

Review

A Survey of Sequential Pattern Based E-Commerce Recommendation Systems

Christie I. Ezeife ^{*,†}  and Hemni Karlapalepu [†]

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

* Correspondence: cezeife@uwindsor.ca

† These authors contributed equally to this 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 review focuses on algorithms of existing E-commerce recommendation systems that are sequential pattern-based. It provides a comprehensive and comparative performance analysis of these systems, exposing their methodologies, achievements, limitations, and potential for solving more important problems in this domain. The review shows that integrating sequential pattern mining of historical purchase and/or click sequences into a user-item matrix for collaborative filtering can (i) improve recommendation accuracy, (ii) reduce user-item rating data sparsity, (iii) increase the novelty rate of recommendations, and (iv) improve the scalability of recommendation systems.

Keywords: recommendation systems; collaborative filtering; sequential patterns; e-commerce; purchase and click stream



Citation: Ezeife, C.I.; Karlapalepu, H. A Survey of Sequential Pattern Based E-Commerce Recommendation Systems. *Algorithms* **2023**, *16*, 467. <https://doi.org/10.3390/a16100467>

Academic Editors: Vicente García-Díaz and Frank Werner

Received: 17 August 2023
Revised: 22 September 2023
Accepted: 23 September 2023
Published: 3 October 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Recommendation systems have become the heart of 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. These suggestions 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 databases through tasks such as 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 purpose of making recommendations in the e-commerce application domain. Different types of recommendation systems accept different input data through explicit rating feedback (e.g., Table 1) and implicit derived feedback. Explicit feedback can be in the form of collecting ratings of products or text comments by users through registration forms/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 with respect to 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, and 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 consisting of input

data instance of a movie recommendation site (Table 1) is 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 with a question mark '?' are the missing or unknown 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; a value of 1 is assigned for the purchased items, while 0 represents non-purchased items by the same user. Analyzing users’ implicit preferences (i.e., the behavior pattern data) has been used widely and has proven 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 the 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 the collection of store items purchased every week for each customer for one month. An example of historical daily purchase data for a 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 the second purchase, Milk in the third purchase, and 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 occur 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 purchase Sugar and Tea together in a subsequent purchase. Support of a sequential pattern in a database table is defined as the number of records the pattern occurred in divided by the total number of records in the databases. A Sequential Historical Database (SHOD) algorithm was used in the 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. For

- (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 $\text{min sup } \zeta$, 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 subsequences S in the given sequence database SDB of items I at the given $\text{min sup } \zeta$ 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 ($\text{min sup } \zeta$) [7].

The input of an e-commerce recommendation system based on a 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, i.e., less informative rating data not reflecting (1) how much a user likes a purchased item with value 1, (2) how frequently

or how long ago a user purchased an item, or (3) what quantity of a product was purchased. One way to improve the input data is to integrate explicit rating with implicit rating drawn from historical purchase or click stream data, or alternatively 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. This can be integrated into the user–item rating matrix, helping to reduce data sparsity and improve recommendation quality and accuracy. SPM can capture customer purchase behavior over time using mined sequential patterns; this is crucial, as the time interval between items is useful for learning the time at what the next item might be purchased. The next purchase decision of a user is often influenced by their recent behaviors, and this approach considers the temporal preferences of users 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 it is generally 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 such as this one, several of the unknown ratings in the input user–item rating matrix in Table 6 can be filled such that all users who have purchased the antecedent items (milk, bread) have a higher chance (say, 0.5 or a more specific determined chance value) of purchasing 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 the quantity of rating values by providing possible values for missing/unrated items. A user–item purchase frequency matrix can then be constructed in which 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, improving the 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 ratings for previously unknown ratings) can be enhanced and improved 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 that user’s history of shopping. This problem can be modeled using one of the most successful methods in the literature, the Collaborative Filtering (CF) technique, which makes use of explicit user rating–item matrix data from the user for the purpose of recommendation. The general technique of CF [1] accepts as its input an incomplete user–item rating matrix with the goal of predicting the unknown rating of a target user or item. For user-based CF, ratings of like-minded users of a target user u are used to make recommendations for user u . The weighted ratings of this group of neighbors are used as the predicted rating of the targeted user u . Similarly, functions can be computed between the rows of the ratings of item–user matrix to discover similar users likely to purchase the same types of products. 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 are greatly reduced when the user's explicit rating behavior data is sparse; second, these methods ignore the time context of user behavior i.e., how the customer's purchase behavior may vary over time, and as such are unable to capture the sequential behavior of users. SPM techniques [7,8] have been used alone recently to make recommendations more effective by extracting sequential patterns of user purchase behavior, as the user's next purchase is affected by their previous purchases and actions. This recommendation often utilizes a user's implicit feedback data; the major advantage of this model is its ability to capture users' 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 these methods (CF and SPM) have shortfalls. In fact, both sequential behavior and users' general taste are important factors that influence user purchasing behavior, as indicated in [9–11]. This motivates a systematic review of the importance of integrating SPM with CF for recommendation systems to improve recommendation quality through more diverse recommendations, closing the high sparsity matrix problem, and thereby making recommendations better by taking into account users' 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, and user scalability), recommendation approaches, improving on understanding of the system's algorithms with example application of system through a clear example, and highlighting their strengths, weaknesses, and future prospects in the recommendation process. The focus of the survey research in this paper is on in-depth understanding of algorithmic methods for collaborative filtering system-based RS that enhance recommendation quality through sequential pattern mining of historical purchase and click stream data. Earlier work and surveys on collaborative filtering based techniques include [12–18]. This work is different from existing surveys or reviews of methods for evaluating recommendation systems, such as [19,20], which provide a framework with no discussion of any algorithms. This survey of more traditional and more technically understandable mining based approaches is different from other related surveys or research on complex deep learning-based sequential recommendation systems [11,21–25], which do not exploit historical and click stream purchase data for tracking temporal customer purchase behavior.

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 [26]. This has 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 [27], 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 whether a user has purchased a product before. This user-item rating matrix is usually sparse, less informative, and leads to poor recommendations [28]. 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 such as Amazon.com is 20,000), i.e., often only a few of the total number of items available in a database are rated by users [29]. Thus, in order to capture more real-life customer purchase behavior and provide the relationship between already purchased items and recommended items, the historical sequential purchase patterns of a user can be analyzed and integrated into the user-item matrix input to enhance and improve the rating quality and quantity by providing the possible values for missing/unrated items. To demonstrate this, consider the historical purchase data in 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, etc.

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 the 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, such as GSP [5], and extract the possible purchase sequential rules (Table 10) from frequent purchase sequences. Using these sequential purchase rules, the unknown ratings in the user–item purchase frequency matrix (e.g., the value of User 1 for the item Cheese in Table 8) can be filled using a predicted value such that all users who have purchased the antecedent items such as (Milk, Butter) from Rule No. 1 of Table 10 have a higher chance (say, 0.5 or a more specific determined chance value for the highly probable purchases determined by the SPs) of purchasing Cheese next. Hence, using Rule No. 1 it can be inferred that as User 1 purchased Milk and Butter in this transaction, there is a high chance that they would purchase Cheese in the same transaction. Hence, we assign a value of 0.5 to the user–item combination (User 1–Cheese). Similarly, (User 2–Cream) is filled using Rule No. 3 and (User 2–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 the 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 rest of this 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 provides 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. Finally, conclusions and future work are discussed 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 transaction data provide sets of preferred items and can be used to predict future customer preferences, researchers have applied association rule mining techniques to extract sequences in order to improve the performance of recommendation systems [30,31]. 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, the accuracy of the recommendation system is improved, as the input becomes more informative before it is fed to CF. Thus, integrating CF and SPM of historical purchase data can improve recommendation quality, reduce data sparsity, and increase the novelty of recommendations. While sequential pattern mining algorithms such as (GSP [5], SPADE [32], and PrefixSpan [33]) mine frequent patterns from sequence database (e.g., of historical purchases or clicks) using a priori-like logic [34], Collaborative Filtering algorithms [1] follow a general four-step logic, to predict missing ratings in the input user–item rating matrix. First, the mean rating is computed for each user u . Second, the similarity between a target user v and all the other users u is computed using a similarity function such as the Pearson correlation coefficient. Third, user v ’s peer group is computed for each of v ’s unrated items (that is, v ’s top similar users). Finally, the predicted rating of user v is computed as the weighted average of the ratings of v ’s top K similar users for the unrated item.

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 include the following ten systems: (1) model-based approach, ChoRec05 [35], (2) pattern segmentation framework, ChoRec09 [36], (3) sequential pattern-based collaborative recommender system, HuaRec09 [37], (4) segmentation-based approach, LiuRec09 [38], (5) hybrid online product recommendation, ChoiRec12 [39], (6) hybrid model (HM), RecSys16 [40], (7) product recommendation system (PRS), RecSys16 [41], (8) sequential pattern-based recommender system, SainiRec17 [42], (9) historical purchase and click stream-based recommendation, HPCRec18 [43], and (10) historical sequential pattern recommendation, HSPCRec19 [6]. A brief overview of these systems is provided next. Each reviewed system has a summary of its algorithm and methodology presented first before an example application of the system’s algorithm to assist with clarity of the technique, where applicable for most systems.

2.1. Model-Based Approach: ChoRec05 [35]

Summary of ChoRec05 [35]: A hybrid recommendation system that combines the 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 [35]. The self organizing map (SOM) allows customer historical purchase transactions arranged in purchase time to be assigned a numbered cluster. A change in customer cluster number as a result of each transaction is used to create customer purchase sequences. Further, association rule mining is used to mine sequential patterns from the built customer sequences. The problem solved by [35] can be phrased as follows. For a given purchase sequence of a target customer c , based on a time period l (with l for example, being a period of every month or three months), for the past $l-1$ periods before time T , find the product p that the target customer will most likely buy at time T . The recommendation procedure is divided into two components, a model-building phase and a recommendation phase.

An Example Application of ChoRec05 System [35]: The problem being solved is stated as: Given the input data set as: Input: historical purchase data in E-commerce dataset including customer ID, purchased items, and duration of transaction., when the expected output data sets are: Output: recommends products to each user., apply ChoRec05 algorithm to solve this recommendation task.

This ChoRec05 [35] algorithm will start by going through the model building phase. Model-building phase: this phase is performed once to create a reliable model from the customer transaction database, which includes transaction clustering, where the transactions are transformed into an input matrix composed of a bit vector. These time-ordered vectors for a given customer represent the purchase history of that customer; the input matrix can be thought of as the customer’s dynamic profile.

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 (e.g., in January, February, March). Using the customer transaction change cluster database (e.g., Table 12, called loci in [35], association rule mining is used to predict customer movement. In Table 12, the first row means that a customer with CID 001 purchase pattern is in Cluster 9 in January (which is time T-2), then moved to Cluster 2 in February (time T-1), and then moved to Cluster 8 in time T (March).

Table 12. Purchase behavior loci of selected customers.

CID	T-2 (Jan)	T-1 (Feb)	T (Mar)
001	9	2	8
002	9	1	2
003	2	9	3
004	9	-	8

Extraction of sequential cluster rules: to mine customer behavior according to purchase time, the association rule R_i [44] is adopted for determining the most frequent rule patterns that equal or exceed a given minimum support and minimum confidence, as shown in the sample rule in Table 13. With association rule mining, in the first phase, frequent patterns are mined from the input database. Then, association rules are derived from these frequent patterns and only rules with confidence equal to or exceeding a set minimum confidence level are retained as being significant. The confidence of a rule is the number of times both the left and right sides of the rule (antecedent \rightarrow consequent) appear in the database divided by the number of times only the left side of the rule appears in the records of the database. A sample mined rules obtained from the customer transaction loci (movement) is shown in Table 13.

$$R_j = r_{j, T - 1 + 1, \dots, r_{j, T - 1} \rightarrow r_{j, T} \text{ (support, confidence)}.$$

Here, the 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 at time T is $r_{j, T}$.

Table 13. Sample derived association rules.

Rules	T-2 (Jan)	T-1 (Feb)	T (Mar)	Support	Confidence
1	9	-	8	0.3	0.667
2	2	9	3	0.1	1.0
3	9	2	8	0.1	1.0
4	9	1	2	0.1	1.0

Then ChoRec05 [35] algorithm will follow up the model building phase, with the recommendation phase as discussed next. Recommendation phase: in this phase, the products that are best matched to the dynamic behaviors of the target customers are found and the relevant transactions are converted into behavior loci 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 (see Table 14). In this table, the selected cluster has four products purchased in descending order of quantity, with most purchased being brand 23 with 5, brand 21 with 4, brand 28 with 3, and brand 27 with 2.

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

Customer	Purchased Products (Brand)
CID 001	Brand 21 (purchased 2), 23 (purchased 3)
CID 002	Brand 21 (purchased 2), 27 (purchased 2)
CID 003	Brand 23 (purchased 2), 28 (purchased 3)

2.2. Pattern Segmentation Framework: ChenRec09 [36]

Summary of ChenRec09 [36] System: Chen et al. [36] proposed a sequential pattern-based recommender system that incorporates the RFM (Recency, Frequency, and Monetary) concept. “Recency” represents the length of the time period since the last purchase, with a lower value corresponding to a higher probability of the customer making repeat purchases. “Frequency” denotes the number of purchases within a specified time period, with a higher frequency indicating stronger customer loyalty. “Monetary” means the amount of money spent in this specified time period; thus, 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 used to generate all RFM sequential patterns from customer’s purchase data. The algorithm was developed by modifying the well-known a priori GSP sequential pattern mining algorithm [5], and consists of iterative phases.

The RFM-Apriori Algorithm goes through Candidate generation phase: first, where the algorithm places all itemsets into the candidate-1 set C_1 , the set of candidate patterns with length 1, then scans the database to find the frequent (large) 1-patterns (L_1) from the candidate itemsets C_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, thereby improving efficiency. Second, supposing that the set of frequent $(k-1)$ -patterns L_{k-1} is already known, it is joined with itself apriori-gen join way to generate candidate RF

patterns of length k , where $k \geq 2$ if they have the same $(k-2)$ -postfix. The algorithm scans the database to determine the supports of the patterns in C_k , then finds L_k by removing those patterns from C_k that have support lower than the minimum support threshold. This iteration is repeated by increasing k by one until no more patterns can be generated. 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. Support values can be accumulated in each leaf node using every data sequence to traverse the tree. 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 is proposed which allows for partitioning the RFM-patterns into segments relevant to the RFM criteria in order 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 [37]

Summary of HuaRec09 [37] System: Huang et al. (2009) proposed a hybrid recommendation system that uses a sequential pattern-based collaborative recommender system to predict 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. Considering all the target customer's transactional sequences in the current time period T and the previous number r periods $T-1, T-2, \dots, T-r$, the active customers most likely to purchase items in the next time period $T+1$ (target prediction period) are determined. 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 goes through the four steps 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) [45].
2. Building dynamic customer profile: dynamic customer buying behaviors can be modeled by analyzing a customer's periodic transaction data.
3. Clustering the customers: the customers are clustered based on their dynamic customer profiles using a genetic algorithm-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 using the product categories involves generating a customer purchase sequence for each customer and discovering the sequential patterns for each cluster using an SPM algorithm such as GSP [5] or PrefixSpan [33].

For model use for the active customer a two-stage recommendation process is followed using 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 the product Category Recommendation Score (CRS). The CRS for the predicted $category_i$ is calculated as follows:

$$CRS^{category_i} = \sum_{period_t} (CATEGORY - SUPPOR_{period_t}^{category_i} \times WEIGHT_{period_t}) \text{ for } T = T_0, T_1, \dots, T_r$$

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:

$$IRS^{item_j} = \sum_{period_t} (Purchase - Frequency_{period_t}^{item_j} \times Weight_{period_t}) \text{ for } T = T_0, T_1, \dots, T_r$$

where $Weight_{period_t}$ is the time window weight in $period_t$ and $Purchase - Frequency_{period_t}^{item_j}$ 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, rather than the of quantity, of purchases during a certain period. The top-N items with larger recommendation scores, excluding items bought by the active customer before, are then recommended to the active customer.

2.4. Segmentation-Based Approach: LiuRec09 [26]

Summary of LiuRec09 System [26]: A hybrid recommendation system which combines the segmentation-based sequential rule method with the segmentation-based KNN-CF method was proposed in [26].

An Example Application of LiuRec09 System:

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

The Segmentation-based Sequential Rule (SSR) method would go through the following steps.

Step 1: Customer clustering. Customers are clustered 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 be merged. Therefore, eight customer clusters can be reduced to four customer segments: loyal, potential, uncertain, and valueless, as based on their RFM patterns 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	40,797.23	R \uparrow , F \downarrow , M \uparrow
Cluster 2	43	119.558	3.791	7342.326	R \uparrow , F \downarrow , M \downarrow
Cluster 3	17	64.294	67.2351	147,315.6	R \downarrow , F \uparrow , M \uparrow
Cluster 4	214	56.696	19.832	40,279.53	R \downarrow , F \uparrow , M \uparrow
Cluster 5	78	57.192	37.846	74,045.92	R \downarrow , F \uparrow , M \uparrow
Cluster 6	367	58.335	9.632	18,677.27	R \downarrow , F \downarrow , M \downarrow
Cluster 7	126	92.246	7.286	14,853.89	R \uparrow , F \downarrow , M \downarrow
Cluster 8	240	73.892	8.496	16,109.99	R \uparrow , F \downarrow , M \downarrow
Average		68.216	14.324	28,638.3	

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↑ (54,691.80)
Potential	104	R↑ (72.260)	F↑ (19.587)	M↑ (40,797.23)
Uncertain	367	R↓ (58.335)	F↓ (9.632)	M↓ (18,677.26)
Valueless	409	R↑ (84.347)	F↓ (7.628)	M↓ (14,801.23)

Step 2: Transaction clustering. Transactions are divided into groups (transaction clusters) based on similar product items and buying patterns. A customer’s transaction clusters are used to identify the sequence of transaction clusters over time. A sample change in customer transactions over 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, an association rule [44] is adopted for determining the most frequent pattern with confidence. From Table 17, a sequential rule $Ap2 \rightarrow Ep3$ (0.4,1) is extracted, 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 their 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: Determine the cluster sequences of target customers and match them. The degree of matching 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 the predicted transaction cluster is calculated and the top N items with the highest frequency counts are returned.

Segmentation-based KNN-CF method (SKCF). In this step, the Pearson’s correlation coefficient for each customer is used to measure the similarity between the target customer and other customers in the same segment, then the k most similar (highest ranked) customers are selected as the k-nearest neighbors of the target customer. 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 in the resulting linear combination of the two methods are selected for recommendation.

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

2.5. Hybrid Online Product Recommendation: ChoiRec12 [39]

Summary of ChoiRec12 [39]: 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 using the number of times that user u purchased

item *i* with respect to the total transactions, which can be used in CF even when the explicit rating is not available.

An Example Application of ChoiRec12 System [39]: For a given a fragment of historical purchase data, such as that in Table 18, where only the purchase dates of items by users are provided as available information, the goal is to recommend suitable items to a user *T*.

Table 18. Purchase dates of items by users.

	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	-	-

The general algorithm follows seven steps to recommend items to user *T*. Step 1: Deriving implicit ratings from user transactions. The implicit rating can be computed based on purchase history 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, a user–item implicit rating matrix created from the historical data can be considered, as in Table 19.

Table 19. Implicit ratings derived from user 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 <i>T</i>	?	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 ratings of items by users with respect to the total numbers of ratings. Thus, the mean rating for user 1 = (3 + 1 + 5)/3 = 3, user 2 = 2.5, user 3 = 2.3, user 4 = 4, and user *T* = 3. The similarities between users are then computed using the cosine similarity, which is provided as follows:

$$\text{Cosine Similarity}(T, b) = \sum_{i=1}^m (R_{T,i}) \cdot (R_{b,i}) / \sqrt{\sum_{i=1}^m (R_{T,i})^2} \cdot \sqrt{\sum_{i=1}^m (R_{b,i})^2}$$

where $(R_{T,i})$ denotes the ratings of users *T* for item *i*; similarly, $R_{b,i}$ denotes the rating of user *b* for item *i*. For example, the calculated similarities between the 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 the cosine similarity between target user *T* and user 1, etc.

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. Thus, the sorted similarities in descending order are $CS(T,4) = 1$, $CS(T,2) = 0.9648$, $CS(T,3) = 0.8944$, $CS(T,1) = 0.7071$. In this case, the top-2 neighbors for target user *T* are 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 the CF-based predicted preference of user *a* on item *i*. For example, the CFPP of a target user *T* on all other items is now

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). The sequence data of each user are generated by sorting the transaction data for each according to the transaction date. Then, frequent items are found using the candidate generation (C_k) and pruning (L_k) processes until the candidate set is empty. Next, the sub-sequences of a target user purchase are matched with derived purchased items by enumerating the target user purchase item. Finally, pattern analysis is conducted based on the predicted preference (SPAPP) of user T on item i. For example, the SPAPP of the 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, and SPAPP (T,5) = 0.5.

Step 6: Integrate CFPP and SPAPP. CFPP and SPAPP are normalized to obtain N_CFPP and N_SPAPP, respectively. The final predicted preference of the target user T on item i, FPP (T,i), is calculated as α times CFPP plus $1 - \alpha$ times SPAPP, where α and $1 - \alpha$ are the respective weights assigned to CF and SPA, and are set to 0.1 and 0.9. The FPP values are 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 the FPP values of the items purchased by the neighbors of the target user, the item with the highest FPP is recommended to the target user T. In the case from Table 20, items 3 and 4 are recommended, as they have the highest FPP values.

2.6. Hybrid Model: HM RecSys16 [40]

Summary of HM RecSys16 System [40]: A hybrid recommender system that combines the prefix span algorithm with traditional matrix factorization was proposed in [40]. SPM aims to find frequent sequential patterns in sequence databases, and is applied in this hybrid model to predict customer payment behavior, contributing to the accuracy of the model. The workflow of the system consists of three phases: Behavior Prediction, CF, and Recommendation.

Purchasing Pattern Extraction

The 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 behavior patterns for a customer browsing or adding an item to the cart. When the recommender system’s behavior monitoring part detects the user’s potential purchasing tendencies, the system fetches the user’s historical behavior record from the sequence database and builds an item–user rating matrix like Table 21, in which each entry contains the historical behavior of the Ith user with respect to the 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

The 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, one representing features of the products and another representing the preferences of users. Multiplying the two matrices provides predictions about the user’s preferences for all products.

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

The r_{ui} represents the rating of item i by user u ; the latent factor model is then 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 preferences collected using the 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 a database called the candidate database. The candidate database is scanned at regular intervals and the sequence containing payment patterns is sent to the recommender system as a potential purchasing sequence. Second, for the potential buyers, the preference information from CF phase which represents the preference degree towards each product is generated. Because the sequential mining phase generates both the payment sequence and the category of the target item, the category-matched items in the preference vector are recommended.

2.7. Product Recommendation System: PRS RecSys16 [41]

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

The PRS (Product Recommendation System) includes two levels of product recommendation; the first level is recommended before product purchase and the other after purchasing. PRS initially collects product data from the electronic store, separates the products according to their type, then clusters them based on their numerical attributes into three separate clusters of high, medium, and low quality using the C-means algorithm. Here, the C-Means clustering algorithm is used to separate products by their types and

create groups with similar features to classify products. This algorithm generates clusters based on fuzzy logic, and does not consider sharp boundaries between the clusters, allowing each feature vector to belong to different clusters by a certain degree. The degree of membership between a feature vector and a cluster is usually considered as a function of its distance from the cluster centroid points, based on minimization of the following objective function:

$$J_m = \sum_{i=1}^N \sum_{j=1}^C u_{ij}^m \|X_i - c_j\|^2, 1 \leq m \leq \infty,$$

where m is any real number greater than 1, u_{ij} is the degree of membership of x_i in cluster j , x_i represents the i -th d -dimensional measured data, c_j is the d -dimensional center of the cluster, and $\| * \|$ is any norm expressing the similarity between any measured data and the center.

Next, the PRS tries to identify the customer's requirements and criteria using an online form that takes information about product such as type, quality, price, brand, etc. This information is used to assign an appropriate cluster to the customer. In the second level, information about the history of the customer's shopping behavior is collected. This information is used to explore the relations between products using the Freespan algorithm of the SPA method. Freespan mines sequential patterns by partitioning the search space and projecting the sequence subdatabases recursively based on the projected itemsets [46]. Eventually, these relations and patterns are provided as product recommendations, recommending products associated with the purchased products, making the customer aware of potentially related products, as relationships between products increase the likelihood of a customer buying those products together.

2.8. Sequential Pattern-Based Recommender System: SainiRec17 [42]

Summary of SainiRec17 System [42]: Saini et al. (2017) tried to find the sequence of all items which were bought regularly, that is, finding the same product purchased every month as well as the different products purchased one after another in a sequence. Users buy certain products in a sequence; for example, most users buy a mobile phone and mobile phone cover in sequence. Thus, this approach tries to find such sequences in the online shopping context. The main objective is to find those sequences that are most frequent among all users, with the intra-duration being the sequence in an online product purchasing system. With the help of the SPADE [32] algorithm, frequent sequential purchase patterns can be found; then, in the next step, a sequence mining algorithm is applied to find those sequences available in the dataset. Finally, the time elapsed between the purchase of first product and next sequential product is calculated by finding the mean and mode of the duration followed by all users. Here, the mean provides the average time gap between products, while the mode provides the duration followed by most of users.

2.9. Historical Clickstream-Based Recommendation: HPCRec18 [43]

Summary of HPCRec18 System [43]: The novel recommendation system called Historical Purchase with Clickstream recommendation system (HPCRec) integrates purchase frequencies and consequential bond relationships between clicks and purchases. The term consequential bond was introduced in this HPCRec system, and originates from the concept that a customer who clicks on certain items in most cases ultimately purchases an item from their list of clicks. Processing this information enhances the user-item rating matrix in terms of both the quantity and quality aspects, thereby improving recommendations. The quality of ratings is improved by capturing the level of interest in a product that has been previously purchased by a user through the record of the normalized frequency of purchase using the unit vector method. The quantity of ratings is improved by the consequential bond between clicks and purchases for sessions without purchases. Finally, the ratings for all the original unknowns are predicted based on this enriched rating matrix using the CF algorithm. The HPCRec system can provide recommendations for infrequent users, and proves that the consequential bond with normalized frequencies is more effective at predicting user interest.

An Example Application of HPCRec18 System [43]: Inputs to HPCRec system [43] are: (1) user–item rating matrix (Table 22) and the consequential table (Table 23), which shows the relationship between user clicks and purchases, and (2) the user item purchase frequency matrix (Table 24), which represents the frequency of a product purchased in the user item rating matrix. The working of the algorithm is demonstrated in the following steps.

Step 1: Normalize the purchase frequency matrix using unit vector formula. The user–item purchase frequency matrix (Table 24) is formed from Table 23, where the value represents the number of times a product was purchased by a user. The purchase frequency is then normalized to a scaled value (0 to 1) to form the normalized user–item purchase frequency matrix (Table 25) using the unit vector formula shown 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 (item1: 1; item2: 2; item3: 0; item4: 3), then the normalized purchase frequency of user 2 for 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 frequency matrix.

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

Here, $LCSR(x, y) = LCS(x, y) / \max(|x|, |y|)$ $LCS(x, y)$ is the longest common sub-sequence between sequence x and sequence y , and is computed as shown below. $LCS(X_i, Y_j) = \emptyset$ if $i = 0$ or $j = 0$; but $LCS(X_i, Y_j) = LCS(X_{i-1}, Y_{j-1}) \cup X_i$ if $X_i = Y_j$; but $LCS(X_i, Y_j) = \max(LCS(X_i, Y_{j-1}), LCS(X_{i-1}, Y_j))$ 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 the consequential table, the clickstream sequence similarity measurement (CSSM) is computed to find similar sessions with a purchase value using the longest common sub-sequence rate (LCSR), for example,

$$LCSR(\langle 3, 5, 2 \rangle, \langle 3, 5, 2, 3 \rangle) = \frac{(\langle 3, 5, 2 \rangle \cap \langle 3, 5, 2, 3 \rangle)}{\max(3, 4)} = 3/4 = 0.75.$$

As there is no purchase information for session 6 in the consequential table (Table 23), the Clickstream similarity between session 6, which is $\langle 3, 5, 2 \rangle$, and other sessions is computed, as shown in Table 26.

Table 26. Compute clickstream sequence similarity measurement(CSSM) information table.

Clicks	Purchase	CSSM
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 (Table 27) using the similarity as weight and purchases as transaction records.

Table 27. Weighted transactional purchase table.

Purchase	$\langle 2 \rangle$	$\langle 2, 3 \rangle$	$\langle 1, 2, 4 \rangle$	$\langle 2, 4, 4 \rangle$	$\langle 1 \rangle$
1	0.37	0.845	0.33	0.245	0.295

Step 4: Call the TWFI (Transaction-based Weighted Frequent Item) function. This takes a weighted transaction table in which weights are assigned to each transaction as input and returns items with weighted support within the given threshold. For example, consider a minimum weighted support = 0.1; then, we have the frequent weighted transaction table 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 as given in Table 29.

Table 29. Support for items in the 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 as shown in Table 30. 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 the weighted support using feature scaling $x' = (x - min) / (max - min)$. For the average weighted support, $max = 1.79$, $min = 0.625$, the new average weighted support for item3 is $(0.845 - 0.625) / (1.79 - 0.625) = 0.189$. 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., $(2:1), (3:0.189), (4:0.167)$). For each one of these items, if the user has not purchased it, add the weight to 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 obtain predicted ratings for all of the original unknowns. This is 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]

Summary of HSPRec19 System [6]: This work was proposed to improve the HPCRec system, which does 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 can explore and enrich the user-item matrix using sequential patterns of customer clicks and purchases in order to better capture customer behavior.

An Example Application of HSPRec19 System [6]: Given the following input data of minimum support, historical user-item purchase frequency matrix, and consequential bond, the expected output is an enriched user-item matrix for CF. The application of the HSPRec19 [6] goes through the following steps to find the required results for improved recommendation.

Consider the consequential bond of clicks and purchases (Table 32) created from click and purchase historical data and the 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)>	?

Step 1: Create a user–item purchase frequency matrix (Table 34) from Table 32, where the number indicates the number of times an item purchased by a user. For example, user 1 purchased butter twice, honey once, etc.

Table 34. 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 the GSP algorithm. The possible purchase sequential rules from frequent purchase sequences are shown in Table 35.

Table 35. Sequential rules from n-frequent sequences.

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

Step 3: Fill in the purchase information in the user–item frequency matrix using sequential purchase rules to obtain a richer matrix as in Table 36.

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 can be seen in Table 33, there is no purchase information for user 3. Thus, to find purchase information for user 3, the relationship between clicks and purchases is analyzed by considering their sequence and recommending items based on the click sequential rule for cases where the user clicks and then does not purchase anything.

Step 5: Compute the Click Purchase Pattern (CPS) similarity using the frequency and sequence of click and purchase patterns. If there is no purchase from among the with clicked item, use the recommended item.

Step 6: Assign a Click Purchase (CPS) similarity value to the purchase patterns present in the consequential bond.

Step 7: Assign weighted purchase patterns to the Weighted Frequent Purchase Pattern Miner (WFPP) and compute weights for the items present in the weighted purchase pattern using the equation below.

$$R_{item_i} = \sum_{i=1}^n CPS_{item_i} / Support_{item_i}$$

Step 8: Use the weights of the items to construct the rich user–item matrix.

Step 9: Normalize the rich user–item purchase frequency matrix (Table 37) to obtain the normalized quantitatively rich user–item matrix (Table 38) using the unit normalization function below.

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

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

In [6], user-based collaborative filtering was used to compare and evaluate the performance of the ChoiRec12, HPCRec18, and HSPRec19 recommendation systems against the traditional CF algorithm in terms of the quality of ratings predictions with respect to the Mean Absolute Error (MAE) predictive accuracy metric by varying the number of users and nearest neighbors. The 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 used to compare and evaluate the performance of the ChoiRec12 [39], HPCRec18 [43], and HSPRec19 [6] recommendation systems against the traditional CF algorithm in terms of the quality of ratings prediction with respect to the Mean Absolute Error (MAE) predictive accuracy metric by varying the 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.

Table 38. Quantitative rich 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

The performance of SP-based E-commerce recommendation systems such as ChoiRec12 [39], HPCRec18 [43], and HSPRec19 [6] was evaluated in terms of the quality of recommendations generated by varying the number of users with respect to classification accuracy measures such as the precision and recall, which evaluate the frequency with which the system makes correct/incorrect decisions. Precision is the fraction of all recommended items that are relevant, while recall is the fraction of all relevant items that are recommended. The results obtained from the experimental comparative analysis of Traditional CF, ChoiRec12 [39], HPCRec18 [43], and HSPRec19 systems conducted by [6] showed that the HSPRec19 system performed the best in comparison to the other recommendation systems, as it uses the SPM (GSP algorithm) to discover frequent historical sequential patterns and analyses clickstream behaviour to improve the consequential bond between clicks and purchases, thereby enhancing the user–item frequency matrix both quantitatively and qualitatively to generate a rich user–item matrix for CF. This results in better recommendations in terms of reduced data sparsity while improving recommendation accuracy, scalability, diversity, and novelty. Thus, of all the reviewed SP-based E-commerce recommendation systems, the HSPRec19 system performs the best in real-life application scenarios.

3. Taxonomy for Sequential Pattern-Based E-Commerce Recommendation Systems

A taxonomy for existing SP-based E-commerce recommendation systems is proposed based on the two categories discussed below.

3.1. Effect of Sequential Patterns on Recommendation Systems with Respect to Improving the Quality and Quantity of the User–Item Rating Matrix Input

The user–item rating matrix is usually sparse, poorly informative, and leads to poor recommendations [28]. In order to capture more real-life customer purchase behavior and provide historical purchase relationships between already purchased items and recommended items, the historical sequential purchase patterns of users are analyzed and integrated into the user–item rating matrix input. This has the effect of enhancing and improving rating quality (specifying the level of interest or value for already-rated items) and quantity (finding possible ratings for previously unknown ratings) using mined sequential patterns. Table 39 shows how the surveyed recommendation systems can improve the quality and quantity of the user–item rating matrix input through the use of sequential patterns and compares them to each other.

3.2. Effect of Sequential Patterns on Recommendation Systems with Respect to Handling the Problems of Sparsity, Novelty and Scalability

In academic environments, the performance evaluation of recommendation systems is dominated by simulation-based experiments using datasets consisting of historical ratings or implicit feedback. 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 a central problem in recommendation systems research. Therefore, increasing prediction accuracy is a relevant goal of research [47]. However, accuracy alone is not enough; recommending items that the user might have bought anyways is of little business value. Focusing on accuracy alone can lead to monotonous recommendations and limited discovery; thus, it is important for recommendation systems to assess multiple (possibly

competing) goals in parallel, for instance, handling data sparsity while improving novelty and scalability of the recommendation systems.

Table 39. How the surveyed recommendation systems improve the quality and quantity of the 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 algorithm 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 algorithm 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 mining with Prefix-Span algorithm used to predict purchases
Product RecSys16	Use of historical and frequency of purchases to improve rating quality	Sequential pattern mining with FreeSpan used to predict purchases
SainiRec17	Use of historical purchases to improve rating quality	Sequential pattern mining with Spade used to predict purchases
HPCRec18	Use of historical, frequency click stream, and consequential bond of purchases to improve rating quality	Analyzes session-based data mined with consequential bonds to predict purchases even for items with no ratings
HSPRec19	Use of historical, frequency click stream, and consequential bond of purchases to improve rating quality	Sequential pattern mining with GSP and consequential bond used to predict purchases even for items with no ratings

Sparsity: in practice, many commercial recommendation systems (e.g., book recommendations on 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 two million books is 20,000 books), i.e., only a few of the total number of items available in a database are rated by users. Thus, in any recommendation system, the number of ratings already obtained is usually very low 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 in providing 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 dataset 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 the performance of the surveyed recommendation systems by examining and comparing how all the surveyed algorithms handle problems such as sparsity, novelty, and scalability while improving the user–item (U-I) rating quality and quantity of recommendation systems through the use of sequential patterns. 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 within the specified range.

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

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 is used. Medium: uses SPM, and cannot integrate any other implicit user behaviors, such as clickstream data. High: uses SPM and integrates additional behavioral data such as clickstream data to enhance the user–item matrix.

Improving Novelty

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

Improving Scalability

Low: no clustering technique is used to reduce the dimensionality of the dataset. Medium: a clustering technique is used to reduce the dimensionality of the dataset. High: a clustering technique is used along with an additional dimensionality reduction technique.

Improving user–item (U-I) rating quality and quantity

Low: no historical user purchases, clickstream behavior, frequency of purchases, or other user purchase behavior is mined for integration into the U-I rating matrix Medium: minimum information, i.e., only one among user purchase behaviors such as association rules (which are not as informative as sequential patterns), historical user purchases, clickstream behavior, and frequency of purchase are incorporated into the U-I rating matrix, resulting in a less complex method of mining user purchase behavior High: more informative historical customer purchase behavior features are mined and incorporated into the U-I rating matrix, such as clickstream behavior, consequential bond information on the historical clicks and purchases of a user, historical sequential purchase behavior (sequential patterns), etc. Early hybrid recommendation systems such as ChoRec05, ChenRec09, HuangRec09, LiuRec09, and ChoiRec12 use association rule mining to improve the quality of rating input. None of these systems incorporates additional customer behavioral data, such as clickstream data or browsing history, from which implicit behavior can be extracted and used to fill in the unknown ratings. Hence, these systems are assigned a “low” level in terms of reducing data sparsity. The HSPRec19 system can achieve this to a higher extent by using SPM (GSP algorithm) to derive sequential patterns, thereby improving both the rating quality and quantity. Thus, this system is assigned a “high” level with respect to reducing data sparsity. The remaining four systems (Hybrid Model RecSys16 [40], Product RecSys16 [40], SainiRec17 [42], and HPCRec18 [43]) do not integrate any additional behavior, instead extracting sequential patterns using SPM algorithms such as PrefixSpan [33], FreeSpan [48], and SPADE [32], which results in reducing data sparsity to a “medium” level. The novelty rate is defined “low” if the items previously purchased by the target user are included in the recommendation list, as novelty accounts for the likelihood of a recommendation system providing recommendations to the user that they are not yet aware of. Thus, the novelty rate is defined “high” if known items are excluded from recommendation, while products associated with the purchased products instead used for recommendation purposes to make the customer aware of potentially related products. The dimensionality of a dataset can be reduced 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 is used by the system, its improvement in terms of scalability is specified as “low”, if a clustering technique is used to reduce the dimensionality of the dataset the improvement in terms of scalability is specified as “medium”, and if a clustering technique along with an additional dimensionality reduction technique is used by the system then the improvement in terms of scalability is specified “high”.

4. Conclusions and Future Work

Recommendation Systems open up new opportunities to retrieve personalized information on the internet, enabling users to have access to products and services which are not readily available to users of the system. Many recommendation systems neglect sequential patterns during recommendation. Thus, to verify the necessity of sequential patterns in E-commerce recommendation systems, in this review we have conducted a survey of existing SP-based E-commerce recommendation systems. In addition, we have developed a taxonomy that classifies these applications by their recommendation method as well as by performance factors such as reducing data sparsity, improving the scalability of recommendation systems, and improving the accuracy and novelty of recommendations. After performing a comparative analysis of traditional CF against few of the surveyed SP-based E-commerce recommendation systems, our results prove that the hybridization of SPM with CF by integrating sequential patterns into the user–item rating matrix input can improve recommendation quality in terms of accuracy, diversity, and novelty. Additionally, we would direct interested readers to several open research subjects that warrant future work in the area of SP-based E-commerce recommendation systems. Ideas for future work in this direction include the following aspects:

1. None of the reviewed studies has exactly measured the level of probability of purchase determined by each SP; instead, the general mid-point of 50% [6] is used as an example. Hence, more information in historical data (such as the frequency of patterns occurring together) should be used to determine the exact level of purchase probability (e.g., 0.5 to 1.0) for each SP.
2. Additional ways of incorporating clickstream sequences/patterns into the user–item rating matrix should be found, with the use of consequential bonds used to improve the quality of the input user–item rating. Moreover, additional information such as contextual data (e.g., the time of year, such as the season, month, or day of the week) should be integrated into user–item preferences.
3. Incorporating a profit or utility factor to find new patterns other than the frequent sequential patterns in historical purchase data could result in more profitable recommendations. Thus, sequential patterns with high utility should be integrated into the process of generating recommendations.
4. In real-world scenarios, items purchased by a user during a certain time period are often from multiple domains, rather than only one domain. There are many sequential dependencies between items from different domains, e.g., the purchase of 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 represent another promising research direction that could generate more accurate recommendations by leveraging information and diverse recommendations from different domains.
5. Another promising line for 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 offline approaches. Although offline evaluation has lower cost, and there is no risk of bias due to active user involvement, as in the case of online and user studies, the results mostly contradict when applied to real-life scenarios involving online and user study evaluations. Therefore, there is a need for more research on evaluation strategies to compare performance based on different measures other than accuracy and offline evaluation, such as real-time evaluations, as well as the novelty, coverage, serendipity, and diversity of recommendations.

Author Contributions: This paper was written by C.I.E. (first author) from an unpublished MSc thesis of her graduate student; the second author, H.K., reviewed and read initial versions of the paper. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Natural Science and Engineering Research Council (NSERC) of Canada under an Operating Grant (OGP 0194134) and a University of Windsor Grant. NSERC: https://www.nserc-crsng.gc.ca/OnlineServices-ServicesEnLigne/Index_eng.asp, accessed on 22 September 2023.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

Meaning of Abbreviations

Acronym	Meaning
CF	Collaborative Filtering
CS	Cosine Similarity
CFPP	CF-based Predicted Preference
CPS	Click Purchase Similarity
FSP	Frequent Sequential Pattern
FPP	Frequent Predicted Preference
GA	Genetic Algorithm
GSP	Generalized Sequential Pattern Mining
PRS	Product Recommendation System

RS/RecSys	Recommendation System
RFM	Recency, Frequency, and Monetary
SOM	Self-Organizing Map
SP	Sequential Pattern
SPM	Sequential Pattern Mining
SPAPP	Sequential Pattern Analysis Predicted Preference
SSR	Segment-based Sequential Rule

References

- Aggarwal, C.C. An introduction to recommender systems. In *Recommender Systems*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 1–28.
- Karlapalepu, H.S. A Taxonomy of Sequential Patterns Based Recommendation Systems. Master's Thesis, University of Windsor, Windsor, ON, Canada, April 2020, *unpublished*.
- Ricci, F.; Rokach, L.; Shapira, B. Recommender Systems: Techniques, Applications, and Challenges. In *Recommender Systems Handbook*; Springer Nature: Berlin/Heidelberg, Germany, 2011; pp. 1–35.
- Han, J.; Pei, J.; Tong, H. *Data Mining: Concepts and Techniques*, 4th ed.; Morgan Kaufman/Elsevier: Burlington, MA, USA, 2023.
- Agrawal, R.; Srikant, R. Mining quantitative association rules in large relational tables. In Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data-SIGMOD '96, Montreal, QC, Canada, 4–6 June 1996; pp. 1–12.
- Bhatta, R.; Ezeife, C.; Butt, M.N. Mining sequential patterns of historical purchases for e-commerce recommendation. In Proceedings of the International Conference on Big Data Analytics and Knowledge Discovery, Linz, Austria, 26–29 August 2019; pp. 57–72.
- Mabroukeh, N.R.; Ezeife, C.I. A taxonomy of sequential pattern mining algorithms. *ACM Comput. Surv.* **2010**, *43*, 3. [[CrossRef](#)]
- Mooney, C.H.; Roddick, J.F. Sequential pattern mining—Approaches and algorithms. *ACM Comput. Surv.* **2013**, *45*, 1–39. [[CrossRef](#)]
- Adomavicius, G.; Tuzhilin, A. Towards the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Trans. Knowl. Data Eng.* **2005**, *17*, 734–749. [[CrossRef](#)]
- Bokde, D.; Girase, S.; Mukhopadhyay, D. Matrix Factorization Model in Collaborative Filtering Algorithms: A Survey. *Procedia Comput. Sci.* **2015**, *49*, 136–146. [[CrossRef](#)]
- Quadrana, M.; Cremonesi, P.; Jannach, D. Sequence-aware Recommender Systems. *ACM Comput. Surv.* **2018**, *1*, 1–35.
- Burke, R. Hybrid recommender systems: Survey and experiments. *User Model. User-Adapt. Interact.* **2002**, *12*, 331–370. [[CrossRef](#)]
- Konstan, J.A.; Miller, B.N.; Maltz, D.; Herlocker, J.L.; Gordon, L.R.; Riedl, J. GroupLens: Applying collaborative filtering to Usenet news. *Commun. ACM* **1997**, *40*, 77–87. [[CrossRef](#)]
- Chun, J.; Oh, J.Y.; Kwon, S.; Kim, D. Simulating the Effectiveness of Using Association Rules for Recommendation Systems. In *Lecture Notes in Computer Science Systems Modeling and Simulation: Theory and Applications*; Springer: Berlin/Heidelberg, Germany, 2004; pp. 306–314.
- Liu, Y.; Liao, W.-k.; Choudhary, A. A two-phase algorithm for fast discovery of high utility itemsets. In Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining, Hanoi, Vietnam, 18–20 May 2005; pp. 689–695.
- Schafer, J.B.; Frankowski, D.; Herlocker, J.; Sen, S. Collaborative filtering recommender systems. In *The Adaptive Web*; Springer: Berlin/Heidelberg, Germany, 2007; pp. 291–324.
- Wei, K.; Huang, J.; Fu, S. A survey of e-commerce recommender systems. Service systems and service management. In Proceedings of the 2007 International Conference on Service Systems and Service Management, Chengdu, China, 9–11 June 2007; pp. 1–5.
- Xu, M.; Liu, F.; Xu, W. A Survey on Sequential Recommendation. In Proceedings of the 2019 6th International Conference on Information Science and Control Engineering (ICISCE), Shanghai, China, 20–22 December 2019; pp. 106–111.
- Patel, D.; Patel, F.; Chauhan, U. Recommendation Systems: Types, Applications, and Challenges. *Int. J. Comput. Digit. Syst.* **2023**, *13*, 850–868. [[CrossRef](#)]
- Zangerle, E.; Bauer, C. Evaluating Recommender Systems: Survey and Framework. *ACM Comput. Surv.* **2022**, *55*, 170. [[CrossRef](#)]
- Bernard, N.; Balog, K. MG-ShopDial: A Multi-Goal Conversational Dataset for E-Commerce. In Proceedings of the ACM SIGIR Proceedings, Taipei, Taiwan, 23–27 July 2023; pp. 2775–2788.
- Fang, H.; Zhang, D.; Shu, Y.; Guo, G. Deep Learning for Sequential Recommendation: Algorithms, Influential Factors, and Evaluations. *Acm Trans. Inf. Syst.* **2020**, *1*, 1–41. [[CrossRef](#)]
- Kerbergen, B.; Sprangers, O.; Schelter, S. Serenade-Low-Latency Session-Based Recommendation in E-Commerce at Scale. In Proceedings of the ACM SigMod Proceedings, Philadelphia, PA, USA, 12–17 June 2022; pp. 150–159.
- Kim, Y.S.; Hwangbo, H.; Lee, H.J.; Lee, W.S. Sequence aware recommenders for fashion E commerce. In *Electronic Commerce Research*; Springer Nature: Berlin, Germany, 2022; pp. 1–22. [[CrossRef](#)]
- Nasir, M.; Ezeife, C.I.; Gidado, A. Improving E-Commerce Product Recommendation using Semantic Context and Sequential Historical Purchases. *Springer Int. J. Soc. Netw. Anal. Min.* **2021**, *11*, 82. [[CrossRef](#)]
- Li, C.; Niu, X.; Luo, X.; Chen, Z.; Quan, C. A Review-Driven Neural Model for Sequential Recommendation. In Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, Macao, China, 10–16 August 2019; pp. 2866–2872.

27. Wang, S.; Hu, L.; Wang, Y.; Cao, L.; Sheng, Q.Z.; Orgun, M. Sequential Recommender Systems: Challenges, Progress and Prospects. In Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, Macao, China, 10–16 August 2019.
28. Bucklin, R.E.; Sismeiro, C. A model of web site browsing behavior estimated on clickstream data. *J. Mark. Res.* **2003**, *40*, 249–267. [[CrossRef](#)]
29. Sarwar, B.; Karypis, G.; Konstan, J.; Riedl, J. Analysis of recommendation algorithms for e-commerce. In Proceedings of the 2nd ACM Conference on Electronic Commerce-EC '00, Minneapolis, MN, USA, 17–20 October 2000; pp. 158–167.
30. Chiu, D.-Y.; Wu, Y.-H.; Chen, A.L.P. An efficient algorithm for mining frequent sequences by a new strategy without support counting. In Proceedings of the 20th International Conference on Data Engineering, Boston, MA, USA, 2 April 2004; pp. 375–386.
31. Kazienko, P.; Pilarczyk, M. Hyperlink Recommendation Based on Positive and Negative Association Rules. *New Gener. Comput.* **2008**, *26*, 227–244. [[CrossRef](#)]
32. Zaki, M.J. SPADE: An efficient algorithm for mining frequent sequences. *Mach. Learn.* **2001**, *42*, 31–60. [[CrossRef](#)]
33. Pei, J.; Han, J.; Mortazavi-Asl, B.; Pinto, H. PrefixSpan: Mining sequential patterns efficiently by prefix-projected pattern growth. In Proceedings of the International Conference on Data Engineering, Heidelberg, Germany, 2–6 April 2001; pp. 215–224.
34. Agrawal, R.; Srikant, R. Fast algorithms for mining association rules. In Proceedings of the 1994 International Conference on Very Large Data Bases (VLDB'94), Santiago de Chile, Chile, 12–15 September 1994; pp. 487–499.
35. Cho, Y.B.; Cho, Y.H.; Kim, S.H. Mining changes in customer buying behavior for collaborative recommendations. *Expert Syst. Appl.* **2005**, *28*, 359–369. [[CrossRef](#)]
36. Chen, Y.L.; Kuo, M.H.; Wu, S.Y.; Tang, K. Discovering recency, frequency, and monetary (RFM) sequential patterns from customer's purchasing data. *Electron. Commer. Res. Appl.* **2009**, *8*, 241–251. [[CrossRef](#)]
37. Huang, C.L.; Huang, W.L. Handling sequential pattern decay: Developing a two-stage collaborative recommender system. *Electron. Commer. Res. Appl.* **2009**, *8*, 117–129. [[CrossRef](#)]
38. Liu, D.; Lai, C.; Lee, W. A hybrid of sequential rules and collaborative filtering for product recommendation. *Inf. Sci.* **2009**, *179*, 3505–3519. [[CrossRef](#)]
39. Choi, K.; Yoo, D.; Kim, G.; Suh, Y. A hybrid online-product recommendation system: Combining implicit rating-based collaborative filtering and sequential pattern analysis. *Electron. Commer. Res. Appl.* **2012**, *11*, 309–317. [[CrossRef](#)]
40. Fang, Z.; Zhang, L.; Chen, K. A behavior mining based hybrid recommender system. In Proceedings of the 2016 IEEE International Conference on Big Data Analysis (ICBDA), Hangzhou, China, 12–14 March 2016.
41. Jamali, S.; Navaei, Y.D. A two-level Product Recommender for E-commerce Sites by Using Sequential Pattern Analysis. *Int. J. Integr. Eng.* **2016**, *8*, 15–24.
42. Saini, S.; Saumya, S.; Singh, J.P. Sequential purchase recommendation system for e-commerce sites. In Proceedings of the IFIP International Conference on Computer Information Systems and Industrial Management, Bialystok, Poland, 16–18 June 2017; pp. 366–375.
43. Xiao, Y.; Ezeife, C.I. E-Commerce Product Recommendation Using Historical Purchases and Clickstream Data. In Proceedings of the Big Data Analytics and Knowledge Discovery Lecture Notes in Computer Science, Regensburg, Germany, 3–6 September 2018; pp. 70–82.
44. Zhao, Q.; Bhowmick, S.S. *Association Rule Mining: A Survey*; Nanyang Technological University: Singapore, 2003; p. 135.
45. Kaymak, U. Fuzzy target selection using RFM variables. In Proceedings of the Proceedings Joint 9th IFSA World Congress and 20th NAFIPS International Conference (Cat. No. 01TH8569), Vancouver, BC, Canada, 25–28 July 2001; Volume 2, pp. 1038–1043.
46. Song, W.; Yang, K. Personalized Recommendation Based on Weighted Sequence Similarity. In *Practical Applications of Intelligent Systems*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 657–666.
47. Jannach, D.; Jugovac, M. Measuring the Business Value of Recommender Systems. *Acm Trans. Manag. Inf. Syst.* **2019**, *10*, 1–23. [[CrossRef](#)]
48. Han, J.; Pei, J.; Mortazavi-Asl, B.; Chen, Q.; Dayal, U.; Hsu, M.-C. Freespan: Frequent pattern projected sequential pattern mining. In Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Boston, MA, USA, 20–23 August 2000; ACM: New York, NY, USA, 2000; pp. 355–359.

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.