrix to improve the performance of CF. In [Zhang et al., 2021] the authors focus on the challenge of the item cold start problem in the setting where the structure of the user-item interaction ratings exhibits high level of non-linearity. They design a CF recommendation approach that adopts neural networks to predict user-item ratings. To mitigate the item cold-start problem, the proposed approach leverages item embeddings to capture the content feature and employs it as auxiliary information.

2.2.2 User Profiling

User profile data can be used as the auxiliary information for the recommender engine to improve recommendation quality when ratings are not sufficient for CF algorithms. User profile information can be obtained from various channels. For example, [Zhou et al., 2011] try to elicit user profiles through an online interview process when new users first join the system. The process progressively querying user responses through a decision tree with each node being an interview question, enabling the recommender to query a user adaptively according to their prior responses. Another source of auxiliary information for building user profiles can be their social media streams [Herce-Zelaya et al., 2020] propose to generate a behavioral profile for each user by extracting implicit information from users' social media data. The users are then be classified based on their profiles and predictions are made using machine learning techniques such as classification trees and random forests. Compared with the approaches that obtaining user profiles through interviews, this social media data based profile building does not require active involvement of users. In the same spirit, a recommender system based on knowledge and social networks among users is presented in [Carrer-Neto et al., 2012]. New user profiling can also be improved using ontology-based approaches. In [Lika et al., 2014], profiles of new users to e-learning recommender systems are improved by incorporating additional learner data. A domain ontology is developed to model the learner and learning objects with their characteristics. Given the consistency provided by using ontology-based modeling of user and learning objects characteristics, collaborative and content-based filtering techniques are used in the recommendation model to generate the top N recommendations. In addition to profiling individual users, classifying new users to existing well profiled clusters has also been proven effective. In [Lika et al., 2014], a Naive Bayes approach is adopted to allocate new users to existing clusters using their demographic data. Predicted rating of the newcomer user for each existing item is computed using the weighted sum of the ratings given by the user's set of neighbors on the underlying items. In [Nahta et al., 2021], user demographic data is also used as input to the proposed recommender system. However, different from the approach in [Lika et al., 2014], users are not clustered based on demographic data in the proposed approach. Instead, user demographic data and item genre are considered as metadata features. These features along with the rating matrix are taken as inputs to a rather complex recommendation framework which consists of several modules including a neural network for rating prediction.

2.2.3 Session Context Information

Session context information such as session properties, e.g., session length, and session behaviors, e.g., users' clicks on items, can also be leveraged to improve recommendation quality in cold start situations. The recommender systems that learn new users' preferences from the sessions associated and generated during their consumption process based on session contexts are called Session-Based Recommender Systems (SBRSs). SBRSs are also useful in mitigating the cold start problem since they can make recommendations without involving rating matrix or user profiles. Given a current session context, a KNN-based SBRS computes the K interactions or sessions that are most similar to the current session and recommend them to the current item as the next choice. Candidate similarity score for an interaction is also calculated to indicate its relevance to the current interaction as the guidance of recommendations. Certain similarity measure like cosine similarity can be used for the calculation ([Ludewig and Jannach, 2018]). As indicated in [Ludewig and Jannach, 2018], KNN-based approaches, e.g., session-KNN, have achieved superior recommendation accuracies even compared with some deep neural network-based approaches on some e-commerce datasets. For SBRSs, deep neural network-based approaches are usually believed to be superior on imbalanced or sparse datasets, or complex datasets with long-term or high-order dependencies ([Zhang et al., 2019]). In [Yu et al., 2016], authors propose a hierarchical deep learning architecture that makes dynamic recommendations based on users' sequential multi-session inter- actions with items. The proposed architecture consists of a high-level model which uses Recurrent Neural Network (RNN) to model users' evolving long-term interests across different sessions and low-level model which uses Temporal Convolutional Networks (TCN) to model the short-term interactions. In this way, both the long-term interests and the short-term interactions are utilized to predict the next interaction. In [Liu et al., 2018], authors propose a dynamic recommender model based on RNN. The model learns a dynamic representation of a user and sequential features among baskets. A user's dynamic interests at different time are captured by the dynamic representation, while interactions of all baskets of in multiple sessions are captured through the learned sequential features.

In [Liu et al., 2018], the authors propose a short-term attention/memory priority model that learns a uniform embedding space with items across sessions and a neural attention model for next-click prediction, such that both users' general interests from the long-term memory of a session context and users' current interests from the short-term memory of the last-clicks are taken into account in session-based recommender systems. Latent representation based SBRSs, on the other hand, capitalize the benefits of their effective representation. It is reported in [Liu et al., 2018] that latent representation-based approaches, although do not involve complex network structures, perform well compared with some deep neural network-based approaches. Other approaches that model the evolving user interests are also proposed in the context of CF. For example, TimeSVD++ ([Koren, 2009]) is a model that simulates the temporal dynamics of user interests by changing static biases and latent factors into time-dependent ones. A modified version of TimeSVD++ is used in [Wei et al., 2017] to capture the temporal dynamics of user preferences and item features. In [Zhang et al., 2014], a latent transition matrix is used to summarize the evolving preferences for each user.

2.3 Association Rule-Based Approaches to Cold-Start Problems

The use of recommendation rules that constructed based on associations among items is also a common practice among researchers to decrease the effect of the cold start. Rules are IF-THEN statements that are used to represent complex relationships among the data. They can be mined automatically from members' history or constructed manually by the developers. In most of the cases, associations rules are integrated with CF, CB based methods to mitigating the cold-start problems by 1) leveraging ensemble-based recommendation; 2) expanding user profiles, content information, and ratings; and 3) reducing the impact of transaction sparsity.

2.3.1 Ensemble association rule-based recommendation

Association rules can be used in combination with other recommendation algorithms to form ensemble-based recommendation approaches. In [Xiao et al., 2018], the authors design a recommender system for online courses recommendation by combining association rules, content filtering, and collaborative filtering approaches to accurately model user interests and their dynamic changes.

In [Viktoratos et al., 2018], the authors focus on context-aware recommendation by combining association rule mining with community created knowledge and ontologies. A scoring function based on probability metrics is also described. Using Association Rules Mining, [Chellatamilan and Suresh, 2011] construct a proposition system for the e-Learning structure to provide researchers with the most appropriate learning materials and e-learning resources. To aggregate data from students, this system utilized an audit review. During the survey, students are asked about their educational background, their experience with technology, and their demographic information. In addition, the system analyzes the Moodle Learning Management System (LMS). In order to generate recommendations, data mining techniques such as association rules and similarity metrics, such as cosine similarity and Jaccard similarity, are used. A major disadvantage of the system is that it requires students to fill out a survey questionnaire in order to gather information about them- selves and their backgrounds. [Tsuji et al., 2014] proposed an approach in which association rules and document similarity metrics are combined with a support vector machine. This approach is demonstrated in a system for recommending books. Instead of relying on a single source of information, such as confidence by association rules, the system recommends books based on multiple sources of information combined with optimal parameters.

Association rules can also be supported by decision rules manually constructed by the developers. In a tourism context, [Viktoratos et al., 2015] propose a methodology which exploits Point-Of-Interest (POI) owners' rules (user defined rules that represent their offering/marketing policy) to recommend contextualized offers to regular users, related with nearby POIs. In a follow up work, [Viktoratos et al., 2017] combine user defined rules (representing regular users' preferences/ daily patterns), POI owners' rules and social connections to recommend nearby POIs along with contextualized offers to regular users.

2.3.2 User Profile and Item Rating Expansion

Association rules can be used to boost the performance of recommender systems in the cold-start settings by enhancing the user profile, content information, and ratings data. In [Shaw et al., 2010], the authors used non-redundant association rule sets to improve recommendation results by expanding user profiles. A comparison was also made between redundant and non-redundant rules in terms of their performance. Using a user profile that includes book borrowings and book meta data, Jomsri [2018] proposed a system for recommending books to students at a university library. By examining association rules between faculty, students, book categories, and titles, a book recommender model was developed. Historical data on user loaning played an important role in developing the recommendation system. [Wakita et al., 2015]used association rules to select relevant brands for users that are similar to their favorite brands. Association rules are used to enhance the model of users' favorite brands. According to the rules, there are frequent occurrences of fashion-brand liking, while the fashion-brand feature allows for a comparison of brands on the basis of their similarity. Combined with these two methods, the proposal presents a novel hybrid method. However, defining users' tastes in terms of several well-known factors does not guarantee successful suggestions because apparel products contain emotional and subjective elements that are difficult to formalize into structured data.

User ratings data can also be enriched using association rules. [Bendakir and Narimel, 2006]proposed a course recommendation system based on the mining of association rules. In order to make recommendations, the system incorporates a data mining process and user ratings. A major characteristic of the system is that it is divided into two phases: an off-line phase involving data mining, and an on-line phase involving student's evaluation on the recommendations. In the off-line phase, association rules are extracted from the data using the Apriori algorithm, and in the on-line phase a student evaluation is conducted to determine whether a new rule should be added or an existing rule should be removed based on certain criteria including course evaluation, number of evaluation, user fidelity and course acceptance threshold. As more student evaluations of the previous recommendations are provided, the system can be enhanced, and the rules are updated as more student evaluations of the previous recommendations are provided. [Leung et al., 2007] implements a hybrid recommendation algorithm which makes use of Cross-Level Association Rules (CLARE) to integrate content information about domain items into collaborative filters. Whenever rating data cannot be used to generate recommendations for an item, CLARE uses the attributes of the item to generate cold-start recommendations. The user provided ratings as well as the item features are passed as inputs to the model building process. It utilizes adaptive-support and fuzzy association rule mining techniques to generate Item- Item associations as well as user-item associations. Similarly, in [Sobhanam and Mariappan, 2013], the association rules are used to add new items with ratings by finding similar items. The process is divided into two phases. The first phase involves pre-processing of the sparse dataset with a software tool called Weka, which then stores the data in Microsoft Access. During pre-processing, missing values in sparse datasets are filled in. In the next step, these new items are clustered using the k-means algorithm.

2.3.3 Transaction Sparsity Mitigation

Association rules can be combined with CF, CB and other recommendation algorithms as a ensemble rule-based approach to cold-start problems. They can also be used to boost the performance of the conventional recommender systems in cold-start settings by enhancing user profiles and user-item ratings. However, to build a high-performance rule model, a large volume of transactions is needed to ensure the coverage and strength of the mined associations. In the settings where transaction data is not sufficient, some techniques can be used.

Multilevel and cross-level associations are also exploited when data that represents associations at item-to-item level is sparse [Leung et al., 2007, Kim et al., 2004]. The idea is to exploit category level associations to which the support is usually stronger. In [Shaw et al., 2010], rather than finding association rules between items, the paper discovers association rules based on topics. In other words, a topic can be viewed as a category. Each user represents a transaction, containing the set of topics they previously viewed, rated, or liked. Therefore, association rules are applied to these transactions to understand how the topics are related to make appropriate recommendations. In [Najafabadi et al., 2017], a clustering technique is employed to ensure the effectiveness of the mined association rules in data sparsity condition. In addition, clustering also reduces the size of data and dimensionality of the item space before performing association rules mining. Another unique feature of the proposed association rule mining is that it captures multiple purchases per transaction in association rules, rather than just counting total purchases made. In the same spirit, [Aher and Lobo, 2012] used a classification algorithm, namely ADTree, to classify instances in a course recommendation application. Based on the ADTree model, an instance is classified by traversing all paths for which all decision nodes are true and summarizing any prediction nodes that are traversed. Recommendations are generated using the Apriori algorithm. Results are compared with those of the Apriori Association Rule algorithm alone. In [Fauzan et al., 2020], K-mode clustering and Apriori association rules are used to tackle transaction sparsity. In order to determine the number of clusters in the dataset, K-mode clustering is used, and then the Ariori algorithm is used in order to generate rules for each cluster. According to the results of testing and evaluation, the Apriori Association Rules method is effective for course recommendation.

2.3.4 Summary and Discussion

In this subsection, I first summarize the reviewed association rule-based approaches to the cold-start problems from the perspective of application areas and data properties. I then discuss the research gap and the position of my proposed approach in the big picture of the literature.

Table 2.3.4 shows a list of reviewed association-rule based recommender systems papers. The approaches from those papers are proposed for the purpose of mitigating the cold-start problems using association rules. Since these papers are already reviewed from the methodology perspective in the previous text, the list is organized based on the application areas and the data properties.

Table 2.1: An overview of the association rule-based recommendation approaches to cold start problems by application areas and data properties.

Field	Data property	Technique	Reference
e-Learning	Data was taken from the Computer Science Department at the University of Montreal. Students' past course choices are analyzed	AR	Bendakir and Narimel [2006]

e-Learning	This study utilizes two datasets, namely the Canvas network person course and the HarvardX-MITX dataset. In the Canvas network dataset, there are more than 325,000 records, each representing one person's activities in one of the 238 courses offered. Data from MITx and HarvardX courses on the EdX platform are included in the HarvardX-MITX dataset for the 2013 academic year (Fall 2012, Spring 2013, and Summer 2013)	AR, K-Mode	Fauzan et al. [2020]
e-Learning	A sample dataset extracted from the Moodle database of a college following the collection of student enrollment data	AR, AD Tree	Aher and Lobo [2012]
e-Learning	A log of students' activity on the Learning Management System (LMS) Moodle, as well as survey data regarding students' academic performance, interactions, and personal characteristics	AR	Chellatamilan and Suresh [2011]
Book	Analysis of user profiles, including borrowing history, book categories, etc	AR	Jomsri [2018]

Book	BookCrossing dataset consists of a list of users along with a list of books as well as the ratings assigned to those books by the users.	CF, AR	Shaw et al. [2010]
Document	Records of library loans and bibliographic data of T University Library	SVM, AR	Tsuji et al. [2014]
Fashion	There are two datasets, one containing brand features and the other containing brand association rules. Our Web crawling program collects the first brand-feature dataset from a social networking site called ZOZOPEOPLE. Our Web crawling program also collects the brand association rules from this site.	Hybrid, AR	Wakita et al. [2015]
Movies	An analysis of the MovieLens dataset from GroupLens was conducted	K-Means, AR	Sobhanam and Mariappan [2013]
Movies	An analysis of the MovieLens dataset from GroupLens was conducted	CF, AR	Leung et al. [2007]
Others	Data was scraped and stored from an e-commerce site	AR	Huang and Duan [2012]

In the following we discuss the need for a new recommendation approach for addressing the cold-start problem studied in this thesis. In this chapter, we have provided a comprehensive review of the existing approaches to the cold-start problems. We first classified the approaches in the literature based on the types of auxiliary or side information that the approach provides to enrich the modeling of user preferences given sparse ratings. These include improved CF-based methods, user profiling, session context information, and association rule-based approaches. All approaches in the first three categories (improved CF-based methods, user profiling, session context information) requires some level of user specific history. As discussed in Chapter 1, this type of data cannot be supplied in the cold-start setting addressed in this thesis. This restriction also applies to the ensemble association rule-based recommendation approaches and association rule-based user profile and item rating expansion.

Some association rule-based recommender systems that designed to mitigate the transaction sparsity issues may not require user specific history. However, they usually require complete rule enumeration which is often intractable for datasets with a very large number of multi-valued attributes. In a recent work [Osadchiy et al., 2019], a pairwise association rule-based recommender algorithm is proposed. The algorithm relies only on item level pairwise associations. It is stated in the study that pairwise association rule mining can be more efficient since it avoids the need to mine all possible association rules, but the item pairs frequencies. It also helps to reduce the cases of low support and low confidence levels for many mined association rules, which prevents them from being included in the model. However, this pairwise association rule-based recommender algorithm recommends items based on the collective preferences of the user. They are unable to introduce any level of personalization. In addition, no category level association rules are included in the model to improve its robustness against data sparsity. We extend the pairwise association rule-based recommender algorithm by integrating a category level association rule model and in-session category level preference information into the recommendation process to provide personalized recommendations in the data sparse settings without requiring user history data.

Chapter 3

Pairwise multi level association rules

The construction of rule models can be extremely expensive when there is a large number and variety of items involved in the application. This is a typical case for online shopping platforms. Due to the necessity of mining association rules that match all possible combinations of observed items, rule model building can be extremely computationally intensive. Furthermore, as a common problem in many applications, data sparsity may have a negative impact on the support and confidence levels of many mined association rules. The sparsity problem is amplified when large datasets are considered, as there may not be enough instances of a given combination of items for the support and confidence values for the association rules to reach a sufficient level. This can lead to a lack of coverage, where the model cannot generate recommendations due to the absence of rules related to the requested items. This may prevent them from being incorporated into models. Without a sufficient number of rules included, an association rule-based recommender system may not be able to produce recommendations when given a less common combination of observed items. In a recent study ([Osadchiy et al., 2019]), a recommender system based on pairwise association rules is proposed. It is stated in the study that pairwise association rule mining can be more efficient since it avoids the need to mine all possible association rules, but the item pairs frequencies.

In this chapter, we further improve the performance of the pairwise association rule-based method described in [Osadchiy et al., 2019] by proposing a Pairwise Multilevel Association Rule-based Recommendation (PMARR) algorithm that utilizes mined pairwise associations at both the item level and the category level. We begin by presenting a pairwise multilevel association hierarchy, which outlines the process of building the pairwise multilevel association rule model, namely the generation of category-level transaction data sets and the mining of pairwise association rules. As a next step, we will discuss the process of creating the multilevel association rule model, also called model training, which is the process of mining association rules at every level, beginning at the item level and working up to the target level.

3.1 Association Rules

A market basket analysis is a data mining technique used by retailers to identify purchasing patterns or associations between items. Essentially, it seeks to determine which combinations of items most frequently appear together in transactions. The problem can be understood as follows: Let $I = \{i_1, i_2, ..., i_n\}$ be the large set of items. Describe a transaction $t \subseteq I$ as a set of items with which an individual user purchases during a single visit. The transaction dataset $\mathcal{T} = \{t\}$ consists of all transactions recorded within a specified period of time [Raeder and Chawla, 2011]. By identifying associations between different items, this process identifies user preferences. To uncover strong rules, market basket analysis uses association rules to analyze purchase history or transaction datasets. An association rule mining (ARM) method analyzes items based on their frequency of appearing together in the transaction database to identify relationships between them [Garcia-Molina, 2008, Piatetsky-Shapiro, 1991]. The rule is defined as $X \rightarrow Y$, (If X, then Y) where $X, Y \subseteq I$. The itemsets X and Y are called antecedent and consequent respectively.

Definition 3.1.1 *The support of a rule is measured in terms of its usefulness, which describes the proportion of transactions which include both items X and Y*

$$sup(X \rightarrow Y) = P(X \cap Y) = \frac{|\{t \in T : X \subseteq t \quad and \quad Y \subseteq t\}|}{|T|} = \frac{\textit{Transactions containing X and Y}}{\textit{Number of total transactions}} \tag{1}$$

Definition 3.1.2 *The confidence of a rule's validity, which is measured in terms of the proportion of transactions which include item Y among those which contain item X*

$$conf(X \to Y) = P(Y|X) = \frac{|\{t \in T : X \subseteq t \quad and \quad Y \subseteq t\}|}{|\{t \in T : X \subseteq t\}|} = \frac{\textit{Transactions containing X and Y}}{\textit{Transactions containing X}} \tag{2}$$

Given a parameter such as minimum support s and minimum confidence c, R is the set of rules $X \to Y$ such that:

$$sup(X \to Y) \geq s \tag{3}$$

$$conf(X \to Y) \geq c \tag{4}$$

The process of generating association rules typically involves two steps. Initial considerations are made by determining which itemsets are frequently found in the dataset and satisfy the minimum support constraint. Following this, a second step will be performed in which all possible rules will be generated from each frequent itemset. Rules that do not satisfy the minimum confidence constraint will be removed. The tuning parameters for quality and count of rules are typically minimum support and minimum confidence. As a result, they play an important role in the generation of rules [Hahsler, 2017]. With the association rules defined, we will examine how they can be used to generate recommendations. There are several recommendation systems that apply association rules mining techniques to a variety of applications in the real world, such as e-Learning systems, e-Commerce systems, and course recommendation systems.

3.2 Pairwise Association Rules

Itemsets can range in size from one item to the total number of items in the collection. An approach that uses brute force to find frequent itemsets would be to create all possible itemsets and check the support value for each one. Furthermore, increasing the size of candidate itemsets will result in a reduction in the number of transactions supported by the itemsets. It is important to note, however, that this approach requires a significant amount of memory and time. Using the anti-monotone property ensures that every subset of a frequent item set is also frequent in order for rule generation to be efficient. According to [DuMouchel and Pregibon, 2001], pairwise associations are more efficient methods of discovering ”interesting” associations because these associations can be discovered based on the finding of pairs of items that appear frequently together, and then using this knowledge to analyze larger items sets containing these pairs. As an example, if ABC appears in a dataset with a certain frequency, then AB, BC, and AC are at least as frequent as the triplets.

PAR is a recent pairwise association rule-based recommendation algorithm ([Osadchiy et al., 2019]) proposed for drawing the collective preferences of groups of end-users. The algorithm relies only on item level pairwise associations. As opposed to collaborative and content-based filtering, the algorithm does not require a long history of users' preferences or a large number of item descriptors. Rather, the algorithm builds a collective model of preferences from the transactions of respondents in a given population. This is done by analyzing the relationships between items purchased by different users. Then, it use that information to predict what items a customer might like based on the items that other customers with similar purchases have chosen.

3.3 Pairwise Multilevel Association Hierarchy

Here we present the design of PMARR which produces user-specific recommendations using a pairwise multilevel association rule model. The multilevel association rule model is built based on a category hierarchy which organizes all available items and their categories into a hierarchical structure. As an example, shown in Figure 3.1, Level 0 refers to the item level in the hierarchy which contains all available items to the users, that is the item set $\{r, p, s, q, l\}$ in the example. Levels 1 and 2 are category levels in the hierarchy. We assume that an entity (an item or a category) in the hierarchy can only belong to one entity at its immediate upper level. The structure of the hierarchy does not limit the number of levels it can have. However, as discussed later in this section, including more category levels does not necessarily translate to better recommendation quality. In the defined hierarchy, C_k is the category set that contains all categories at level k. $c_k^{(i)}$ is the category at level k that contains $k-1$ level entity i. For example, in Figure 3.1, $c_1^{(r)}$ is a Level 1 category that contains item r. In case both entities i and j belong to the same upper-level category, we say $c_k^{(i)} = c_k^{(j)}$.

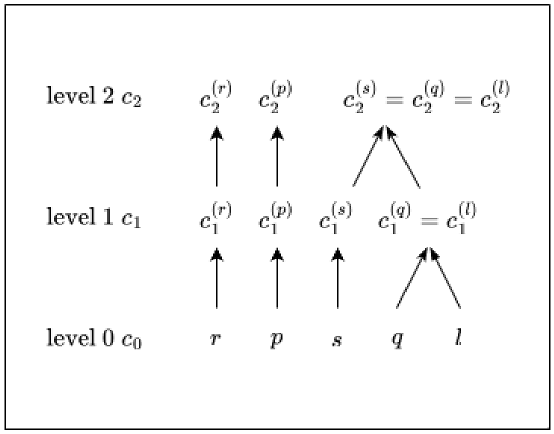


Figure 3.1: Example of category hierarchy

There are two steps involved in building the pairwise multilevel association rule model, namely category-level transaction data set generation and pairwise association rule mining. The first step converts the item-level transaction data set $\mathcal{T}_0$ to upper-level transaction data sets. For example, in Figure 1, a transaction $t_0 \in \mathcal{T}_0$, $t_0 = \{p, s, q\}$ is converted to Level 1 transaction $t_1 = \{c_1^{(p)}, c_1^{(s)}, c_1^{(q)}\}$ and Level 2 transaction $t_2 = \{c_2^{(p)}, c_2^{(s)}\}$, $t_1 \in \mathcal{T}_1$ and $t_2 \in \mathcal{T}_2$. The general process of generating category transaction data sets is described in Algorithm 1. Before running Algorithm 1, a meaningful and domain-relevant hierarchy should be established first based on domain experts' opinions. Algorithm 1 uses the item level transaction data set $\mathcal{T}_0$ to generate the transaction sets for higher category levels up to a target level L^{tar}. The value of L^{tar} is determined by the recommender system designer based on the data sparsity of the actual data set and the balance that the designer wants to strike between the system's ability to recommend and its recommendation quality (more explanation will follow in the next paragraph). The algorithm first define the transaction sets from $\mathcal{T}_1$ to $\mathcal{T}_{L^{tar}}$. It then works its way up from the item level transaction set to generate transaction sets at all levels until L^{tar}.

The second step of building the multilevel association rule model is called model training, which mines association rules for each of the levels from the item level to L^{tar}. The association rules mined are pairwise rules between two entities at the same level in the hierarchy. The confidence of a pairwise rule $c_k^{(i)} \rightarrow c_k^{(j)}$ at Level k, denoted as $conf(c_k^{(i)} \rightarrow c_k^{(j)})$, is defined in Equation 3.1, which is the conditional probability that $c_k^{(j)}$ appears in a transaction in $\mathcal{T}_k$ given that $c_k^{(i)}$ appears.

$$conf(c_k^{(i)} \rightarrow c_k^{(j)}) = \frac{P(c_k^{(i)}, c_k^{(j)})}{P(c_k^{(i)})} \tag{5}$$

The average support of pairwise association rules will increase at a higher level in the hierarchy. However, the relevance of items recommended using higher-level association rules to observed user selections will decrease since higher-level association rules will bring in a category of items to be recommended. Therefore, the system designer needs to decide how high in the hierarchy the transaction sets should be generated and mined, that is to determine the value of L^{tar}. Given the extent of the data sparseness at the item level and the actual hierarchical structure of the data entities, the value of L^{tar} can be determined empirically. For the restaurant data used in our experiments, L^{tar} is set to be 1, that is only one category level is involved since the data set only supports one level of meaningful category.

Algorithm 1 Category-transaction-data-sets $(\mathcal{T}_0, L^{tar})$: Generating category transaction data sets up to the target level L^{tar}

Require: $\mathcal{T}_0$, L^{tar}, item level transaction set and target category level

$\mathcal{T}_1, \mathcal{T}_1, \ldots \mathcal{T}_{L^{tar}} \leftarrow \emptyset$
for each $0 \leq k < L^{tar}$ **do**
  for each $t_k \in \mathcal{T}_k$ **do**
    $t_{k+1} \leftarrow \{categoryOf(i) | \forall i \in t_k\}$
    $\mathcal{T}_{k+1} \leftarrow \mathcal{T}_{k+1} \cup t_{k+1}$
  end for
end for
return $\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_{L^{tar}}$

3.4 Training of PMAAR Based Model

Detailed procedure of pairwise association rule mining at a single Level k is described in Algorithm 2. The algorithm mines all pairwise association rules from Level k transaction set $\mathcal{T}_k$ and computes their respective confidence scores. The number of occurrences of a category $c_k^{(i)}$ at Level k is counted in $OD[c_k^{(i)}]$. Each time a pair of entities $\{c_k^{(i)}, c_k^{(j)}\}$ is observed in a transaction in $\mathcal{T}_k$, the number of co-occurring are updated in $CD[c_k^{(i)}, c_k^{(j)}]$. The algorithm also calculates the confidence score $CF[c_k^{(i)}, c_k^{(j)}]$, which describes how likely it is that $c_k^{(i)}$ and $c_k^{(j)}$ will appear in a transaction as a result of $c_k^{(i)}$ appearing in a transaction. Algorithm 2 will be used $L^{tar}+1$ times to compute pairwise association rule sets from item level to L^{tar}. Collectively, these association rule sets form the pairwise multilevel association rule model based on which PMARR makes recommendations.

Algorithm 2 Train($\mathcal{T}_k$): Training the model based on categorical level association rules

Require: $\mathcal{T}_k$, item-level transactions set where k is category relation level

$OD \leftarrow \emptyset$, individual occurrences
$CD \leftarrow \emptyset$, co-occurrences
$CF \leftarrow \emptyset$, confidence score of co-occurrences
for each $t \in \mathcal{T}_k$ **do**
    for each $c_k^{(i)} \in t$ **do**
        if $c_k^{(i)} \notin OD \;\&\; c_k^{(i)} \notin CD$ **then**
            $OD[c_k^{(i)}] \leftarrow 0$
            $CD[c_k^{(i)}] \leftarrow \emptyset$
        end if
        $OD[c_k^{(i)}] \leftarrow OD[c_k^{(i)}] + 1$
        for each $c_k^{(j)} \in t \;\&\; c_k^{(j)} \neq c_k^{(i)}$ **do**
            if $c_k^{(j)} \notin CD[c_k^{(i)}]$ **then**
                $CD[c_k^{(i)}, c_k^{(j)}] \leftarrow 0$
            end if
            $CD[c_k^{(i)}, c_k^{(j)}] \leftarrow CD[c_k^{(i)}, c_k^{(j)}] + 1$
        end for
    end for
end for
$CF[c_k^{(p)}, c_k^{(q)}] \leftarrow \frac{CD[c_k^{(p)}, c_k^{(q)}]}{OD[c_k^{(p)}]} \; \forall p, q \in CD$
return CF

3.5 Summary

In this chapter, an improved pairwise association rule-based recommendation algorithm PMAAR is presented that combines mined pairwise associations at both the item and category levels. PMAAR is able to leverage the strengths of both item-level and category-level pairwise associations to suggest more relevant recommendations. It was demonstrated how to build a pairwise multilevel association rule model, namely by the generation of category-level transaction data sets and the mining of pairwise association rules. This enables the model to identify relationships between items that may not be evident when looking only at the item-level data. At every level, beginning at the item level and progressing to the target level, we discuss the training process for mining association rules.

Chapter 4

Recommendations Based on Multilevel Association Rules

In general, recommendations based on association rules are built on the collective preferences of users, so they are not tailored to a particular user's preferences. Association rules look for relationships between items that have been purchased together, so they don't take into account the individual user's tastes, likes, and dislikes. This means that the recommendations are based on general trends and may not be the most suitable for any particular user. To introduce some degree of personalization, we incorporate category preferences derived from the user's in-session activity, such as clicking on items or visiting them.

In this chapter, the following topics are discussed. In section 4.1, the formal approach for deriving user preferences for a category level is presented based on the activity of the user during the session. Section 4.2 describes the algorithm for recommending items in accordance with the trained model and the derived preferences. In section 4.3, a working example is presented that simulates the recommendation process. In section 4.4, the chapter is summarized

4.1 In-session Category Preferences

In the recommendation process, PMARR first evaluates an individual active user's category level preferences for each of the m categories $\{c_k^{(1)}, c_k^{(2)}, \ldots, |c_k^{(m)}|\}$ at level $k > 0$. Given each of the categories incurred in the user u's in-session interactions with the system, we count the category occurrence frequencies in the interactions and store them in a frequency vector $\vec{v_u} = [v_u^1, v_u^2, \ldots, v_u^m]$, where v_u^i is the frequency count on the category $c_k^{(i)}$ given user u's in-session interactions. The frequency of each of the categories is used to compute the user's preference for that category using a softmax σ function as follows:

$$\sigma(\vec{v_u}, i) = \frac{e^{v_u^i}}{\sum_{q=1}^{m} e^{v_u^q}} \tag{6}$$

where $\sigma(\vec{v_u}, i)$ returns the preference score of user u on category i. The preference scores of user u computed by the softmax function on each of the categories at Level k are stored in the weighted preferences vector W_k^u, where $W_k^u = [\sigma(\vec{v_u}, 1), \ldots, \sigma(\vec{v_u}, m)]$.

4.2 Recommendation Process

The PMARR algorithm is based on the idea that users' preferences change over time and that it is important to take into account their most recent interactions when making recommendations. To do this, PMARR considers the set of items that the user interacted with during their current session. It then uses this information, along with pre-computed confidence scores and weighted preferences, to generate recommendations that take into account the user's current interests. PMARR also takes into account the category preferences of the user. It assigns greater weight to categories that the user has interacted with frequently during the session. The Algorithm 3 calculates a recommendation score for each item associated with an active item that the user interacted with during the session. It then uses these scores to recommend the top N items to the user. This helps to ensure that the recommendations are tailored to the user's specific interests. The PMARR algorithm is particularly useful in scenarios where users' preferences change frequently, such as in e-commerce or content recommendation systems. By taking into account a user's most recent interactions, PMARR can provide recommendations that are more relevant and up-to-date than other algorithms. After observing the set of items I_c that the active user u interacted in-session, PMARR uses pre-computed confidence scores CF and the weighted preferences $\{W_1^u, W_2^u, \ldots W_{L^{tar}}^u\}$ as well as I_c to produce recommendations. This approach also allows for greater flexibility in the recommendation process, as the system can easily adjust to changes in the user's preferences or behavior. Additionally, by incorporating the user's category level preferences, the system can generate recommendations that not only match the user's interests but also align with their preferred level of specificity. The recommendation score $RF[c_0^{(j)}]$ is calculated for each item $c_0^{(j)} \in C_0$ associated with an item interacted by an user in the session. For each level $k > 0$, the confidence score $CF_k[c_k^{(i)}, c_k^{(j)}]$ of pair association from $c_k^{(i)}$ to $c_k^{(j)}$ is multiplied by the weight preference $W_k^u[c_k^{(j)}]$ of the consequent entity $[c_k^{(j)}]$. In this way categories incurred frequently in the session are given greater weight, incorporate user's category level preferences into the recommendation process. The PMARR algorithm is described in Algorithm 3. The algorithm recommends the top N items based on the RF scores calculated by Line 7 in Algorithm 3.

Algorithm 3 Recommend($I_c, CF, N, \{W_1^u, W_2^u, \ldots, W_{L^{tar}}^u\}$)

Require: $I_c \subseteq I$, Items with which the user has interacted during a session
Require: CF, confidence scores of each level i.e., $\{CF_0, CF_1, \ldots CF_k\} \in CF$
Require: N, Number of recommendations
Require: $W_{1,2,\ldots,L^{tar}}^u$, weighted category preference based on user session

```
1:  Initilize RF ← ∅, recommendation vector
2:  for each c_0^(i) ∈ I_c do
3:      for each c_0^(j) ∈ CF_0[c_0^(i)] do
4:          if c_0^(j) ∉ RF then
5:              RF[c_0^(j)] ← 0
6:          end if
7:          RF[c_0^(j)] ← RF[c_0^(j)] + CF_0[c_0^(i), c_0^(j)] + Σ_{k=1}^{L^tar} CF_k[c_k^(i), c_k^(j)] * W_k^u[c_k^(j)]
8:      end for
9:  end for
10: return Top N of RF
```

4.3 Worked Example

An example of a PMARR recommendation process is presented in the following section. In this example we have 5 items $\{i_1, i_2, i_3, i_4, i_5\}$ at level 0. Assume that there are three categories $\{A, B, C\}$ at Level 1. As a result, each item at level 0 will be assigned to one of the categories $\{A, B, C\}$ at level 1. This will continue from one level to another until the desired level of categorization is reached. For the sake of simplicity, we have only considered two levels in this example.

Item level transactions	Category level transactions
1· $\{\mathbf{i_1}, \mathbf{i_2}, \mathbf{i_5}\}$	1· $\{\mathbf{A}, \mathbf{B}\}$
2· $\{\mathbf{i_1}, \mathbf{i_2}, \mathbf{i_3}\}$	2· $\{\mathbf{A}, \mathbf{B}, \mathbf{C}\}$
3· $\{\mathbf{i_1}, \mathbf{i_2}\}$	3· $\{\mathbf{A}, \mathbf{B}\}$
4· $\{\mathbf{i_1}, \mathbf{i_2}, \mathbf{i_4}\}$	4· $\{\mathbf{A}, \mathbf{B}\}$
5· $\{\mathbf{i_1}, \mathbf{i_3}, \mathbf{i_4}, \mathbf{i_5}\}$	5· $\{\mathbf{A}, \mathbf{B}, \mathbf{C}\}$

Table 4.1: Item and category level transactions

$\{i_1, i_4\}$ belong to A, $\{i_2, i_5\}$ belong to B and $\{i_3\}$ belong to C. Table 4.1 shows item-level transactions and converted category-level transactions using hierarchy mapping. Item-level transactions are those that are recorded at an individual item, whereas category-level transactions are those that are aggregated and recorded at the category level. Based on the transaction sets at the item and category levels, we can calculate confidence scores or conditional probabilities for all possible pairs. In Table 4.2, pairwise associations are shown at both the item and category levels, along with their confidence scores. The confidence scores help to quantify the strength of the relationships and provide more meaningful insights into the data. Following the training of the model, recommendations are generated based on analyzing the interactions of the users during the session. Our example assumes that the user has interacted with items i_1, i_5. For each item incurred in the user u's interaction with the system in session, we count the frequency of category occurrences in the interactions and store them in a frequency vector. Here, A has a frequency count of 1, B has a frequency count of 1 and C has a frequency count of 0. After using the softmax function on the frequency vector, the relative preference for each category is calculated as follows: $A(0.42), B(0.42), C(0.16)$. In the recommendation phase, the RF score for each item in the itemset except for the items in the session is calculated. Specifically, the score comprises the confidence scores multiplied by the preference for the item's category at each level except level 0. The RF score of the potential recommendations are $\mathbf{i_2(1.72), i_3(1.03), i_4(1.32)}$. Based on the score, the top recommendation is $\mathbf{i_2}$

Item level model	Category level model	User in-session interactions and preference weights	Recommendations
1· $\mathbf{i_2} \rightarrow \mathbf{i_1}$ 1.0 2· $\mathbf{i_1} \rightarrow \mathbf{i_2}$ 0.8 3· $\mathbf{i_3} \rightarrow \mathbf{i_1}$ 1.0 4· $\mathbf{i_1} \rightarrow \mathbf{i_3}$ 0.4 5· $\mathbf{i_4} \rightarrow \mathbf{i_1}$ 1.0 6· $\mathbf{i_1} \rightarrow \mathbf{i_4}$ 0.4 7· $\mathbf{i_5} \rightarrow \mathbf{i_1}$ 1.0 8· $\mathbf{i_1} \rightarrow \mathbf{i_5}$ 0.4 9· $\mathbf{i_3} \rightarrow \mathbf{i_2}$ 0.5 10· $\mathbf{i_2} \rightarrow \mathbf{i_3}$ 0.25 11· $\mathbf{i_4} \rightarrow \mathbf{i_2}$ 0.5 12· $\mathbf{i_2} \rightarrow \mathbf{i_4}$ 0.25 13· $\mathbf{i_5} \rightarrow \mathbf{i_2}$ 0.5 14· $\mathbf{i_2} \rightarrow \mathbf{i_5}$ 0.25 15· $\mathbf{i_4} \rightarrow \mathbf{i_3}$ 0.5 16· $\mathbf{i_3} \rightarrow \mathbf{i_4}$ 0.5 17· $\mathbf{i_5} \rightarrow \mathbf{i_3}$ 0.5 18· $\mathbf{i_3} \rightarrow \mathbf{i_5}$ 0.5 19· $\mathbf{i_5} \rightarrow \mathbf{i_4}$ 0.5 20· $\mathbf{i_4} \rightarrow \mathbf{i_5}$ 0.5	1· **A** $\rightarrow$ **B** 1.0 2· **B** $\rightarrow$ **A** 1.0 3· **C** $\rightarrow$ **A** 1.0 4· **A** $\rightarrow$ **C** 0.4 5· **C** $\rightarrow$ **B** 1.0 6· **B** $\rightarrow$ **C** 0.4	In the session, user interacted $\{i_1, i_5\}$. The user's preference weights for the three categories are $\mathbf{A}(\mathbf{0.42})$, $\mathbf{B}(\mathbf{0.42})$, $\mathbf{C}(\mathbf{0.16})$	The RF scores are $\mathbf{i_2}(\mathbf{1.72}), \mathbf{i_3}(\mathbf{1.03}), \mathbf{i_4}(\mathbf{1.32})$. Top 1 recommendation is $\mathbf{i_2}$

Table 4.2: The process of PMARR recommendation on the example dataset

4.4 Summary

In this chapter, we explored a process for evaluating an active user's preferences for each of the categories in order to introduce personalization by identifying the most relevant items based on their in-session behavior. Additionally, we discussed how the system may dynamically adjust the user's preferences over time, in order to provide an even more tailored experience. Next, we outlined the different steps of the process, from data collection to training the model to generating recommendations. We also discussed how the system is able to take into account the user's changing preferences and adjust the model accordingly. Finally, we provided an example of how the system works.

Chapter 5

Experiments and Results

In the previous chapter, a rule-based recommender system was developed using pairwise multi-level association algorithm. The purpose of this section is to conduct an experiment with PMARR. We demonstrate how our solution addresses the requirements, one by one. We will also demonstrate the merits of the framework solution and explain how it is superior to the other solution as a whole through an experiment. A description of the experimental setup is provided in the first section. A description of the results of the experiment is presented in the second section. In the final section, we summarize our observations regarding the results of the experiment.

5.1 Experiments

5.1.1 Dataset

PMARR is designed for cold-start situations where customers visit an e-commerce platform anonymously and the platform only has the data of historic basket transactions from customers. For an active customer in a session, the platform also tracks their in-session interaction, such as item selection or browsing. As an example, when building Bidgala, a website that sells artwork online, we had many page visits and interactions, but most of them were done by anonymous users (users who did not log in). In those situations, it was not possible to collect explicit user data, such as ratings or user profiles. Since the data from Bidgala [1] is not open for our academic publication and

[1]https://www.bidgala.com

we did not find other suitable open data sets which contain separated in-session interactions, we adopt the Restaurant and Consumer data set from the UCI Machine Repository (Medelln [2012]) to our setting by assuming a part of a customer's restaurant visits as their in-session interactions when recommending restaurants for the customer. In terms of transactions, the Restaurant and Consumer data set contains 1161 user rating records provided by 138 customers on their restaurant visits. At the item level, the data set contains 769 restaurants. At the category level, it contains 59 cuisines. Each restaurant exclusively belongs to a cuisine, which satisfies the assumption of our category hierarchy structure. Table 5.1 summarizes the statistics of this data set.

Types	**Total No.**	**Average No. (per Restaurant)**	**Average No. (per Consumer)**
Consumers	138	-	-
Restaurant	769	-	7.55
Payment	5	2.13	-
Cuisine	59	-	2.39
Rating	1161	8.93	8.41

Table 5.1: A statistical analysis of the Restaurant Consumer dataset

5.1.2 Experimental Procedure

Using the data set, We compare the performance of PMARR with three closely related association rule-based recommendation algorithms, namely basic Association Rules (AR), Multilevel Association Rules (MAR), and Pairwise Association Rules (PAR). AR makes recommendations based on regular association rules in which all combinations of the observed items are considered in the antecedents. In our experiment, we use FP-growth algorithm (frequent patterns algorithm) Li et al. [2006] to mine AR associations. FP-growth is based on building a frequent-pattern tree structure that is both efficient and scalable for mining association rules. The MAR algorithm is a pure multilevel association rule-based recommendation algorithm. It is adapted from Kim and Kim [2003] to operate on our category hierarchy structure. PAR is a recent pairwise association rule-based recommendation algorithm Osadchiy et al. [2019]. PMARR improves the performance of PAR by incorporating category-level user preference information into the recommendation score calculation.

All algorithms are tested using the same training and testing data splits. Monte Carlo cross-validation splits are used. Unlike K-fold cross-validation splits, Monte Carlo cross-validation splits the data set not by groups or folds, but by random splits. In each split, we randomly sample 100 consumers to train a model and the remaining 38 consumers to test the model. In order to reduce the impact of noise and make sure that the result is convergent, multiple splits are performed in the experiments. Every trained model generates a set of rules that include associations between restaurants as well as those between cuisines. For transactions incurred by a user from a test split, we consider the first few visited restaurants as in-session items interacted with by the user. In the experiments, the number of in-session items ranges from 2 to 4 items. After determining the in-session items, we hide other items for a user in the test set. In this way, we create in-session test data from the original data set and use the hidden items as the ground truth to verify the performance of the recommendation algorithms.

For recommendation performance evaluation, recall measurements are used to determine the proportion of relevant items that have been recommended Bellogin et al. [2011]. Using recall statistics, we calculated the percentage of restaurants that were correctly predicted in relation to the total number of restaurants visited by the consumer in our experiment. The true positive predictions are those that were found in the set of restaurants visited by the consumer. To assess the quality of a set of recommendations, a normalized discounted cumulative gain (nDCG) metric is used Burges et al. [2005]. In a recommendation, the most relevant items should appear first, followed by the medium relevant items, and then the irrelevant items. In this context, we assess the quality of restaurants that are recommended to consumers. In general, the formula is as follows:

$$nDCG = \frac{DCG}{IDCG} \tag{7}$$

$$DCG = \sum_{i=1}^{K} \frac{(2^{r^{(i)}} - 1)}{log(i+1)} \tag{8}$$

where $r^{(i)}$ is the relevance score of the i^{th} restaurant. Relevance is determined by using a value of 0 for an incorrect prediction as well as a value of 1 for an accurate prediction [Osadchiy et al., 2019]. As a result, the ideal Discounted Cumulative Gain (IDCG) in our case is always 1, which corresponds to a single correct prediction. The standard FP-growth association rule mining task was completed using an open-source data mining software library SPMF (Fournier-Viger et al. [2016]). We set both the minimum support and minimum confidence to the lowest value (0.0001). This will allow us to complete the mining process of the data set on our machine with as many association rules as possible. The evaluation was conducted on a Mac Pro (3.2 GHz 8-core M1, 16 GB).

5.1.3 Analysis of Results

In this study, we propose a recommendation algorithm based on pairwise multilevel association rules and which addresses the cold-start problem in its entirety. According to the observed results, the proposed algorithm performs well in comparison with current state of the art systems as well as meets the relevant criteria. Using a rule-based model and category preference utilization approach, the implemented algorithm funnels restaurants that have co-occurred at different levels of the category hierarchy together, thereby making the system more robust. We present the evaluation results of the performance of PMARR in comparison with that of AR, MAR, and PAR.

In-session Activities	Algorithm	Top 2	Top 5	Top 10	Top 15	Top 20
2	PMARR	0.24	0.41	0.64	0.74	0.80
	PAR	0.12	0.31	0.53	0.53	0.76
	MAR	0.19	0.34	0.58	0.69	0.75
	FP-growth	0.14	0.26	0.29	0.31	0.34
3	PMARR	0.2	0.42	0.64	0.66	0.85
	PAR	0.15	0.31	0.53	0.56	0.79
	MAR	0.15	0.36	0.58	0.73	0.82
	FP-growth	0.19	0.35	0.29	0.42	0.44
4	PMARR	0.2	0.37	0.64	0.77	0.84
	PAR	0.15	0.3	0.52	0.66	0.79
	MAR	0.15	0.33	0.56	0.71	0.8
	FP-growth	0.12	0.27	0.41	0.49	0.5

Table 5.2: An analysis of the recall scores of the PMARR, MAR, PAR, and Fp-Growth algorithms

Recall score is a metric used to evaluate the effectiveness of recommendation systems. It measures the percentage of relevant items that a recommendation system is able to retrieve out of all the relevant items available. In other words, it quantifies how well a recommendation system is able to identify the items that a user is interested in, based on their input activities. The provided results presents recall scores for generating top 2, 5, 10, 15, and 20 recommendations for each of the four algorithms, based on different input activity scenarios. The results show that PMARR algorithm performs the best in most cases, followed by PAR and MAR algorithms. FP-growth algorithm performs the worst in most cases. When the number of in-session activities is two, PMARR has the highest recall score for generating top 2, 5, and 10 recommendations. For generating top 15 and 20 recommendations, PMARR still has the highest scores, but PAR and MAR have similar recall scores. This suggests that PMARR is a suitable algorithm for generating recommendations for in-session activities that involve two activities. For the number of in-session activities of three, PMARR has the highest recall score for generating top 2 and 5 recommendations, and the second-highest recall score for generating top 10, 15, and 20 recommendations. PAR has the highest recall score for generating top 10 recommendations, but its recall score for generating top 15 and 20 recommendations is lower than PMARR and MAR. This indicates that PMARR is a good choice for generating recommendations for in-session activities involving three activities, but PAR may be more appropriate for generating top 10 recommendations. When the number of in-session activities is four, PMARR has the highest recall score for generating top 2 and 10 recommendations, and the second-highest recall scores for generating top 5, 15, and 20 recommendations. PAR has the highest recall score for generating top 15 and 20 recommendations, but its recall score for generating top 2, 5, and 10 recommendations is lower than PMARR and MAR. This suggests that PMARR is a good choice for generating recommendations for in-session activities involving four activities, and PAR may be more suitable for generating top 15 and 20 recommendations. With an increase in the number of recommendations, the recall score in Figure 5.1 for PMARR increased linearly. The recall score remains relatively unchanged when the number of in-session items as input is increased, due to the fact that the recommendations generated by the algorithm are unique to the users. Even though they may be relevant to the user based on their in-session activities, they may not have been seen previously by the user. Serendipity is an important component of recommender systems that can enhance users' satisfaction and engagement. As a result of incorporating diversity and novelty into the recommendation process, PMARR introduce users to items they may otherwise not have been exposed. As shown in Figure 5.1, PMARR has achieved an average increase of 31%, 18%, 12%, 8%, and 5% over MAR in terms of recall for each value of Top-N at three different ranges of in-session items. Compared with PAR at different number of in-session items, PMARR showed similar trends with 55%, 30%, 20%, 15%, and 7% higher recall scores. The standard AR algorithm FP-Growth has made very poor recommendations. It's performance is 40%, 38%, 81%, 87%, 98% lower than that of PMARR in terms of recall.

Top N	Algorithm	In-session activities: 2	In-session activities: 3	In-session activities: 4
5	PMARR	0.38	0.37	0.34
	PAR	0.31	0.30	0.30
	MAR	0.31	0.30	0.30
	FP-growth	0.23	0.24	0.21
10	PMARR	0.49	0.49	0.47
	PAR	0.38	0.39	0.39
	MAR	0.43	0.41	0.41
	FP-growth	0.26	0.28	0.27
15	PMARR	0.54	0.54	0.52
	PAR	0.44	0.45	0.44
	MAR	0.48	0.47	0.46
	FP-growth	0.27	0.29	0.28

Table 5.3: An analysis of the nDCG scores of the PMARR, MAR, PAR, and Fp-Growth algorithms

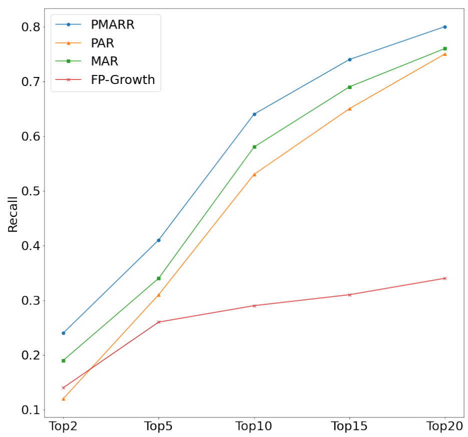
To evaluate both the relevance of the recommended items and their position in the list, we used nDCG scores (normalized discounted cumulative gain) for 5, 10, and 15 top N recommendations. The nDCG score is a widely used metric for measuring the effectiveness of a recommender system. We used two, three, and four in-session activities as input to the recommender systems, and evaluated them for different values of N (i.e., 5, 10, and 15). The results of our evaluation showed that PMAAR consistently outperformed the other three recommender systems for all values of N and for different numbers of in-session activities used as input. Specifically, for the case of 5 top N recommendations with two in-session activities as input, PMAAR scored 0.38, while the other systems scored lower, with MAR scoring 0.31, PAR scoring 0.27, and FP-Growth scoring 0.23. Similarly, for the case of 15 top N recommendations with four in-session activities as input, PMAAR scored 0.52, while the other systems scored lower, with MAR scoring 0.46, PAR scoring 0.44, and FP-Growth scoring 0.28. Interestingly, we observed that the performance of all four systems decreased as the value of N increased. This suggests that it becomes more challenging for the systems to recommend relevant items as the number of recommended items increases. Overall, our study provides valuable insights into the performance of different recommender systems using nDCG scores. Our results suggest that PMAAR is the best-performing system among the four evaluated systems for recommending items to users based on their in-session activities. In Figure 5.2, we can see the results of the nDCG scores. Considering three different ranges of in-session items, the PMARR score for each value of Top-N is 15%, 21%, and 13% higher on average compared with the MAR score for each value of Top-N. Similarily, it scores 31%, 24%, and 19% higher on average than PAR for each value of Top-N for each range of in-session items for each value of Top-N. In comparison to the other approaches, FP-Growth performed poorly. Compared to the FP-Growth nDCG score, PMARR has an average increase of 88%, 74%, and 76%.

To summarize, PMARR is designed to be able to adapt to the changing preferences of users over time. By incorporating both the most recent set of items that a user has interacted with and their category preferences, PMARR can provide more accurate and personalized recommendations compared to other algorithms. Overall, PMARR represents a significant improvement over traditional recommendation algorithms by incorporating the changing preferences and interests of users over time. It is a powerful tool for businesses looking to provide personalized recommendations to their customers and can help to increase customer satisfaction and loyalty.

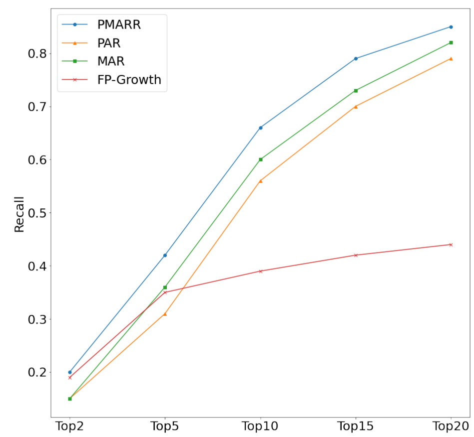
5.2 Threats to Validity

Several factors may contribute to the potential interpretation of the results as presented as partially invalid. It's important to acknowledge them in the thesis. Here are the threats to validity:

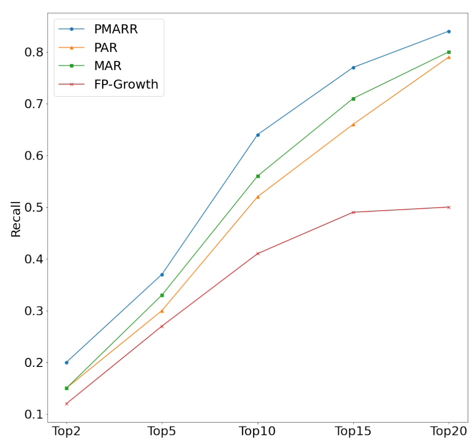
- Experimental results are based on the author's own interpretation of competing algorithms. The author evaluates the performance of different algorithms using a variety of metrics and experiments in order to determine which algorithm performs the best in a particular situation. These results are then used to make conclusions about which algorithm is the most suitable for a particular task.
- There is a difference between the dataset used for the comparison and the dataset used by the competing solutions. In order to apply multi-level associations, the experiment requires a dataset that contains categorical hierarchy. Data-sets used in competing solutions are either



(a) Number of in-session items : 2

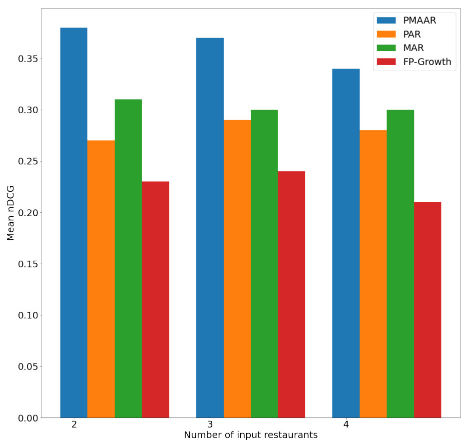


(b) Number of in-session items : 3

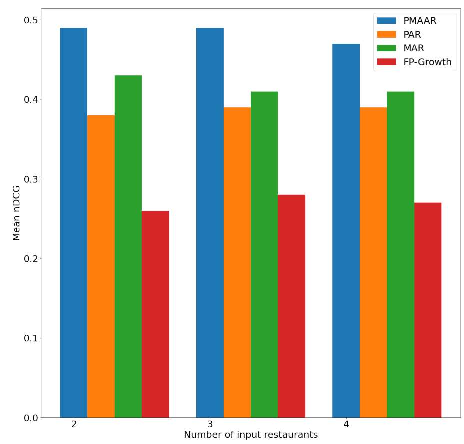


(c) Number of in-session items : 4

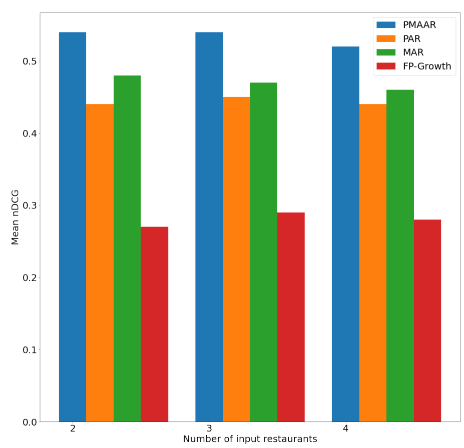
Figure 5.1: Recall curve for different number of in-session items



(a) Top 5 recommendations



(b) Top 10 recommendations



(c) Top 15 recommendations

Figure 5.2: Ratio of mean nDCG for top N recommendations to the number of in-session items

proprietary or do not contain the data necessary for the analysis. Due to this, we had to use a different dataset that met our needs.

- In the experiments, only one data set of relatively small size is used. We chose the dataset based on its accessibility, as it is publicly available. We wanted to ensure that the experiments could be easily repeated and verified, so we selected a dataset that was readily available and of a size that would not cause any significant strain on computing resources. Additionally, the dataset contains the necessary categorization of items, which was key to the experiments being conducted.

- Despite the algorithm's claim that it supports multi-level categories, the tests only use one level of categories for items. Due to the fact that the data set used in our experiments only supports one level of meaningful categories, only one category level is involved in our experiments. Furthermore, adding more category levels does not necessarily translate into better recommendations. The relevance of items recommended using higher-level association rules to observed user selections will decrease since higher-level association rules will bring in a category of items to be recommended. Therefore, the system designer needs to decide how high in the hierarchy the transaction sets should be generated and mined, that is to determine the target value.

5.3 Summary

This chapter concludes the section of the thesis that deals with experimentation and evaluation metrics. The section explored methodological approaches to experimentation and the two metrics that can be used to evaluate the results. We found that our proposed approach outperformed three other existing algorithms, namely MAR, PAR, and FP-Growth. Our experiments provided evidence that our proposed approach is more accurate, efficient, and robust than existing algorithms. Moreover, our proposed approach produced results that were consistent across different experiments, indicating that it is a reliable and effective approach for dealing with the problem at hand. Next, we will discuss the conclusion and future scope of the research.

Chapter 6

Conclusion

As we have shown throughout this thesis, association rules can be used to overcome the cold-start problem for new users in recommender systems. In this section, we provide a summary of our work.

6.1 Overview

Our first objective was to establish the issue of new user cold start as the core problem of interest in recommender systems. In order to mitigate the problem of cold-start, many existing methods rely on demographics and social relationships to predict customer preferences. However, there are a number of scenarios in which existing approaches may prove to be challenging. A few of these situations include situations where users interact anonymously, applications that require a high degree of privacy, and difficulties in constructing models of personal behavior and preferences. To make accurate recommendations with limited interaction, we must solve the cold start problem for new users. Secondly, association rule-based framework is proposed to analyze consumer visits to restaurants and recommend restaurants based on those visits.

This thesis proposes a pairwise multilevel association rule-based recommendation approach to the cold-start problem. The proposed approach does not require personal user preference data and rating matrixes. It also reduces the computation required for rule mining by avoiding the need to mine all possible association rules. By allowing category-level associations and users' in-session activities to be integrated into the recommendation process, the proposed approach outperforms standard association rule-based and recent pairwise association rule-based recommendation approaches.

6.2 Limitations

Despite the fact that the solution proposed in this thesis meets the requirements and achieves the goals, some limitations still remain:

- If the amount of data read and processed exceeds the amount of memory available, an error will occur. Our framework does not yet contain a solution that we already have. It is possible to use packages that support multi-processing or distributed processing.
- In order to find the most appropriate higher order attribute, our framework must undergo a series of trials and errors. The process also involves the acquisition of domain knowledge.
- In our framework, we are solely dependent on the past behavior of consumers. Ratings and explicit feedback are completely ignored. A hybrid solution that utilizes explicit feedback is required.
- As of now, we have not conducted many tests on the scalability of the framework. Increasing the number of higher order levels, for example, will result in an increase in our training speed.

6.3 Future Work

In the long run, such work should lead to an impactful change in the field of recommendation systems. As well as addressing the limitations described in the previous section, we outline several areas of future work in more detail:

- We plan to address the potential threats to validity identified in this study to further validate the experimental results. One possible approach is to use multiple datasets with varying levels of categorical hierarchy to evaluate the performance of different algorithms. In-addition, explore the use of more advanced methods for evaluating the suitability of different algorithms, such as machine learning techniques. By addressing these potential limitations, future research can provide more robust and reliable insights into the suitability of different algorithms for the particular task.

- We plan to extend the proposed pairwise multilevel association model to incorporate associations among content-related attributes, which, we expect, can further improve the recommendation performance in cold-start situations.

- Our framework is written in Python, but can only be accessed via the command line. In order to integrate it into multiple domains, we must provide a rest API

- For the generation of pair-wise association rules, our system utilizes the SPMF tool. The scripts must be run manually. It is our intention to automate the complete process.

Bibliography

C. C. Aggarwal et al. *Recommender systems*, volume 1. Springer, 2016.

S. B. Aher and L. Lobo. Mining association rule in classified data for course recommender system in e-learning. *International Journal of Computer Applications*, 39(7):1–7, 2012.

A. Bellogin, P. Castells, and I. Cantador. Precision-oriented evaluation of recommender systems: an algorithmic comparison. In *Proceedings of the fifth ACM conference on Recommender systems*, pages 333–336, 2011.

A. Bellogín, P. Castells, and I. Cantador. Neighbor selection and weighting in user-based collaborative filtering: a performance prediction approach. *ACM Transactions on the Web (TWEB)*, 8(2): 1–30, 2014.

Bendakir and Narimel. Using association rules for course recommendation. In *Proceedings of the AAAI workshop on educational data mining*, volume 3, pages 1–10, 2006.

C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender. Learning to rank using gradient descent. In *Proceedings of the 22nd international conference on Machine learning*, pages 89–96, 2005.

W. Carrer-Neto, M. L. Hernández-Alcaraz, R. Valencia-García, and F. García-Sánchez. Social knowledge-based recommender system. application to the movies domain. *Expert Systems with applications*, 39(12):10990–11000, 2012.

T. Chellatamilan and R. Suresh. An e-learning recommendation system using association rule mining technique. *European Journal of Scientific Research*, 64(2):330–339, 2011.

J. Chicaiza and P. Valdiviezo-Diaz. A comprehensive survey of knowledge graph-based recommender systems: Technologies, development, and contributions. *Information*, 12(6):232, 2021.

L. O. Colombo-Mendoza, R. Valencia-García, A. Rodríguez-González, G. Alor-Hernández, and J. J. Samper-Zapater. Recommetz: A context-aware knowledge-based mobile recommender system for movie showtimes. *Expert Systems with Applications*, 42(3):1202–1222, 2015.

W. DuMouchel and D. Pregibon. Empirical bayes screening for multi-item associations. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 67–76, 2001.

F. Fauzan, D. Nurjanah, and R. Rismala. Apriori association rule for course recommender system. *Indonesia Journal on Computing (Indo-JC)*, 5(2):1–16, 2020.

P. Fournier-Viger, J. C.-W. Lin, A. Gomariz, T. Gueniche, A. Soltani, Z. Deng, and H. T. Lam. The spmf open-source data mining library version 2. In *Joint European conference on machine learning and knowledge discovery in databases*, pages 36–40. Springer, 2016.

H. Garcia-Molina. *Database systems: the complete book*. Pearson Education India, 2008.

D. Goldberg, D. Nichols, B. M. Oki, and D. Terry. Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 35(12):61–70, 1992.

M. Hahsler. arulesviz: Interactive visualization of association rules with r. *R J.*, 9(2):163, 2017.

J. Herce-Zelaya, C. Porcel, J. Bernabé-Moreno, A. Tejeda-Lorente, and E. Herrera-Viedma. New technique to alleviate the cold start problem in recommender systems using information from social media and random decision forests. *Information Sciences*, 536:156–170, 2020.

S. Huang and L. Duan. E-commerce recommendation algorithm based on multi-level association rules. In *Advances in Electronic Commerce, Web Application and Communication*, pages 479–485. Springer, 2012.

P. Jomsri. Fucl mining technique for book recommender system in library service. *Procedia Manufacturing*, 22:550–557, 2018.

B. M. Kim, Q. Li, J.-W. Kim, and J. Kim. A new collaborative recommender system addressing three problems. In *PRICAI 2004: Trends in Artificial Intelligence: 8th Pacific Rim International Conference on Artificial Intelligence, Auckland, New Zealand, August 9-13, 2004. Proceedings 8*, pages 495–504. Springer, 2004.

C. Kim and J. Kim. A recommendation algorithm using multi-level association rules. In *Proceedings IEEE/WIC International Conference on Web Intelligence (WI 2003)*, pages 524–527. IEEE, 2003.

Y. Koren. Collaborative filtering with temporal dynamics. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 447–456, 2009.

Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.

C. W.-k. Leung, S. C.-f. Chan, and F.-l. Chung. Applying cross-level association rule mining to cold-start recommendations. In *2007 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology-Workshops*, pages 133–136. IEEE, 2007.

X. Li, Z.-H. Deng, and S. Tang. A fast algorithm for maintenance of association rules in incremental databases. In *Advanced Data Mining and Applications: Second International Conference, ADMA 2006, Xi'an, China, August 14-16, 2006 Proceedings 2*, pages 56–63. Springer, 2006.

B. Lika, K. Kolomvatsos, and S. Hadjiefthymiades. Facing the cold start problem in recommender systems. *Expert Systems with Applications*, 41(4):2065–2073, 2014.

Q. Liu, Y. Zeng, R. Mokhosi, and H. Zhang. Stamp: short-term attention/memory priority model for session-based recommendation. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 1831–1839, 2018.

M. Ludewig and D. Jannach. Evaluation of session-based recommendation algorithms. *User Modeling and User-Adapted Interaction*, 28:331–390, 2018.

J. Medelln, Rafael Serna. Restaurant consumer data. UCI Machine Learning Repository, 2012.

R. Nahta, Y. K. Meena, D. Gopalani, and G. S. Chauhan. Embedding metadata using deep collaborative filtering to address the cold start problem for the rating prediction task. *Multimedia Tools and Applications*, 80:18553–18581, 2021.

M. K. Najafabadi, M. N. Mahrin, S. Chuprat, and H. M. Sarkan. Improving the accuracy of collaborative filtering recommendations using clustering and association rules mining on implicit data. *Computers in Human Behavior*, 67:113–128, 2017.

U. Ocepek, J. Rugelj, and Z. Bosnić. Improving matrix factorization recommendations for examples in cold start. *Expert Systems with Applications*, 42(19):6784–6794, 2015.

T. Osadchiy, I. Poliakov, P. Olivier, M. Rowland, and E. Foster. Recommender system based on pairwise association rules. *Expert Systems with Applications*, 115:535–542, 2019.

M. J. Pazzani. A framework for collaborative, content-based and demographic filtering. *Artificial intelligence review*, 13(5):393–408, 1999.

G. Piatetsky-Shapiro. Discovery, analysis, and presentation of strong rules. *Knowledge discovery in databases*, pages 229–238, 1991.

T. Raeder and N. V. Chawla. Market basket analysis with networks. *Social network analysis and mining*, 1(2):97–113, 2011.

P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl. Grouplens: An open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 ACM conference on Computer supported cooperative work*, pages 175–186, 1994.

B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*, pages 285–295, 2001.

V. K. Sejwal, M. Abulaish, et al. Crecsys: A context-based recommender system using collaborative filtering and lod. *IEEE Access*, 8:158432–158448, 2020.

G. Shaw, Y. Xu, and S. Geva. Using association rules to solve the cold-start problem in recommender systems. In *Pacific-Asia conference on knowledge discovery and data mining*, pages 340–347. Springer, 2010.

J. Shokeen. A comparison of collaborative filtering-based recommender systems. *Journal of Emerging Technologies and Innovative Research*, 5(4):868–871, 2018.

H. Sobhanam 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*, pages 1–5, 2013. doi: 10.1109/ICCCI.2013.6466121.

K. Tsuji, N. Takizawa, S. Sato, U. Ikeuchi, A. Ikeuchi, F. Yoshikane, and H. Itsumura. Book recommendation based on library loan records and bibliographic information. *Procedia-social and behavioral sciences*, 147:478–486, 2014.

I. Viktoratos, A. Tsadiras, and N. Bassiliades. A context-aware web-mapping system for group-targeted offers using semantic technologies. *Expert Systems with Applications*, 42(9):4443–4459, 2015.

I. Viktoratos, A. Tsadiras, and N. Bassiliades. Modeling human daily preferences through a context-aware web-mapping system using semantic technologies. *Pervasive and Mobile Computing*, 38: 14–40, 2017.

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*, 101:78–90, 2018.

Y. Wakita, K. Oku, H.-H. Huang, and K. Kawagoe. A fashion-brand recommender system using brand association rules and features. In *2015 IIAI 4th International Congress on Advanced Applied Informatics*, pages 719–720. IEEE, 2015.

H. Wang, N. Wang, and D.-Y. Yeung. Collaborative deep learning for recommender systems. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1235–1244, 2015.

Y. Wang, S. C.-F. Chan, and G. Ngai. Applicability of demographic recommender system to tourist attractions: a case study on trip advisor. In *2012 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology*, volume 3, pages 97–101. IEEE, 2012.

J. Wei, J. He, K. Chen, Y. Zhou, and Z. Tang. Collaborative filtering and deep learning based recommendation system for cold start items. *Expert Systems with Applications*, 69:29–39, 2017.

W. Woerndl and J. Schlichter. Introducing context into recommender systems. In *Proceedings of AAAI workshop on recommender systems in E-commerce*, pages 138–140, 2007.

J. Xiao, M. Wang, B. Jiang, and J. Li. A personalized recommendation system with combinational algorithm for online learning. *Journal of ambient intelligence and humanized computing*, 9: 667–677, 2018.

F. Yu, Q. Liu, S. Wu, L. Wang, and T. Tan. A dynamic recurrent model for next basket recommendation. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 729–732, 2016.

C. Zhang, K. Wang, H. Yu, J. Sun, and E.-P. Lim. Latent factor transition for dynamic collaborative filtering. In *Proceedings of the 2014 SIAM international conference on data mining*, pages 452–460. SIAM, 2014.

S. Zhang, L. Yao, A. Sun, and Y. Tay. Deep learning based recommender system: A survey and new perspectives. *ACM computing surveys (CSUR)*, 52(1):1–38, 2019.

Y. Zhang, Z. Liu, and C. Sang. Unifying paragraph embeddings and neural collaborative filtering for hybrid recommendation. *Applied Soft Computing*, 106:107345, 2021.

K. Zhou, S.-H. Yang, and H. Zha. Functional matrix factorizations for cold-start recommendation. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 315–324, 2011.