

Review

Not peer-reviewed version

A Survey of Sequential Pattern Based E-Commerce Recommendation Systems

[Christie I. Ezeife](#)^{*} and Hemni Sri-Rajeswari Karlapalepu

Posted Date: 24 August 2023

doi: 10.20944/preprints202308.1728.v1

Keywords: Recommendation systems; collaborative filtering; sequential patterns; e-commerce; purchase and clickstream history



Preprints.org is a free multidiscipline platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Article

SP-Based EC RecSys survey A Survey of Sequential Pattern Based E-Commerce Recommendation Systems

Christie Ezeife ^{1,2,*} and Hemni Sri-Rajeswari Karlapalepu ^{1,2,†}

¹ School of Computer Science, University of Windsor, 401 Sunset Ave., Windsor, N9B3P4, ON, Canada; hemnisrirajeswarik@gmail.com

² School of Computer Science, University of Windsor, 401 Sunset Ave., Windsor, N9B3P4, ON, Canada

* Correspondence: cezeife@uwindsor.ca

† These authors contributed equally to this unpublished original work.

Abstract: E-commerce recommendation systems usually deal with massive customer sequential databases, such as historical purchase or click stream sequences. Recommendation systems' accuracy can be improved if complex sequential patterns of user purchase behavior are learned by integrating sequential patterns of customer clicks and/or purchases into the user-item rating matrix input of collaborative filtering. This reviews focuses on algorithmic techniques of existing E-commerce recommendation systems that are sequential pattern based such as ChoRec05, ChenRec09, HuangRec09, LiuRec09, ChoiRec12, Hybrid Model RecSys16, Product RecSys16, SainiRec17, HPCRec18 and HSPCRec19. It provides a comprehensive and comparative performance analysis of these systems, exposing their methodologies, achievements, limitations, and potentials for solving more important problems in this domain. The review showed that integrating sequential pattern mining of historical purchase and/or click sequences into user-item matrix for collaborative filtering (i) improved recommendation accuracy (ii) reduced user-item rating data sparsity (iii) increased novelty rate of recommendations and (iv) improved scalability of the recommendation system.

Keywords: e-commerce recommendation systems; collaborative filtering; sequential patterns; e-commerce; purchase and clickstream history

1. Introduction

Recommendation systems have become the heart of many Internet-based companies such as Google, YouTube, Facebook, Netflix, LinkedIn, and Amazon. Recommendation systems provide suggestions for items that can be of use to a user. The suggestions provided are aimed at supporting users in various decision-making processes, such as what items to buy, what music to listen to, or what news to read ([1–3]). Pattern mining consists of discovering interesting, useful, and unexpected patterns in the databases through tasks like association rule mining, frequent pattern mining and sequential pattern mining [4]. These data mining tasks are generally used by recommendation systems to generate a meaningful representation and learning of historical user purchase data. This work focuses on systems that mine sequential patterns of customer purchase history for purposes of making recommendation in e-commerce application domain. Different types of recommendation systems accept different input data through explicit feedback (e.g., Table 1) and implicit feedback (e.g., Table 3). Explicit feedback can be in the form of collecting ratings of products or text comments by users through registration form/asking explicitly for interests and preferences, where users select numeric values from a specific evaluation system (e.g., a five-star rating system) to specify their likes and dislikes of different items. Implicit feedback includes behaviors such as purchase history, browsing history, search patterns, time spent on specific pages, links followed by a user, button clicks, user data from social network platforms. For example, the simple act of a user buying or browsing an item can be viewed as an endorsement of that item. Such forms of feedback are commonly used by online merchants such as Amazon.com [1]. A sample user-item rating matrix, input data instance of a movie

recommendation site (Table 1) can be considered as an example of explicit feedback information. Each cell in Table 1 is the rating value (preference) of a user for a movie on a 5-point scale (i.e. from 1 to 5) and the preferences marked as question mark '?' are the missing values that need to be predicted.

Table 1. A sample user's click and purchase behavior data.

User/Item	Terminator	Deadpool	Mission Impossible	James Bond	Fast & Furious
Alex	2	?	3	?	5
Bob	3	1	5	?	?
Catherine	1	?	?	3	4
David	2	4	1	1	?

Consider a user's click and purchase behavior data as shown in Table 2. This sample user's click and purchase behavior indicates that the customer ended up purchasing few items from the list of clicked items.

Table 2. An example movie site User-item rating matrix.

User Id	Click	Purchase
1	Cheese, Butter, Milk, Cream, Honey, Bread	Cream, Butter, Milk, Honey

Now, an implicit user's transaction (binary) user-item purchase matrix (Table 3) is created by analyzing the list of items purchased by the user and a value of 1 is assigned for the purchased items and 0 represents non purchased items by a user. Analyzing user's implicit preferences (i.e., the behavior pattern data) has been used widely and proved to be useful in practice for constructing input user-item matrix when explicit rating information on items is not available or needs to be made more informative by integrating more learned historical customer purchase behavior.

Table 3. An implicit user-item purchase matrix.

User/item	Milk	Bread	Butter	Cream	Cheese	Honey
User 1	1	0	1	1	0	1

Sequential pattern mining (SPM) discovers interesting subsequences as patterns (Sequential patterns) in a sequence database that can be used later by end users or management to find associations between different items or events in their data for purposes such as marketing campaigns, business reorganization, prediction and planning in the domain of E-commerce. A Sequence database stores a number of records, where all records are sequences $\{s_1, s_2, \dots, s_n\}$ that are arranged with respect to time [4]. A sequence database can be represented as a tuple $\langle \text{SID}, \text{sequence-item sets} \rangle$, where SID: represents the sequence identifier and sequence-item sets specify the sets of items (purchased, watched, etc.) enclosed in parenthesis () in time order (such as every day, week, month) they are purchased by the SID. An example sequence database is retail customer transactions or purchase sequences in a grocery store showing, for each customer, the collection of store items they purchased every week for one month. An example of historical daily purchase data of grocery store is shown in Table 4. It contains CustomerID, PurchasedItems for the set of purchased items by customers, and Timestamp for the time of purchase.

Table 4. Historical purchase data.

CustomerID	PurchasedItems	Timestamp
01	Bread, Milk	10, Sep 2019 00:48:44
02	Bread	11, Sep 2019 10:48:44
01	Bread, Milk, Sugar	15, Sep 2019 10:48:44
02	Sugar, Tea	16, Sep 2019 09:48:44
01	Milk	18, Sep 2019 00:48:44
01	Tea, Sugar	19, Sep 2019 00:48:44

A sequential database can be constructed from such historical purchase data by considering a period of time (day, week, and month). In this case, the purchase sequential database from historical purchase data (Table 4) is presented in Table 5, where SID (01) contains the sequence $\langle (\text{Bread, Milk}), (\text{Bread, Milk, Sugar}), (\text{Milk}), (\text{Tea, Sugar}) \rangle$. This means that customer (01) first purchased Bread and Milk together then purchased Bread, Milk and Sugar together in second purchase and Milk in third purchase, finally, Tea and Sugar together in the last purchase.

Table 5. Sequential database from historical purchase data.

SID	Sequences
01	$\langle (\text{Bread, Milk}), (\text{Bread, Milk, Sugar}), (\text{Milk}), (\text{Tea, Sugar}) \rangle$
02	$\langle (\text{Bread}), (\text{Sugar, Tea}) \rangle$

Sequential patterns are ordered sets of items (events) that are occurring with respect to time [5]. A sequential pattern is denoted in angular brackets $\langle \rangle$, and each itemset contains sets of items, where each itemset enclosed in parenthesis () separated by commas represents a set of items purchased at the same time in one market visit. For example, from Table 5, $\langle (\text{Bread}), (\text{Sugar, Tea}) \rangle$ is a frequent sequential pattern if the minimum support of 75% is used in this database to mine frequent sequential patterns that have occurred up to this minimum support times in the sequential database. This means, that most customers would first purchase Bread, in one visit and also purchase Sugar and Tea together in a subsequent purchase. A Sequential Historical Database (SHOD) algorithm was used in HSPRec system [6] to generate sequential database from historical purchase database similar to Table 4.

The problem of SPM can now be formally described as follows, Given:

- (i) a set of sequential records (called sequences) representing a sequential database, $SDB = \{s_1, s_2, \dots, s_n\}$ with sequence identifiers $1, 2, 3, \dots, n$
- (ii) a minimum support threshold called $\min \sup \zeta$ and
- (iii) a set of k unique candidate items or events $I = \{i_1, i_2, \dots, i_k\}$;

SPM algorithms discover the set of all frequent sub-sequences, S in the given sequence database SDB , of items I at the given $\min \sup \zeta$, that are interesting for the user. A sequence s , is said to be a frequent sequence or a sequential pattern, if its support (the percentage of the total number of database records the sequence appears in), is greater than or equal to the minimum support ($\min \sup \zeta$) [7].

The input of an e-commerce recommendation system, based on collaborative filtering approach, is usually a binary user-item rating matrix (Table 6), only showing whether or not an item has been purchased or liked by a user previously. Thus, the user-item rating matrix can be extremely sparse and with low quality input data (less informative rating data, which does not reflect: (1) how much a user likes a purchased item with value 1; (2) how frequently or how long ago a user purchased an item; (3) what quantity of a product was purchased. One way to improve this input data is to integrate explicit rating with implicit rating drawn from historical purchase or click stream data or to use learning

algorithms such as sequential pattern mining, SPM, of historical purchase and click stream data to extract more informative customer purchase and click stream data behavior, that can be integrated into the user-item rating matrix. This will help to reduce the user-item rating matrix data sparsity and improve the recommendation quality and accuracy. SPM can capture customer purchase behavior over time using mined sequential patterns which is crucial since the time interval between items is useful to learn at what time next item might be purchased. The next purchase decision of a user is often influenced by their recent behaviors and this considers the temporal preference of the user as a sequence of purchased items. An example frequent sequential pattern (FSP) that can be mined from a relevant E-Commerce purchase historical sequential database is $\langle (milk, bread), (milk, cream) \rangle$. This indicates that generally, it is learned from the historical purchase database that whenever customers buy milk and bread together in one week, they come back in the following week to buy milk and cream together.

Table 6. An E-commerce User-item rating matrix.

User Id/Products	Milk	Bread	Butter	Cream	Cheese
User1	1	1	1	?	1
User2	1	1	?	?	?
User3	1	?	?	1	1
User4	1	1	1	?	?

This sequential rule can be written as $(milk, bread) \rightarrow (milk, cream)$. With a sequential rule like this, some of the unknown ratings in the input user-item rating matrix of Table 6 can be filled such that all users who have purchased the antecedent items (milk, bread) have a higher chance of (say 0.5 or some more specific determined chance value) of purchasing also cream next. With this information, the ratings for users 1, 2 and 4 for cream can be changed from unknown to 0.5. In this way a sequential pattern can be used to improve on the quantity of rating values by providing the possible value for the missing/unrated items. A user-item purchase frequency matrix can then be constructed, where each value represents the quantity of a product purchased by a user. This purchase frequency is then normalized to a scaled value (0 to 1) representing how interested a user is in one item as compared to other items to improve rating quality. If these historical sequential purchase patterns of a user are analyzed and integrated into the user-item matrix input, the rating quality (specifying level of interest or value for already rated items) and quantity (finding possible rating for previously unknown ratings) can be enhanced and improved by using the mined sequential patterns. Thus, the recommendation quality can be improved in terms of accuracy, scalability and novelty.

An important task for e-commerce sites is to make predictions about what users might buy in the future, based on user's history of shopping. This problem can be modeled by using one of the most successful methods in the literature which is Collaborative Filtering (CF) technique that makes use of explicit user rating item matrix data from the user for the purpose of recommendation. A major advantage of this model is its ability to capture general taste for recommendation. However, this kind of algorithm has two obvious shortcomings. First, the effectiveness of such algorithms will be greatly reduced when the user's explicit rating behavior data is sparse, the second is these methods ignore the time context of user behavior (how the customer's purchase behavior may vary over time), i.e., they are unable to capture the sequential behavior of users. SPM techniques [7,8], also have been used alone recently to make recommendations more effective by extracting sequential patterns of user purchase behavior because the user's next purchase will be affected by their previous purchases and actions. This recommendation often utilizes user's implicit feedback data and the major advantage of this model is its ability to capture user sequential purchase behavior for recommendations. However, this SPM recommendation model alone cannot capture a user's general taste. It can be seen that both of the methods (CF and SPM) have some shortfall. In fact, both sequential behavior and user's general taste

are important factors that influence user’s purchasing behavior as indicated in [9–11]. This motivates a systematic review on the importance of integrating SPM with CF for recommendation systems, to improve the recommendation quality, through more diverse recommendations, closing the high sparsity matrix problem and thus, making recommendations better by taking into account the user’s general taste and sequential behavior.

The review of these Sequential pattern based collaborative E-commerce recommendation systems involves comparison of their features, such as their recommendation accuracy, user-rating matrix input data sparsity ratio and functionalities (e.g., ability to recommend novel and diverse products, ability to scale up to frequently changing products, or user scalability), recommendation approaches, improving on understanding of the system’s algorithms with example application of system through a clear example, highlighting their strengths, weaknesses and future prospects in the recommendation process. Focus of the survey research in this paper is on indepth understanding of algorithmic methods for collaborative filtering systems based RS that, also enhance recommendation qualilty through sequential pattern mining of historical purchase and click stream data. This work is different from existing surveys or research that have reviewed methods for evaluating recommendation systems by providing a framework with no discussion of any algorithms, such as [12,13]. This survey on more traditional, more technically understandable mining based approaches, is also different from other related surveys or research on complex deep learning based sequential recommendation systems ([11,14–16]), which are still not exploiting historical and click stream purchase data.

1.1. Reasons for Sequential Pattern Mining in E-Commerce Recommendation

1. User-Item Interactions Are Sequentially Dependent: In E-commerce recommendation systems, the crucial task is to identify the next purchase items from customer purchase behaviors [18]. This essentially led to the development of Sequential pattern-based recommendation systems. These systems suggest items that may be of interest to a user by mainly modelling the sequential dependencies over the user-item interactions in a sequence [19], possibly through mining of sequential patterns [6].
2. Improve the quality and quantity of ratings: Recommendation systems in E-commerce suffer from uninformative rating data which usually only represent if a user has purchased a product before. This user-item rating matrix is usually sparse, less informative and leads to poor recommendations [20]. In these systems, even active customers may have purchased only under 1% of the products (1% of 2 million products in an E-Commerce store like Amazon.com is 20,000), i.e., only a few of the total number of items available in a database are often rated by users [21]. Thus, in order to capture more real-life customer purchase behavior and to provide the relationship between already purchased items and recommended items, historical sequential purchase patterns of a user are analyzed and integrated into the user-item matrix input to enhance and improve the rating quality and quantity by providing the possible values for some missing/unrated items. To demonstrate this, consider a historical purchase data (Table 7)).

Table 7. Historical purchase data.

CustomerID	PurchasedItems	Timestamp
User1	Cream, Butter, Milk	2017.06.05.13.38.00
User1	Honey, Butter	2017.06.06.09.40.20
User2	Butter, Cheese	2017.06.05.19.40.16
User 2	Cheese, Honey	2017.06.06.10.40.16

Step 1: Create a user-item purchase frequency matrix (Table 8) from the historical purchase data (Table 7), where the values indicate the number of times an item was purchased by a user. For example, User 1 purchased butter twice, Honey once and so on.

Table 8. User-item frequency matrix from historical purchase data.

User/item	Milk	Bread	Butter	Cream	Cheese	Honey
User 1	1	0	1	1	0	1
User 2	?	?	1	?	2	1

Step 2: Now, convert historical purchase data (Table 7) to a sequential database (Table 9) by considering the period of time (day, week, and month) of the purchase.

Table 9. Purchase sequential database from historical purchase data.

SID	Purchase Sequences
1	< (Cream, Butter, Milk), (Honey, Butter) >
2	< (Butter, Cheese), (Cheese, Honey) >

Step 3: Create frequent sequential purchase patterns from the sequential database (Table 9) using any SPM algorithm like GSP [5] and the possible purchase sequential rules (Table 10) from frequent purchase sequences are extracted. Using these sequential purchase rules, some of the unknown ratings in user-item purchase frequency matrix (e.g., value of User1 for item cheese in Table 8) can be filled by using a predicted value such that all users who have purchased the antecedent items like (milk, butter) from Rule No:1 of Table 10 have a higher chance of (say 0.5 or some more specific determined chance value for the highly probable purchases determined by the SPs) purchasing also cheese next. Hence, using Rule No:1 it, can be inferred that as the user1 purchased milk and butter in this transaction, there are high chances that he would even purchase cheese in the same transaction. Hence, assign a value of 0.5 to the user-item combination (User1-Cheese). Similarly, (User2-Cream) is filled using Rule No:3 and (User2-Milk) is filled using Rule No:2.

Table 10. Sequential rules created from n-frequent sequences.

Rule No	Sequential rule
1	Milk, Butter → Cheese
2	Cream, Cheese → Milk
3	Cheese, Honey → Cream

Step 4: The final enriched user-item frequency matrix created with help of sequential rules as described above is shown in Table 11.

Table 11. Rich user-item frequency matrix with sequential rule.

User/item	Milk	Bread	Butter	Cream	Cheese	Honey
User 1	1	?	2	1	0.5	1
User 2	0.5	?	1	0.5	2	1

In this way, the historical sequential purchase patterns of a user are analyzed and integrated into the user-item matrix input to enhance and improve the rating quality and quantity.

1.2. Outline of the Paper

The article is organized as follows. Section 2 reviews existing algorithms and presents surveys of sequential pattern-based E-commerce recommendation systems with examples. Section 3 gives the proposed classification of techniques with comparative performance analysis of the reviewed algorithms, along with discussions of the features used in the classification of the algorithms. Conclusions and Future Work are given in Section 4.

2. Existing Sequential Pattern-Based E-Commerce Recommendation Systems

The main aim of e-commerce websites is to turn their visitors into customers. As the transaction data provides sets of preferred items and can be used to predict future customer preferences, some researchers applied association rule mining technique to extract sequences to improve performance of recommendation systems [17,22]. However, such systems incorporate customer transaction data from only a single temporal period, which omits the dynamic nature of a customer's access sequences. Unlike association rules, sequential patterns [8] may suggest that a user who accesses a new item in the current time period is likely to access another item in the next time period. Thus, SPM techniques have been used for extracting complex sequential patterns of user purchase behavior and if these patterns are learned and included in the user-item matrix input, accuracy of the recommendation system will be improved as the input becomes more informative before it is fed to CF. Thus, integrating CF and SPM of historical purchase data will improve the recommendation quality, reduce the data sparsity and increase the novelty of recommendations.

Existing E-commerce recommendation systems that can be found in the literature, which have combined CF with some form of historical purchase sequences (SPM) to recommend items to users are these ten systems referred to as (1) Model Based Approach, ChoRec05, [23], (2) Pattern Segmentation Framework, ChoRec09, [24], (3) Sequential pattern based collaborative recommender system, HuaRec09, [25], (4) Segmentation based approach, LiuRec09, [26], (5) Hybrid Online Product recommendation, ChoiRec12, [27], (6) Hybrid Model HM, RecSys16, [28], (7) Product Recommendation System PRS, RecSys16 [29], (8) Sequential pattern based recommender system, SainiRec17, [30], (9) Historical purchase and clickstream-based recommendation, HPCRec18, [31], and (10) Historical Sequential Pattern Recommendation, HSPCRec19, [6]. A brief overview of these systems is provided next.

2.1. Model Based Approach: ChoRec05 [23]

A hybrid recommendation system that combines Self-Organizing Map (SOM) clustering technique and Association rule based sequential cluster rules was proposed for mining the changes in customer buying behavior over time in [23]. The recommendation procedure is divided into two components called a model building phase and a recommendation phase. Example of ChoRec05

Input: Historical purchase data in ecommerce dataset including customer-Id, purchased items, duration of transaction.

Output: Recommended products to each user.

Algorithm:

Model-building phase: A model-building phase is performed once to create a reliable model from the customer transaction database that includes: Transaction clustering, where the transactions are transformed into an input matrix composed of a bit vector and these time-ordered vectors for a customer represents the purchase history of the customer and this input matrix can be thought of as the dynamic profile of the customer.

Identification of cluster sequences: The cluster sequence of a customer is learned by identifying the cluster to which each transaction of the customer belongs, during each time period.

Table 12. Behavior loci of customers.

CID	T-2	T-1	T
001	10	3	9
002	10	1	3
003	3	10	4
004	10	-	9
005	1	-	10
006	4	-	3
007	-	9	3
008	3	-	1
009	-	4	-

Extraction of sequential cluster rules: To mine customer behavior according to purchase time, association rule [32], R_i is adopted for determining the most frequent pattern with confidence.

$R_j = r_j, T-1+1, \dots, r_j, T-1 \rightarrow r_j, T$ (support, confidence).

where rule R_j indicates that, if the locus of a customer is $r_j, T-1 \rightarrow r_j, T$, then, the behavior cluster for the customer is r_j, T at time T.

Table 13. Derived Association Rules.

Rules	T-2	T-1	T	Support	Confidence
1	10	-	9	0.3	0.667
2	3	10	4	0.1	1.0
3	10	3	9	0.1	1.0
4	10	1	3	0.1	1.0
5	1	-	10	0.1	1.0
6	-	10	4	0.1	1.0
7	3	-	4	0.2	0.5
8	3	-	1	0.2	0.5
9	-	3	9	0.1	1.0

Recommendation phase: In this phase, given the target customers, the products that are best matched to the dynamic behaviors of these customers are found and the target customer's transactions are converted into behavior locus using the SOM clustering model, as in the previous phase. Finally, the best-matching loci stored in the association rule base are extracted and the top N items are recommended to the target customer, i.e., the most frequently purchased products from among the products in the cluster. *Number in parenthesis denotes purchase quantity.

Table 14. A product list purchased by other target customers in selected cluster.

Customer	Purchased Products (Brand)
CID 001	Brand 31 (2), 33 (3)
CID 002	Brand 31 (2), 37 (2)
CID 003	Brand 33 (2), 38 (3)

2.2. Pattern Segmentation Framework: ChenRec09 [24]

Chen et al. [24] proposed a sequential pattern-based recommender system that incorporates RFM (Recency, Frequency and Monetary) concept, where “Recency” represents the length of a time period since the last purchase; a lower value corresponds to a higher probability that the customer will make repeat purchases. “Frequency” denotes the number of purchases within a specified time period; a higher frequency indicates stronger customer loyalty. “Monetary” means the amount of money spent in this specified time period; if a customer has a higher monetary value, the company should focus more resources on retaining that customer. RFM sequential patterns are then defined and a novel algorithm, named RFM-Apriori is developed, for generating all RFM sequential patterns from customer’s purchase data. The algorithm was developed by modifying the well-known Apriori (GSP) algorithm [5] and consists of iterative phases.

RFM-Apriori Algorithm:

Candidate generation: First, the algorithm places all itemsets into C_1 , the set of candidate F patterns with length 1, and then scans the database to find the frequent 1-patterns (L_1). An itemset is used as a unit to expand the patterns, rather than just an Item as it can reduce the number of phases needed to complete the algorithm, thus improving efficiency. Second, suppose the set of frequent (k-1)-patterns, L_{k-1} is already known, it is joined with itself Apriori-join way to generate candidate RF patterns of length k, where $k \geq 2$, if they have the same (k-2)-postfix. The algorithm then scans the database to determine the supports of the patterns in C_k , and then finds L_k by removing those patterns from C_k with supports lower than the minimum support. This phase is repeated by increasing k by one, until no more patterns can be generated.

Counting supports by traversing an inverse candidate tree: To count supports, an inverse candidate tree is used to store all candidate patterns in C_{Ik} , where a leaf node corresponds to a candidate pattern. Using every data sequence to traverse the tree, support values can be accumulated in each leaf node. This is an efficient method of determining whether a candidate pattern satisfies the recency constraint. This traversal procedure is a recursive program by which all subsequences in T can be matched with all candidate patterns in C_{Ik} . If a matched subsequence can be found that satisfies both the recency and monetary constraints for a pattern (leaf node), the rfm-support and rf-support of this pattern is increased by one. If it satisfies only the recency constraint, however, only the rf-support is increased by one. Using RFM-Apriori algorithm, a pattern segmentation framework was proposed, which allows to partition the RFM-patterns into segments relevant to the RFM criteria, to generate valuable information on customer purchasing behavior for managerial decision-making. By partitioning the patterns into groups based on the RFM indices, a retailer can further compare, contrast, and aggregate these groups of patterns to find possible changes in purchasing patterns over time.

2.3. Sequential pattern based collaborative recommender system: HuaRec09 [25]

Huang et al. (2009) proposed a hybrid recommendation system which is a sequential pattern based collaborative recommender system that predicts the customer’s time-variant purchase behavior in an e-commerce environment where the customer’s purchase patterns may change gradually. A two-stage recommendation process is developed to predict customer purchase behavior for the product categories, as well as for product items. The time window weight is introduced to provide higher importance on the sequential patterns closer to the current time period that possess a larger impact on the prediction than patterns relatively far from the current time period. Given all the target customer’s transactional sequences in the current time period T and the previous number r periods, $T - 1, T - 2, \dots, T - r$. This study determines the active customers most likely to purchase items in the next time period T + 1 (target prediction period). The proposed system consists of model training for the target customers and model use (implementation) for the active customers. Active customers are selected from the target customer to receive recommendations during model use. The steps in each

of these modules are discussed below.

Model training for the target customers consists of:

1. Identifying the target customers: The target customers can be identified according to customer behavioral variables such as recency, frequency and monetary expenditure (RFM model) [33].
2. Building dynamic customer profile: Dynamic customer buying behaviors can be modeled by analyzing the customer's periodic transaction data.
3. Clustering the customers: The customers are clustered based on their dynamic customer profiles using the genetic algorithm GA-based clustering approach.
4. Sequential pattern mining for each cluster: A cluster's sequential patterns represent the buying behavior of the customers in that cluster. The proposed sequential pattern-based prediction on the product categories involves generating the customer purchase sequence for each customer and discovering the sequential patterns for each cluster using any SPM algorithm like GSP [5], PrefixSpan [34].

Model use for the active customer

A two-stage recommendation process is followed by the cluster selection for the active customer which includes predicting the top-M product categories and recommending the top-N product items. The top-M product categories are predicted based on the value of product Category Recommendation Score (CRS). The CRS for the predicted $category_i$ is calculated as follows: $CRS^{category_i} = \sum_{period_t} (CATEGORY - SUPPORT_{period_t}^{category_i} \times WEIGHT_{period_t})$ for $T = T_0, T_1, \dots, T_r$ where $WEIGHT_{period_t}$ is the time window weight in $period_t$.

Top-N product items recommendation: The possible top-N items that the active customer will probably purchase in the target period are generated by calculating the recommendation score for each item in the top-M product categories. The Item Recommendation Score (IRS) for an item among the top-M product categories is calculated as follows. $IRS^{item_j} = \sum_{period_t} (Purchase - Frequency_{period_t}^{item_j} \cdot Weight_{period_t})$ for $T = T_0, T_1, \dots, T_r$

where $Weight_{period_t}$ is the time window weight in $period_t$. where, $Purchase - Frequency_{period_t}^{item_j}$ is the frequency of $item_j$ bought by all customers in the same cluster in $period_t$. The Purchase-Frequency is defined as the number of times instead of quantity that customers brought during a certain period. The top-N items with larger recommendation scores, excluding items that have been bought by the active customer before are recommended to the active customer.

2.4. Segmentation based approach - LiuRec09 [18]

A hybrid recommendation system which combines segmentation-based sequential rule method with the segmentation-based KNN-CF method, was proposed in [18].

Example of LiuRec09

Assume an E-commerce historical purchase data containing purchase items, with frequency of purchase, price and transaction time as input.

Segmentation-based Sequential Rule (SSR) method

Step 1: Customer clustering: Cluster the customers into distinct groups based on their RFM values (Recency, Frequency, and Monetary). The RFM patterns of each cluster are identified by assigning \uparrow or \downarrow ; according to whether the RFM value of a cluster is larger than or smaller than the overall average RFM value.

Clusters with the same pattern are combined into one cluster. For example, clusters 3, 4 and 5 in Table 15 have the same pattern, similarly, clusters 2, 7 and 8 can also be merged. Therefore, eight customer clusters can be reduced to four customer segments - loyal, potential, uncertain, and valueless based on their RFM patterns as shown in Table 16.

Table 15. K-means clusters based on the normalized RFM values.

	No. of Customers	R (Recency)	F (Frequency)	M (Monetary)	Patterns
Cluster 1	104	72.260	19.587	40797.23	R↑, F↓, M↑
Cluster 2	43	119.558	3.791	7342.326	R↑, F↓, M↓
Cluster 3	17	64.294	67.2351	147315.6	R↓, F↑, M↑
Cluster 4	214	56.696	19.832	40279.53	R↓, F↑, M↑
Cluster 5	78	57.192	37.846	74045.92	R↓, F↑, M↑
Cluster 6	367	58.335	9.632	18677.27	R↓, F↓, M↓
Cluster 7	126	92.246	7.286	14853.89	R↑, F↓, M↓
Cluster 8	240	73.892	8.496	16109.99	R↑, F↓, M↓
Average		68.216	14.324	28638.3	

Table 16. Four customer segments from combining clusters with similar RFM patterns.

Customer Segment	No. of Customers	R (Recency)	F (Frequency)	M (Monetary)
Loyal	309	R↓ (57.239)	F↑ (26.987)	M↑ (54691.80)
Potential	104	R↑ (72.260)	F↑ (19.587)	M↑ (40797.23)
Uncertain	367	R↓ (58.335)	F↓ (9.632)	M↓ (18677.26)
Valueless	409	R↑ (84.347)	F↓ (7.628)	M↓ (14801.23)

Step 2: Transaction clustering: Transactions are divided into groups (transaction clusters) based on similar product items and buying patterns. Customer's transaction clusters are used to identify the sequence of transaction clusters over time. A sample change of customers' transactions in three periods are displayed in Table 17.

Table 17. Change in customer buying behavior.

	Period 1	Period 2	Period 3
Customer 1		AB	E
Customer 2	B		D
Customer 3		A	E

Step 3: Mining customer behavior from transaction clusters: To mine customer behavior according to purchase time, association rule [32] is adopted for determining the most frequent pattern with confidence. From Table 17, we can extract a sequential rule $Ap2 \rightarrow Ep3$ (0.4,1) with support of 40 percent and confidence of 100 percent. According to this rule, if a customer's purchase behavior in period P2 is in transaction cluster A, then his/her behavior in P3 will be in transaction cluster E. The other sequential rules $Bp2 \rightarrow Ep3$ (0.2,1) and $Bp1 \rightarrow Dp3$ (0.2,1) can be obtained similarly.

Step 4: The determination and match of the cluster sequences of target customers: The degree of match between a target customer's buying behavior and a sequential rule is calculated by a fitness measure.

Step 5: Recommendation: Finally, the frequency count of each item in predicted transaction cluster is calculated and the top N items with highest frequency count are returned.

Segmentation-based KNN-CF method (SKCF): In this step, for each customer, Pearson's correlation coefficient is used to measure the similarity between target customer and other customers in the same

segment and the k most similar (highest ranked) customers are selected as the k-nearest neighbors of the target customer. Then, the N most frequent products not yet purchased by the target customer u (in period T) are selected as the Top-N recommendations.

Hybrid recommendation method

SSR and SKCF are combined linearly with a weighted combination, as shown below, where α and $(1 - \alpha)$ are the weights of SKCF and SSR methods respectively. The product items with the Top-N values of the resulting linear combination of the two methods are selected for recommendation.

Product Rating = $(1 - \alpha) * SequentialRule + \alpha * CollaborativeFiltering$

2.5. Hybrid Online Product recommendation: ChoiRec12 [27]

Choi, Yoo, Kim and Suh (2012) proposed a hybrid recommendation system that uses a combination of CF and SPM. This system extracts implicit ratings based on purchase history by using the number of times user u purchased item i with respect to total transactions, which can be used in CF even when the explicit rating is not available.

Example of ChoiRec12: Consider the fragment of historical purchase data given in Table 18, where only purchase time is provided as available information, and the goal is to recommend suitable items to user T.

Table 18. [27] historical user-item matrix.

	Item 1 Date	Item 2 Date	Item 3 Date	Item 4 Date	Item 5 Date
User 1	01/01	-	01/02	01/03	-
User 2	01/01	-	01/02	01/03	01/04
User 3	-	01/01	01/02	-	01/03
User 4	01/01	01/02	01/03	-	-

Step 1: Deriving implicit ratings from user transactions: The implicit rating can be computed based on purchase history by using the number of times user u purchased item i with respect to total transactions. For example, user 1 purchased item 1 one time out of three transactions. In the same way, consider a user-item implicit rating matrix created from the historical data as given in Table 19.

Table 19. Implicit rating derived from user's transactions.

	Item 1	Item 2	Item 3	Item 4	Item 5	Mean Rating
User 1	3	?	1	5	?	3
User 2	4	?	3	1	2	2.5
User 3	?	1	2	?	4	2.3
User 4	5	4	3	?	?	4
User T	?	4	3	2	?	3

Step 2: Calculating mean rating and user similarity based on the implicit rating: The mean rating is computed by adding all the rating of users on items with respect to total numbers of ratings. So, Mean of rating for User 1 = $(3+1+5)/3 = 3$, User 2 = 2.5, User 3 = 2.3, User 4 = 4 and User T = 3. Compute similarities between users using Cosine similarity, which is given as:

$Cosine(T, b) = \sum_{i=1}^m (R_{T,i}) \cdot (R_{b,i}) / \sqrt{\sum_{i=1}^m (R_{T,i})^2} \cdot \sqrt{\sum_{i=1}^m (R_{b,i})^2}$ where $(R_{T,i})$ denotes the ratings of users T on item i similarly $R_{b,i}$ denotes the rating of user b on item i. For example, the calculated similarities between target user T and every other user will be: $CS(T,1)=0.7071$, $CS(T,2)=0.9648$,

$CS(T,3)=0.8944$, $CS(T,4) =1$, where $CS(T,1)$ means Cosine Similarity between target user T and user 1 and so on.

Step 3: Finding Top k nearest neighbors of target user T: This is done by sorting the user’s similarities in descending order and then selecting the top k (where $k=2$) neighbors. So, the sorted similarities in descending order will be $CS(T,4) = 1$, $CS(T,2) = 0.9648$, $CS(T,3) = 0.8944$, $CS(T,1) = 0.7071$. The top 2 neighbors for target user T will be User 4 and User 2.

Step 4: Calculating the CF-based predicted preference (CFPP): The rating information of the top k neighbors is then used to predict CF-based predicted preference of user a on itemi. For example, the CFPP of a target user T on all other items will now be $CFPP(T, item1) = 4.7455$, $CFPP(T, item2) = 3.5$, $CFPP(T, item3) =3.2365$, $CFPP(T, item4) = 2$ and $CFPP(T,5) = 3$.

Step 5: Deriving sequential patterns and computing purchase item-based score (SPAPP): Generate sequence data of each user by sorting transaction data for the person according to the transaction date. Then, find frequent items using the process of candidate generation (C_k) and pruning (L_k) until the candidate set is empty. Now, match subsequences of a target user purchase with derived purchased items by enumerating target user purchase item. Finally, calculate the pattern analysis based predicted preference (SPAPP) of user T on item i. For example, SPAPP of target user on item 1 is $SPAPP (T,1) = 0$, similarly, $SPAPP (T,2) = 0$, $SPAPP (T,3) = 0.75+0.5+0.5=1.25$, $SPAPP (T,4) = 0.5+0.5+0.5=1.5$, $SPAPP (T,5) = 0.5$.

Step 6: Integrate CFPP and SPAPP: CFPP and SPAPP are normalized to get N_CFPP and N_SPAPP , respectively. Target user T’s final predicted preference on item i, $FPP (T,i)$, is calculated as α times CFPP plus $1 - \alpha$ times SPAPP, where α and $1 - \alpha$ are weights given to CF and SPA and are set to 0.1 and 0.9 respectively. The FPP values are as shown in Table 20.

Table 20. Integrating CFPP and SPAPP.

	CFPP	SPAPP	N_CFPP	N_SPAPP	FPP
Item 1	4.7455	0.7071	1	0	0.5
Item 2	3.5	0.9648	0.5463	0	0.273
Item 3	3.2365	0.8944	0.4504	0.8333	0.6419
Item 4	2	1	0	1	0.5
Item 5	3 0.333	0.3642	0.3333	0.3488	

Step 7: Recommend the item having highest rank: After obtaining FPP values of items purchased by neighbors of the target user, the item having highest FPP is recommended to target user T. In the case from Table 20, to recommend two items, then item3 and item4 will be recommended because they have the highest FPP values.

2.6. Hybrid Model - HM RecSys16, [28]

A hybrid recommender system that combines the prefix span algorithm with traditional matrix factorization was proposed in [28]. SPM aims to find frequent sequential patterns in sequence database and is applied in this hybrid model to predict customer’s payment behavior thus contributing to the accuracy of the model. The workflow of the system consists of three phases: Behavior Prediction Phase, CF Phase and Recommend Phase.

Purchasing Pattern’s Extraction

BPM (Behavior pattern model) utilizes the Prefix-span algorithm to extract the most prevailing purchasing sequences from the warehouse in real time and match the sequences with customer’s behavior pattern who is browsing or adding an item to cart. When the recommender system’s behavior monitoring part detects the user’s potential purchasing tendencies, the system will fetch the user’s

historical behavior record from sequence database and build an item-user rating matrix and each entry contains historical behavior of the Ith user to Jth product.

Table 21. Fang’s Item-user rating matrix.

Item_Id/ User_id	562	529	267	858	241
10001569		2	4		1
100022999	1	1	2		
10000003		1			
100009489	3		2	1	3
100018271		1			
100020308			3		

Matrix Factorization-based Collaborative Filtering CF method is used to find a set of customers whose purchased and rated items overlap the user’s purchased and rated items. The algorithm generates recommendations based on a few customers who are most similar to the user and generates preference tendencies of the users based on their historical purchasing record. The basic matrix factorization model is used which factorizes the user-item matrix into two matrices where one represents features of the products and another represents the preferences of users. Multiplying the two matrices, gives back predictions about user’s preference to all products.

$$r_{ui} = q_i^T * p_u$$

The r_{ui} represents user u ’s rating of item i , and latent factor model is used to learn the factor vectors p_u and q_i by minimizing the regularized squared error on the set of known ratings.

$\sum_{(u,i) \in k} (r_{ui} - q_i^T \cdot p_u)^2 + \lambda (\|q_i\|^2 + \|p_u\|^2)$ Recommendation Phase The payment behavior patterns extracted from the behavior prediction phase and the preference collected from CF method are combined to select target items as suggestions. In the first step, the customer’s real-time behavior sequences are generated and stored in database called candidate database. The candidate database will be scanned at a regular interval and sequence contains payment patterns will be sent to recommender system as potential purchasing sequence. Secondly, for those potential buyers, generate the preference information from CF phase which represents the preference degree towards each product. Since sequential mining phase will not only generate the payment sequence, but also the category of the target item, the category matched items in preference vector to recommend, will be chosen.

2.7. Product Recommendation System: PRS RecSys16, [29]

Jamali & Navaei (2016) proposed a two-level product hybrid recommendation system which combines C-Means clustering algorithm and Freespan algorithm. At first, the available products are clustered by using C-Means algorithm to create groups of products with similar characteristics. Then, the second level considers the customer’s behavior and their purchase history for drawing the relationships between products by using Sequential Pattern Analysis (SPA) method. These relationships, eventually, will lead to appropriate recommendation for customers and also increases the likelihood of selling related products in electronic transactions.

Their PRS (Product Recommendation System) includes two levels of product recommendation: first level is recommended before product purchase and the other one, after purchasing. PRS initially collects product’s data from electronic store, separates the products according to their type and are then clustered based on their numerical attributes in three separate clusters of high, medium and low quality by C-means algorithm. Here, C-Means clustering algorithm is used to separate products by their types and create groups with similar features and thereby classify products. The algorithm generates clusters based on fuzzy logic and does not consider sharp boundaries between the clusters, thus allowing each

feature vector to belong to different clusters by a certain degree. The degree of membership of a feature vector to a cluster is usually considered as a function of its distance from the cluster centroid points. It is based on minimization of the following objective function: $J_m = \sum_{i=1}^N \sum_{j=1}^C u_{ij}^m \|X_i - c_j\|^2$, $1 \leq m < \infty$, where m is any real number greater than 1, u_{ij} is the degree of membership of x_i in the cluster j , x_i is the i th of d -dimensional measured data, c_j is the d -dimension center of the cluster, and $\| * \|$ is any norm expressing the similarity between any measured data and the center.

Next, the PRS tries to identify customer's requirements and criteria using an online form that takes information about product such as type, quality, price, brand, etc. Thus, this information is used to assign an appropriate cluster to the customer. In the second level, information about history of customer's shopping behavior is collected. This information is used to explore relations between products by Freespan algorithm of SPA method. Freespan mines sequential patterns by partitioning the search space and projecting the sequence sub-databases recursively based on the projected itemsets [35]. Eventually, these relations and patterns will be provided as product recommendations, as it recommends associated products to the products purchased, since the relationships between the products will increase the likelihood of buying the products together and this makes the customer aware of potentially related products.

2.8. Sequential Pattern-based Recommender System: SainiRec17 [30]

Saini et al. (2017) tried to find the sequence of all items which were bought regularly, that is, not only finding the same product purchased every month, but, also the different products purchased one after another in a sequence. Users buy some products in a sequence, for example, most of the users buy a mobile phone and mobile cover in a sequence. So, the authors tried to find out such kind of sequences, in online shopping. Thus, the main objective of this article is to find the sequences frequent among all users and Intra-duration in the sequence in an online product purchasing system. With the help of SPADE [36] algorithm, frequent sequential purchase patterns were found and in the next step, sequence mining algorithm was applied to find the sequences available in the dataset. Finally, the time that elapsed between the purchase of first product and next sequential product was calculated by finding the mean and mode of the duration followed by all users. Here, mean gives the average time gap between products, whereas, mode gives the duration followed by most of the users.

2.9. Historical Clickstream-based Recommendation: HPCRec18 [31]

A novel recommendation system called Historical Purchase with Clickstream recommendation system (HPCRec) was proposed which integrates purchase frequencies and consequential bond relationship between clicks and purchases. The term consequential bond was introduced in this HPCRec system and is originated from the concept that customer who clicks on some items will ultimately purchase an item from a list of clicks in most of the cases. By processing this information, it enhances the user-item rating matrix in both quantity and quality aspects and then improves recommendations. The quality of ratings was improved by capturing the level of interest in a product already purchased by a user before, through record of normalized frequency of purchase using the unit vector method. The quantity of ratings was improved with consequential bond between clicks and purchases, for the sessions without purchases. Finally, the ratings for all the original unknowns are predicted based on this enriched rating matrix using CF algorithm. HPCRec system can provide recommendations for infrequent users and it proves that the consequential bond with the normalized frequencies are more effective at predicting user interest.

Algorithm: Input to HPCRec system are 1) Consequential table (Table 23) which shows the relationship between user clicks and purchases and 2) User item purchase frequency matrix (Table 24) which represents the frequency of a product purchased from user item rating matrix (Table 22). The algorithm is demonstrated below:

Step 1: Normalize purchase frequency matrix using unit vector formula: Form user-item purchase frequency matrix (Table 24) from Table 23, where value represents the number of times product

purchased by a user. Normalize purchase frequency to a scaled value (0 to 1) to form Normalized user-item purchase frequency matrix (Table 25) using unit vector formula below:

$$\text{Normalized } r_{ui} = r_{ui} / \sqrt{r_{u1}^2 + r_{u2}^2 + \dots + r_{un}^2}$$

Table 22. User-item rating matrix.

Customer/ Item	1	2	3	4
1	?	1	1	?
2	1	1	?	1
3	1	?	?	?

Table 23. Consequential table.

Session Id	User Id	Clicks	Purchase
1	1	1, 2	2
2	1	3, 5, 2, 3	2, 3
3	2	2, 1, 4	1, 2, 4
4	2	4, 4, 1, 2	2, 4, 4
5	3	1, 2, 1	1
6	3	3, 5, 2	

For example, if user 2 purchases are item1: 1, item2: 2, item3: 0, item4: 3, then normalized purchase frequency for user 2 on item 2 is $2 / \sqrt{1^2 + 2^2 + 0^2 + 3^2} = 0.53$.

Table 24. U-I purchase frequency.

Customer/ Item	1	2	3	4
1	?	2	1	?
2	1	2	?	3
3	1	?	?	?

Table 25. Normalized U-I purchase freq matrix.

Customer/ Item	1	2	3	4
1	?	0.89	0.45	?
2	0.27	0.53	?	0.8
3	1	?	?	?

$LCSR(x, y) = LCS(x, y) / \max(|x|, |y|)$ $LCS(x, y)$ is longest common subsequence between sequence x and sequence y and is computed by:

$LCS(X_i, Y_i) = \emptyset$ if $i=0$ or $j=0$; but $LCS(X_i, Y_i) = LCS(X_{i-1}, Y_{j-1}) \cup X_i$ if $X_i = Y_j$; but $LCS(X_i, Y_i) = \text{longest}(LCS(X_i, Y_{j-1}), LCS(X_{i-1}, Y_j))$ if $X_i \neq Y_j$, where $\max(|x|, |y|)$ is the maximum length of two sequence.

Step 2: Compute clickstream sequence similarity measurement (CSSM): For each session without a

purchase in consequential table, compute clickstream sequence similarity measurement (CSSM) to find similar sessions with purchase value using longest common subsequence rate (LCSR). For example, $LCSR(<3,5,2>, <3,5,2,3>) = \frac{(<3,5,2> \cap <3,5,2,3>)}{\max(3,4)} = 3/4 = 0.75$. As there is no purchase information of session 6 in consequential table (Table 23), compute Clickstream similarity between session 6 which is $<3,5,2>$ and other sessions and is as shown below:

Table 26. CSSM Info table.

CSSM	Info	table
1, 2	2	0.37
3, 5, 2, 3	2, 3	0.845
2, 1, 4	1, 2, 4	0.33
4, 4, 1, 2	2, 4, 4	0.245
1, 2, 1	1	0.295

Step 3: Form a weighted transaction table using the similarity as weight and purchases as transaction records.

Table 27. Weighted transactional purchase table.

Purchase	< 2 >	< 2, 3 >	< 1, 2, 4 >	< 2, 4, 4 >	< 1 >
1	0.37	0.845	0.33	0.245	0.295

Step 4: Call TWFI (Transaction-based Weighted Frequent Item) function, which takes a weighted transaction table, where weights are assigned to each transaction as input and returns items with weighted support in a given threshold. For example, let's consider minimum weighted support=0.1, then, we will have frequent weighted transaction table as shown in Table 28.

Table 28. Weighted frequent transactional purchase table.

Purchase (Transaction Records)	2	2, 3	1, 2, 4	2, 4, 4	1
Weight	0.37	0.845	0.33	0.245	0.295

Step 5: Calculate support to form a distinct item from set of all the transactions.

Table 29. Support for item in weighted frequent table.

Item	1	2	3	4
Support	2	4	1	3

Step 6: Compute the average weighted support for each item using AWS as AW multiplied by support, where AW is the sum of the item weight divided by support. For example, $AWS(1) = 0.33 + 0.295 = 0.625$, $AWS(4) = 0.33 + 0.245 + 0.245 = 0.82$.

Table 30. Weight for items in purchase pattern table.

Item	1	2	3	4
AWS	0.625	1.79	0.845	0.82

Step 7: Normalize weighted support using feature scaling $x' = (x - \min) / (\max - \min)$ So, for the average weighted support, $\max = 1.79$, $\min = 0.625$, the new average weighted support for

item3 is (0.845 minus 0.625) divided by (1.79 minus 0.625) = 0.189. All the weighted supports are $< 1 : 0, 2 : 1, 3 : 0.189, 4 : 0.167 >$.

Step 8: Return all the items that have a normalized weighted support greater than or equal to minimum weighted support (e.g., (2:1),(3:0.189),(4:0.167)). Then for each one of these items, if user has not purchased it, add the weight into the normalized user-item matrix.

Step 9: Return to step 2 if there are more sessions without a purchase, otherwise, run the CF algorithm using the updated rating matrix to get predicted ratings for all of the original unknowns as demonstrated in Table 31.

Table 31. User-item rating matrix with predicted ratings.

	Item 1	Item 2	Item 3	Item 4
User 1	0.63	0.89	0.45	0.49
User 2	0.27	0.53	0.35	0.8
User 3	1	0.74	0.27	0.33

2.10. Historical Sequential Pattern Recommendation: HSPRec19, [6]

This work was proposed to improve the HPCRec system which did not integrate frequent sequential patterns to capture more real-life customer sequence patterns of purchase behavior inside consequential bond. Thus, the authors proposed an algorithm called HSPRec (Historical Sequential Pattern Recommendation System), which explored enriching the user-item matrix with sequential pattern of customer clicks and purchases to capture better customer behavior.

Example of HSPRec

Input: Minimum Support, Historical user-item purchase frequency matrix and consequential bond

Output: An enriched user-item matrix for CF

Consider the consequential bond of clicks and purchases (Table 32) created from click and purchase historical data and daily sequential database (Table 33) created from historical transaction data by considering the period of time (day, week, and month).

Table 32. Consequential table from click and purchase historical data.

User Id	Click	Purchase
1	Cheese, Butter, Milk, Butter, Cream, Cheese, Honey, Cream, Butter	Cream, Butter, Milk, Honey, Butter
2	Cheese, Cream, Honey, Butter	Butter, Cheese, Cheese, Honey
3	Cheese, Milk	?

Table 33. Daily sequential database.

SID	Click Sequence	Purchase Sequence
1	<(Cheese, Butter, Milk, Butter, Cream, Cheese), (Honey, Cream, Butter)>	<(Cream, Butter, Milk), (Honey, Butter)>
2	<(Cheese, Cream, Honey, Butter)>	<(Butter, Cheese), (Cheese, Honey)>
3	<(Cheese, Milk)>	?

Algorithm: Step 1: Create a user-item purchase frequency matrix (Table 34) from Table 32, where the number indicates, the number of times item purchased by a user. For example, User 1 purchased butter twice, Honey once and so on.

Table 34. A User-item frequency matrix.

User/ item	Milk	Bread	Butter	Cream	Cheese	Honey
User 1	1	?	2	1	?	1
User 2	?	?	1	?	2	1
User 3	?	?	?	?	?	?

Step 2: Create frequent sequential purchase patterns from daily sequential database (Table 33) using GSP algorithm. In this case, the possible purchase sequential rules from frequent purchase sequences are:

Table 35. Sequential rules from n-frequent sequences.

Rule No	Sequential rule
1	<i>Milk, Butter → Cheese</i>
2	<i>Cream, Cheese → Milk</i>
3	<i>Cheese, Honey → Cream</i>
4	<i>Honey → Cream</i>
5	<i>Honey → Milk</i>

Step 3: Fill purchase information in user-item frequency matrix using sequential purchase rules.

Table 36. Rich user-item frequency matrix with sequential rules.

User/ item	Milk	Bread	Butter	Cream	Cheese	Honey
User 1	1	?	2	1	1	1
User 2	?	?	1	1	2	1
User 3	?	?	?	?	?	?

Step 4: As it can be seen in Table 33, that there is no purchase information for user 3, to find purchase information for user 3, analyze the relationship between click and purchase considering their sequence and recommend item from the click sequential rule, where the user clicks but does not purchase anything. Step 5: Compute Click Purchase Pattern (CPS) similarity using frequency and sequence of click and purchase patterns. If there is no purchase along with click item, then use the recommended item. Step 6: Assign Click Purchase (CPS) similarity value to the purchase patterns present in the consequential bond. Step 7: Assign weighted purchase patterns to Weighted Frequent Purchase Pattern Miner (WFPP) and compute a weight for item present in weighted purchase pattern using the equation: $R_{item_i} = \sum_{i=1}^n CPS_{item_i} / Support_{item_i}$ Step 8: the weight of item to make user-item matrix rich. The computed rich user-item purchase frequency matrix is shown in Table 37.

Table 37. Rich user-item purchase frequency matrix.

User/ item	Milk	Bread	Butter	Cream	Cheese	Honey
User 1	1	?	2	1	1	1
User 2	?	?	1	1	2	1
User 3	0.63	?	0.61	0.63	0.56	0.59

Step 9: Normalize rich user-item purchase frequency matrix to get normalized quantitatively rich user-item matrix (Table 35) using unit normalization function given below

$$\text{Normalized } r_{ui} = r_{ui} / \sqrt{r_{ui_1}^2 + r_{ui_2}^2 + \dots + r_{ui_n}^2}$$

Table 38. Quantitative rich purchase user-item purchase frequency matrix.

User/ item	Milk	Bread	Butter	Cream	Cheese	Honey
User 1	0.35	?	0.70	0.35	0.35	0.35
User 2	0.35	?	0.35	0.35	0.70	0.35
User 3	0.48	?	0.53	0.38	0.47	0.40

In [6], user-based collaborative filtering was used to compare and evaluate the performance of recommendation systems ChoiRec12, HPCRec18, and HSPRec19 against traditional CF algorithm in terms of quality of ratings prediction with respect to predictive accuracy measure Mean Absolute Error (MAE) metric by varying number of users and nearest neighbors. MAE compares the predicted ratings to actual user ratings over a test sample in a recommendation system and is defined as the average absolute difference between predicted ratings and actual ratings. User-based collaborative filtering was also used to compare and evaluate the performance of recommendation systems ChoiRec12, HPCRec18, and HSPRec19 against traditional CF algorithm in terms of quality of ratings prediction with respect to predictive accuracy measure Mean Absolute Error (MAE) metric by varying number of users (left side graph) and nearest neighbors (right side graph). MAE compares the predicted ratings to actual user ratings over a test sample in a recommendation system and is defined as the average absolute difference between predicted ratings and actual ratings. The performance of SP-based E-commerce RS like ChoiRec12, HPCRec18, and HSPRec19 systems were evaluated by in terms of quality of recommendations generated by varying number of users with respect to classification accuracy measures such as precision and recall, which evaluates the frequency of the system making correct/incorrect decisions. Precision is the fraction of all recommended items that are relevant, and Recall is the fraction of all relevant items that were recommended. The results obtained from the experimental comparative analysis of Traditional CF, ChoiRec12, HPCRec18 and HSPRec19 systems conducted by [6] have shown that HSPRec19 system performed the best in comparison to the other recommendation systems as it used SPM (GSP algorithm) to discover frequent historical sequential patterns and analysed the clickstream behaviour for improving the consequential bond between clicks and purchases to enhance user-item frequency matrix quantitatively and qualitatively to generate a rich user-item matrix for CF thereby, resulting in better recommendations in terms of reduced data sparsity and improved recommendation accuracy, scalability, diversity and novelty. Thus, out of all the reviewed SP-based E-commerce RS, it is found that HSPRec19 system for the purpose of recommendation in a real-life application scenario performs best.

3. A Taxonomy of Sequential Pattern-based E-Commerce Recommendation Systems

Taxonomy for existing SP-based E-commerce RS is proposed based on the following two categories:

3.1. Effect of Sequential Patterns on Recommendation Systems with respect to improving the Quality and Quantity of User-Item Rating Matrix Input

The user-item rating matrix is usually sparse, less informative and leads to poor recommendations [20]. In order to capture more real-life customer purchase behavior, and provide historical purchase relationship between already purchased items and recommended items, historical sequential purchase patterns of a user are analyzed and integrated into user-item rating matrix input. This has the effect of enhancing and improving rating quality (specifying level of interest or value for already rated items) and quantity (finding possible rating for previously unknown ratings) by using mined sequential patterns. Table 39 shows how the surveyed recommendation systems improved the quality and quantity of user-item rating matrix input with the use of sequential patterns in comparison to each other.

Table 39. How surveyed recommendation systems improved quality and quantity of input rating matrix.

Rec System	Improving rating quality	Improving rating quantity
ChoRec05	No use of historical purchases or clickstream data, no rating quality improvement	association rule mining used to predict purchases
ChenRec09	RFM - Recency, Frequency and Monetary used to improve rating quality	Modified Apriori used to predict purchases
HuangRec09	No use of historical purchases or clickstream data, no rating quality improvement	association rule mining used to predict purchases
LiuRec09	RFM - Recency, Frequency and Monetary used to improve rating quality	Modified Apriori used to predict purchases
ChoiRec12	use of historical, frequency of purchases, relative preference to improve rating quality	association rule mining used to predict purchases
Hybrid Model RecSys16	use of clicks, collect add-to-cart, payment, etc. to improve rating quality	sequential pattern mined with Prefix-Span used to predict purchases
Product RecSys16	use of historical and frequency of purchases to improve rating quality	sequential pattern mined with FreeSpan used to predict purchases
SainiRec17	use of historical purchases to improve rating quality	sequential pattern mined with Spade used to predict purchases

Table 39. Cont.

Rec System	Improving rating quality	Improving rating quantity
HPCRec18	use of historical, frequency click stream, consequential bond of purchases, to improve rating quality	Analyzed the session-based data mined with consequential bond used to predict purchases even for items with no ratings
HSPRec19	use of historical, frequency click stream, consequential bond of purchases, to improve rating quality	sequential pattern mined with GSP and consequential bond to predict purchases even for items with no ratings

3.2. Effect of Sequential Patterns on Recommendation Systems with respect to handling the problems of Sparsity, Novelty and Scalability

In academic environments, the evaluation of recommendation systems performance is dominated by simulation-based experiments on historical rating or implicit feedback datasets. The quality of the output of an algorithm can then be assessed with the help of accuracy metrics. Being able to accurately predict the relevance of items for users is and will be a central problem of recommendation systems research. Increasing the prediction accuracy therefore is a relevant goal of research [37]. But accuracy alone is not enough. Recommending items that the user might have bought anyways might be of little business value. Hence, focusing on accuracy alone can lead to monotonous recommendations and limited discovery. Thus, it is important that the recommendation systems can assess multiple, possibly competing goals in parallel such as handling data sparsity, improving novelty and scalability of the recommendation systems.

Sparsity: In practice, many commercial recommendation systems (e.g., book recommendation in Amazon.com) are used to evaluate large product sets. In these systems, even active customers may have purchased only under 1% of the products (1% of 2 million books is 20,000 books) i.e. only a few of the total number of items available in a database are often rated by users. Thus, in any recommendation system, the number of ratings already obtained is usually very less when compared to the number of ratings that needs to be predicted. This results in a sparse user-item matrix and generates weak or poor recommendations as a result of insufficient rating information.

Novelty: The novelty evaluates the likelihood of a recommendation system to give recommendations to the user that they are not aware of, or that they have not seen in the past.

Scalability: It has become increasingly easy to collect large number of ratings and implicit feedback information from various users in recent years. In such cases, the size of the data set continues to increase over time. As a result, it has become increasingly essential to design recommendation systems that can perform effectively and efficiently in the presence of large amounts of data.

Table 40 shows the effect of SP on surveyed recommendation systems performance by examining how all the surveyed algorithms handled the problems like sparsity, novelty, scalability and improved the User-Item (U-I) rating quality and quantity of recommendation systems with the use of sequential patterns in comparison to each other. The interpretation of the terms high, medium and low (in Table 40) with respect to the individual functionalities is defined below, followed by an explanation as to why these systems are in a specified range.

Table 40. Effect of SP on surveyed recommendation systems performance.

Rec Sys Performance Factor	Reducing Data Sparsity	Improving Novelty	Improving Scalability	Improving U-I rating quality	Improving U-I rating quantity
ChoRec05	Low	High	Medium	Low	Low
ChenRec09	Medium	Low	Low	Medium	Medium
HuangRec09	Low	High	Medium	Low	Low
LiuRec09	Low	High	Medium	Low	Low
ChoiRec12	Medium	Low	Low	Medium	Medium
Hybrid Model RecSys16	Medium	Low	Medium	Medium	Medium
Product RecSys16 RecSys16	Medium	High	High	Medium	Medium
SainiRec17	Medium	Low	Low	Medium	Medium
HPCRec18	High	Low	Low	High	High
HSPRec19	High	High	Medium	High	High

Reducing Data Sparsity

Low: No use of SPM, instead Association rule mining was used. Medium: Used SPM but couldn't integrate any other implicit user behavior like clickstream data etc. High: Used SPM and integrated additional behavioral data like clickstream data to enhance user/item matrix.

Improving Novelty

Low: previously purchased items by the target user were also included in the recommendation list. High: known items were excluded from being recommended and associated products to purchased products were. used for recommendation purposes to make the customer aware of potentially related products.

Improving Scalability

Low: No clustering technique was used to reduce the dimensionality of the dataset. Medium: A clustering technique was used to reduce the dimensionality of the dataset. High: A clustering technique along with an additional dimensionality reduction technique was used.

Improving U-I (User-Item) rating quality and quantity

Low: No user's historical purchases, clickstream behavior, frequency of the purchase or other user purchase behavior was mined to be integrated into the U-I rating matrix Medium: Minimum information such as only one among the user purchase behavior like association rules (which are not as informative as sequential patterns), user's historical purchases, clickstream behavior, frequency of the purchase are incorporated into the U-I rating matrix which is a less complex method of mining user purchase behavior High: More informative customer purchase historical behavior features are mined and incorporated into U-I rating matrix such as clickstream behavior, consequential bond information of historical clicks and purchases of a user, historical sequential purchase behavior (sequential patterns) etc. Early hybrid recommendation systems like ChoRec05, ChenRec09, HuangRec09, LiuRec09 and ChoiRec12 used association rule mining for improving the quality of rating input. None of these systems incorporated additional customer behavioral data like clickstream data or browsing history from which implicit behavior can be extracted and is used to fill the unknown ratings. Hence these systems are assigned a "low" level on reducing data sparsity. HSPRec19 system could achieve this to a higher extent by using SPM (GSP algorithm) to derive sequential patterns for improving the rating quality and quantity. Thus, this system is assigned a "high" level on reducing data sparsity. The remaining four systems (Hybrid Model RecSys16, Product RecSys16, SainiRec17 and HPCRec18) didn't

integrate any additional behavior but extracted the sequential patterns using SPM algorithms like PrefixSpan [34], FreeSpan [38] and SPADE [36] which resulted in reducing data sparsity to a “medium” level. The novelty rate is defined “low” if the previously purchased items by the target user were included in recommendation list because novelty accounts for the likelihood of a recommendation system to give recommendations to the user that they are not aware of. Thus, the novelty rate is defined “high” if the known items were excluded from being recommended and associated products to the purchased products were used for recommendation purposes to make the customer aware of potentially related products. The dimensionality of a dataset is reduced by using either a clustering technique or by explicitly using a dimensionality reduction technique and sometimes both. Downsizing the data dimension leads to an increase in the scalability of the recommendation system. Thus, if no clustering technique was used by the system, then improving the scalability was specified as “low” and if a clustering technique was used to reduce the dimensionality of the dataset then improving the scalability was specified to be “medium” and if a clustering technique along with an additional dimensionality reduction technique was used by the system then improving the scalability was specified “high”.

4. Conclusions and Future Work

Recommendation Systems open new opportunities of retrieving personalized information on the internet by enabling users to have access to products and services which are not readily available to users on the system. Many recommendation systems neglect sequential patterns during recommendation. Thus, to verify the necessity of sequential patterns in E-commerce recommendation systems, a survey of the existing SP-based E-commerce RS is conducted, and a taxonomy is developed that classifies these applications by their recommendation method and performance factors like reducing data sparsity, improving scalability of recommendation systems and improving accuracy and novelty of recommendations. Furthermore, after performing a comparative analysis of traditional CF against few of the surveyed SP-based E-commerce RS, the results have proved that the hybridization of SPM with CF by integrating sequential patterns into the user-item rating matrix input, improved the recommendation quality in terms of accuracy, diversity and novelty. Additionally, we would like to direct the reader to open research subjects that warrant future works in the area of SP-based E-commerce RS and the ideas for future work in this direction include:

1. None of the reviewed studies exactly measured the level of probability of purchase determined by each SP, instead the general mid-way of 50% [6] was used for example. Hence, more information (such as the frequency of the patterns occurring together) in the historical data should be used to determine the exact level of probability of purchase (e.g., 0.5 to 1.0) for each SP.
2. More possible ways of incorporating click stream sequences/patterns into the User-Item rating matrix should be found with the use of consequential bond to improve the input User-Item rating quality. Also, additional information such as contextual data (e.g., time of the year, such as season or month, or day of the week etc.) should be integrated into user-item preferences.
3. Incorporating the factor of profit or utility for finding patterns (apart from just finding the frequent sequential patterns) from historical purchase data would result in profitable recommendations. Thus, high utility sequential patterns should be integrated into the recommendation generation processes.
4. In real world, items purchased by a user during a certain time period are often from multi-domains rather than one domain. Essentially, there are some sequential dependencies between items from different domains (e.g., the purchase of a car insurance after the purchase of a car). Such cross-domain sequential dependencies are ignored in most sequential pattern-based recommendation systems. Therefore, cross-domain recommendation systems is another promising research direction to generate more accurate recommendations by leveraging information and diverse recommendations from different domains.
5. Another good line of future research is the evaluation strategy used to assess the performance of sequential pattern-based recommendation systems, as all the reviewed studies were evaluated based

on the offline approaches. Although the offline evaluation is of lower cost with no bias of response from active user involvements as in the case of online and user studies, the results mostly contradict when applied in real-life applications with the online and user studies evaluations. Therefore, there is a huge need for more research on the evaluation strategies to compare performance based on different performance measures other than accuracy and offline evaluation, like real-time, novelty, coverage, serendipity and diversity of recommendations.

Conflicts of Interest: On behalf of all authors, the corresponding author states that there is no conflict of interest.

References

1. Aggarwal, C. C. (2016). An introduction to recommender systems. In *Recommender Systems*, Springer, 1-28.
2. Karlapalepu, Hemni S. (2020). A Taxonomy of Sequential Patterns Based Recommendation Systems. Unpublished MSc thesis, University of Windsor, Windsor, Ontario, Canada, April.
3. Ricci, F., Rokach, L., Shapira, B., Kantor, P. B. (Eds.). (2011). *Recommender Systems Handbook*.
4. Han, J., Kamber, M., Pei, J. (2011). *Data mining: Concepts and techniques*. Amsterdam: Elsevier.
5. Agrawal, R., Srikant, R. (1996). Mining quantitative association rules in large relational tables. *Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data - SIGMOD '96*, 1-12.
6. Bhatta, R., Ezeife, C., Butt, M. N. (2019). Mining sequential patterns of historical purchases for e-commerce recommendation. In *International conference on big data analytics and knowledge discovery*, 57-72.
7. Mabroukeh, N. R. Ezeife, C. I. (2010). A taxonomy of sequential pattern mining algorithms. *ACM Comput. Surv.* 43, 1, 3.
8. Mooney, C. H., Roddick, J. F. (2013). Sequential pattern mining – approaches and algorithms. *ACM Computing Surveys*, 45(2), 1-39.
9. Adomavicius, G., Tuzhilin, A. (2005). Towards the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17, 6, , 734–749.
10. Bokde, D., Girase, S., Mukhopadhyay, D. (2015). Matrix Factorization Model in Collaborative Filtering Algorithms: A Survey. *Procedia Computer Science*, 49, 136-146.
11. Quadrana, M., Cremonesi, P., Jannach, D. (2018). Sequence-aware Recommender Systems. *ACM Computing Surveys*, 1(1), 1-35.
12. Patel, D., Patel, F., Chauhan, U. (2023). Recommendation Systems: Types, Applications, and Challenges. *International Journal of Computing and Digital Systems*, Vol. 13, No. 1, pp. 850-868.
13. Zangerle, E., Bauer, C. (2022). Evaluating Recommender Systems: Survey and Framework. *ACM Computing Surveys*, Vol. 55, No. 8, Article 170. pp. 1-37.
14. Fang, H., Zhang, D., Shu, Y., Guo, G. (2020). Deep Learning for Sequential Recommendation: Algorithms, Influential Factors, and Evaluations, *ACM Transactions on Information Systems*, Vol. 1, No. 1, Article 1, pp. 1-41.
15. Nasir, M., Ezeife, C. I., Gidado, A. (2021). Improving E-Commerce Product Recommendation using Semantic Context and Sequential Historical Purchases. *Springer's International Journal of Social Networks Analysis and Mining*. Vol. 11, Issue 82, pp. 1-25.
16. Kazienko, P., Pilarczyk, M. (2008). Sequence aware recommenders for fashion E-commerce, *Electronic Commerce Research*, Springer Nature, pp. 1-22. <https://doi.org/10.1007/s10660-022-09627-8>.
17. Kazienko, P., Pilarczyk, M. (2008). Hyperlink Recommendation Based on Positive and Negative Association Rules. *New Generation Computing*, 26(3), 227-244.
18. Li, C., Niu, X., Luo, X., Chen, Z., Quan, C. (2019). A Review-Driven Neural Model for Sequential Recommendation. *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, 2886-2872.
19. Wang, S., Hu, L., Wang, Y., Cao, L., Sheng, Q. Z., Orgun, M. (2019). Sequential Recommender Systems: Challenges, Progress and Prospects. *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*.
20. Bucklin, R. E., Sismeiro, C. (2003). A model of web site browsing behavior estimated on clickstream data. *Journal of marketing research*, 40, 249-267.

21. Sarwar, B., Karypis, G., Konstan, J., Riedl, J. (2000). Analysis of recommendation algorithms for e-commerce. *Proceedings of the 2nd ACM Conference on Electronic Commerce - EC '00*, 158-167.
22. Chiu, D.-Y., Wu, Y.-H., Chen, A. L. P. (2004). An efficient algorithm for mining frequent sequences by a new strategy without support counting. In *Proceedings of the 20th International Conference on Data Engineering*. 375–386.
23. Cho, Y. B., Cho, Y. H., Kim, S. H. (2005). Mining changes in customer buying behavior for collaborative recommendations. *Expert Systems with Applications*, 28 (2), 359-369.
24. Chen, Y. L., Kuo, M. H., Wu, S. Y., Tang, K. (2009). Discovering recency, frequency, and monetary (RFM) sequential patterns from customer's purchasing data. *Electronic Commerce Research and Applications*, 8(5), 241-251.
25. Huang, C. L., Huang, W. L. (2009). Handling sequential pattern decay: Developing a two-stage collaborative recommender system. *Electronic Commerce Research and Applications*, 8(3), 117–129.
26. Liu, D., Lai, C., Lee, W. (2009). A hybrid of sequential rules and collaborative filtering for product recommendation. *Information Sciences*, 179(20), 3505-3519.
27. Choi, K., Yoo, D., Kim, G., Suh, Y. (2012). A hybrid online-product recommendation system: Combining implicit rating-based collaborative filtering and sequential pattern analysis. *Electronic Commerce Research and Applications*, 11 (4), 309-317.
28. Fang, Z., Zhang, L., Chen, K. (2016). A behavior mining based hybrid recommender system. 2016 IEEE International Conference on Big Data Analysis (ICBDA).
29. Jamali, S., Navaei, Y. D. (2016). A two-level Product Recommender for E-commerce Sites by Using Sequential Pattern Analysis. *International Journal of Integrated Engineering*, 8(1).
30. Saini, S., Saumya, S., Singh, J. P. (2017). Sequential purchase recommendation system for e-commerce sites. In *IFIP International Conference on Computer Information Systems and Industrial Management* (pp. 366-375).
31. Xiao, Y., Ezeife, C. I. (2018). E-Commerce Product Recommendation Using Historical Purchases and Clickstream Data. *Big Data Analytics and Knowledge Discovery Lecture Notes in Computer Science*, (pp. 70–82).
32. Zhao, Q., Bhowmick, S. S. (2003). Association rule mining: A survey. Nanyang Technological University, Singapore, 135.
33. Kaymak, U. (2001). Fuzzy target selection using RFM variables. *Proceedings Joint 9th IFSA World Congress and 20th NAFIPS International Conference* (Cat. No. 01TH8569), 2, 1038-1043.
34. Pei, J., Han, J., Mortazavi-Asl, B., Pinto, H. (2001). PrefixSpan: Mining sequential patterns efficiently by prefix-projected pattern growth. In *Proceedings of the International Conference on Data Engineering*. 215–224.
35. Song, Wei., Yang, Kai. (2014). Personalized Recommendation Based on Weighted Sequence Similarity. In *Practical Applications of Intelligent Systems*. Springer, (657–666).
36. Zaki, M. J. (2001). SPADE: An efficient algorithm for mining frequent sequences. *Mach. Learn.* 42, 31–60.
37. Jannach, D., Jugovac, M. (2019). Measuring the Business Value of Recommender Systems. *ACM Transactions on Management Information Systems*, 10(4), 1-23.
38. Han, J., Pei, J., Mortazavi-Asl, B., Chen, Q., Dayal, U., HSU, M.-C. (2000). Freespan: Frequent pattern projected sequential pattern mining. In *Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, New York, 355–359.
39. Burke R., 2002. Hybrid recommender systems: Survey and experiments. *User modeling and user-adapted interaction*. 12(4), 331–370.
40. Chun, J., Oh, J. Y., Kwon, S., Kim, D. (2004). Simulating the Effectiveness of Using Association Rules for Recommendation Systems. *Lecture Notes in Computer Science Systems Modeling and Simulation: Theory and Applications*, 306-314.
41. Konstan, J. A., Miller, B. N., Maltz, D., Herlocker, J. L., Gordon, L. R., Riedl, J. (1997). GroupLens: Applying collaborative filtering to Usenet news. *Communications of the ACM*, 40(3), 77-87.
42. Liu, Y., Liao, W.-k., Choudhary, A. (2005). A two-phase algorithm for fast discovery of high utility itemsets. In *Pacific-asia conference on knowledge discovery and data mining* (pp. 689-695).
43. Schafer, J. B., Frankowski, D., Herlocker, J., Sen, S. (2007). Collaborative filtering recommender systems. In *The adaptive web* (pp. 291-324). Springer, Berlin, Heidelberg.

44. Wei, K., Huang, J., Fu, S. (2007). A survey of e-commerce recommender systems. Service systems and service management, 2007 international conference on, (pp.1-5).
45. Xu, M., Liu, F., Xu, W. (2019). A Survey on Sequential Recommendation. In 2019 6th International Conference on Information Science and Control Engineering (ICISCE) (pp. 106-111). IEEE.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.