A Survey on Session-based Recommender Systems

SHOUJIN WANG, Macquarie University, Australia
LONGBING CAO, University of Technology Sydney, Australia
YAN WANG, QUAN Z. SHENG, and MEHMET A. ORGUN, Macquarie University, Australia
DEFU LIAN, University of Science and Technology of China, China

Recommender systems (RSs) have been playing an increasingly important role for informed consumption, services, and decision-making in the overloaded information era and digitized economy. In recent years, session-based recommender systems (SBRSs) have emerged as a new paradigm of RSs. Different from other RSs such as content-based RSs and collaborative filtering-based RSs that usually model long-term yet static user preferences, SBRSs aim to capture short-term but dynamic user preferences to provide more timely and accurate recommendations sensitive to the evolution of their session contexts. Although SBRSs have been intensively studied, neither unified problem statements for SBRSs nor in-depth elaboration of SBRS characteristics and challenges are available. It is also unclear to what extent SBRS challenges have been addressed and what the overall research landscape of SBRSs is. This comprehensive review of SBRSs addresses the above aspects by exploring in depth the SBRS entities (e.g., sessions), behaviours (e.g., users' clicks on items), and their properties (e.g., session length). We propose a general problem statement of SBRSs, summarize the diversified data characteristics and challenges of SBRSs, and define a taxonomy to categorize the representative SBRS research. Finally, we discuss new research opportunities in this exciting and vibrant area.

CCS Concepts: • **Information systems** → *Information systems applications*; *Data mining*; *Information retrieval*; • **Computing methodologies** → *Machine learning*; • **Applied computing** → *Electronic commerce*;

Additional Key Words and Phrases: recommender systems, session-based recommender systems, recommendations, session-based recommendations

ACM Reference format:
Shoujin Wang, Longbing Cao, Yan Wang, Quan Z. Sheng, Mehmet A. Orgun, and Defu Lian. 2021. A Survey on Session-based Recommender Systems. *ACM Comput. Surv.* 54, 7, Article 154 (July 2021), 38 pages.
https://doi.org/10.1145/3465401

1 INTRODUCTION

Recommender Systems (RSs) have evolved into a fundamental tool for making more informative, efficient and effective choices and decisions in almost every daily aspect of life, working,

This work was supported by Australian Research Council Discovery Grants (DP180102378, DP190101079, and FT190100734).

Authors' addresses: S. Wang (corresponding author), Y. Wang, Q. Z. Sheng, M. A. Orgun, Department of Computing, Macquarie University, Sydney, NSW 2109, Australia; emails: shoujinwang@foxmail.com, {yan.wang, michael.sheng, mehmet.orgun}@mq.edu.au; L. Cao, FEIT, University of Technology Sydney, Sydney, NSW, 2007, Australia; email: longbing.cao@uts.edu.au; D. Lian, School of Data Science, University of Science and Technology of China, Hefei, Anhui, 230027, China; email: liandefu@ustc.edu.cn.

0360-0300/2021/07-ART154 $15.00
https://doi.org/10.1145/3465401

business operations, study, entertaining, and socialization [124]. Their roles have become ever important in the increasingly overloaded age of digital economy where users have to make choices from usually massive and rapidly increasing contents, products and services (which are uniformly called *items*). A variety of RS research areas have emerged with great success, such as content-based RSs [2, 77], collaborative filtering-based RSs [22, 90], and hybrid RSs [7] that combine the first two.

However, those RSs tend to utilize all historical *user-item interactions* (*interactions* for short, referring to the direct or indirect user actions on items, e.g., the list of a user's clicks on items) [8] to learn each user's long-term and static preferences on items. Such a practice is often associated with an underlying assumption that all of the historical interactions of a user are equally important to her current preference. This may not be the reality in the real-world cases and there are two major reasons. First, a user's choice on items not only depends on her long-term historical preference but also depends on her short-term recent preference and the time-sensitive context (e.g., the recently viewed or purchased items). This short-term preference is embedded in the user's most recent interactions [46], which often account for a small proportion of her historical interactions. Second, a user's preference toward items tends to be dynamic rather than static, that is evolving over time.

To bridge these gaps in RSs, **Session-based Recommender Systems (SBRSs)** have emerged with increasing attention in recent years. Different from the aforementioned RSs, SBRSs learn users' preferences from the *sessions* associated and generated during the consumption process. Each *session* is composed of multiple user-item interactions that happen together in a continuous period of time, e.g., a basket of products purchased in one transaction visit, which usually lasts for several minutes to several hours. By taking each session as the basic input unit, an SBRS is able to capture both a user's short-term preference from her recent sessions and the preference dynamics reflecting the change of preferences from one session to another for more accurate and timely recommendations. In this article, with the term SBRSs, we refer to all RSs that are centered on the session data to recommend the next interaction or the next partial session (i.e., the remaining interactions) in current session, or the next session with multiple interactions (cf. Section 2.2). This definition covers those narrowly conceived SBRSs in some studies [37, 84] that recommend the next interaction in the current session only. Also, it covers both session-based and session-aware RSs discussed in Reference [84].

There are a variety of studies on SBRSs described by different terms in the literature with different settings and assumptions, targeting different application domains. For example, Hidasi et al. [38] built an SBRS on anonymous session data by assuming a strict order over the interactions (e.g., click an item or watch a movie) within each session to predict the next item to click or the next video to watch. Hu et al. [43] built another SBRS on non-anonymous session data without the order assumption inside sessions to recommend the next item to purchase. Jing et al. [49] devised an SBRS on non-anonymous session data with order assumption inside sessions for the next music or movie recommendations.

Although SBRSs are widespread in various domains and many related studies have been conducted, there are many inconsistencies in the area of SBRSs caused by the diverse descriptions, settings, assumptions and application domains. There is not a unified framework that well categorize them and there are no unified problem statements for SBRSs. More importantly, no systematic discussion is available on the unique characteristics of SBRSs including their problem and session data, the research challenges incurred by the characteristics, and the research landscape and gaps in addressing the challenges. There is not a systematic categorization of all the representative and state-of-the-art approaches for SBRSs. These gaps have limited the theoretical development and practical applications of SBRSs.

To address the above significant aspects and gaps, this article provides a comprehensive and systematic overview and survey of the field of SBRSs:

- We provide a unified framework to categorize the studies on SBRSs, which can reduce the confusions and inconsistent views in the field of SBRSs.
- For the first time, our work proposes a unified problem statement of SBRSs, where an SBRS is built on top of formal concepts: user, item, action, interaction, and session.
- We provide a comprehensive overview of the unique characteristics of session data as well as the challenges of SBRSs incurred by them. To the best of our knowledge, this is the first such description.
- A systematic classification and comparison of SBRS approaches are made to provide an overall view on how the challenges have been addressed and what progress has been achieved in the SBRS area.
- Each class of approaches for SBRSs have been briefly introduced with key technical details to provide an in-depth understanding of the progress achieved for SBRSs.
- Lastly, open issues and prospects for the SBRS research are discussed.

2 RELATED WORK

There are a variety of studies on not only SBRSs but also **Sequential Recommender Systems (SRSs)** [119]. SRS is an area closely relevant to but different from SBRSs. Even in the area of SBRSs, there are many different sub-areas, e.g., next-interaction (e.g., purchase an item) recommendations, next-session (e.g., basket) recommendations. As a result, a variety of corresponding specific works described by different terms exist in the literature, including session-based recommendations, next-item/song recommendations, next-basket recommendations, session-based recommender systems, sequential recommender systems, and so on. Although quite similar, these works are usually applicable for different scenarios, with different settings and assumptions, belonging to different areas, i.e., SBRSs or SRSs, or some of the above sub-areas. It is not uncommon that these superficially similar but actually different works not only cause confusions between SBRSs and SRSs, but also lead to significant inconsistencies within the area of SBRSs. Below, we first clarify the concepts and differences between SBRSs and SRSs, then provide a framework to categorize the relevant SBRS studies, and clarify the difference between this article and the related work.

2.1 SBRSs vs. SRSs

SBRSs and SRSs are built on session data and sequence data, respectively, while they are often mixed up by some readers. So it is necessary to first clarify the difference between session data and sequence data. A *session* is a list of interactions with a clear *boundary*, while the interactions may be chronologically *ordered* (in ordered sessions) [38, 85], or *unordered* (in unordered sessions) [43, 117]. A boundary refers to the starting-ending interaction pair to start and end a specific session in a transaction event. An ordered (unordered) session refers to a session in which the interactions are (not) chronologically ordered. The session data from a given user usually consists of multiple sessions happening at different time and separated by multiple boundaries with non-identical time intervals between sessions (cf. Figure 1(a)). A *sequence* is a list of historical elements (e.g., item IDs) with clear *order*. The sequence data from a given user often contains a single sequence with only one boundary for it (cf. Figure 1(b)). In most sequence data, the timestamps are used to sort the elements inside a sequence only while no explicit time intervals are included and considered [84]. A boundary usually indicates co-occurrence-based dependencies [8] over the interactions or elements within it. Co-occurrence-based dependencies constitute the foundation of SBRSs, especially for those built on unordered session data. Order implies clear sequential

Table 1. A Comparison between Session Data and Sequence Data

Data type		Boundary	Order	Time interval	Main relations embedded
Session data	Unordered session	Multiple	No	Non-identical	Co-occurrence-based dependencies
	Ordered session	Multiple	Yes	Non-identical	Co-occurrence-based dependencies and sequential dependencies
Sequence data		Single	Yes	Not included	Sequential dependencies

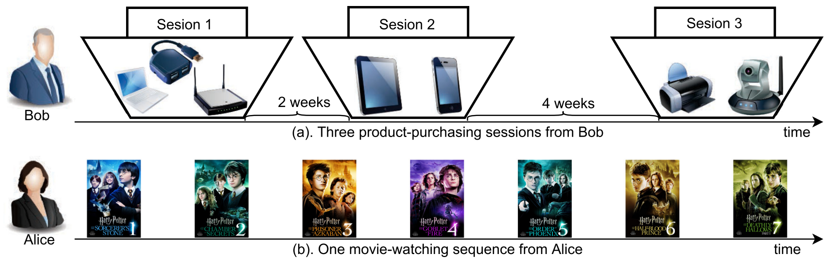
Sesion 1
Sesion 2
Sesion 3
Bob
2 weeks
4 weeks
(a). Three product-purchasing sessions from Bob
time
Alice
(b). One movie-watching sequence from Alice
time

Fig. 1. Session data vs. sequence data.

dependencies among the interactions or elements inside a session or a sequence. We show the difference between the session data and the sequence data from a certain user in Table 1.

An SBRS aims to predict either the unknown part (e.g., an item or a batch of items) of a session given the known part, or the future session (e.g., the next-basket) given the historical sessions via learning the intra- or inter-session dependencies. Such dependencies usually largely rely on the co-occurrence of interactions inside a session and they may be sequential or non-sequential [43]. In principle, an SBRS does not necessarily rely on the order information inside sessions, but for ordered sessions, the naturally existing sequential dependencies can be utilized for recommendations. In comparison, an SRS predicts the successive elements given a sequence of historical ones by learning the sequential dependencies among them. Several survey papers focus particularly on SRSs, including sequence-aware recommender systems [84], deep learning for sequential recommendations [24] and sequential recommender systems [119]. In this survey, we particularly focus on the area of SBRSs with an emphasis on the unique characteristics of session data together with the corresponding challenges they have brought to SBRSs, and the representative and the state-of-the-art approaches for SBRSs.

2.2 A Framework for Organizing SBRS Work

The variety of existing work on SBRSs can be generally categorized into three sub-areas fitting a unified categorization framework to reduce the aforementioned inconsistencies and confusion. According to the difference on the recommendation tasks, the sub-areas include *next interaction recommendation*, *next partial-session recommendation*, and *next session recommendation*. Given the known part (i.e., happened interactions) of a session, next interaction recommendation aims to recommend the next possible interaction in the current session by mainly modeling intra-session dependencies. It is usually simplified to predict the next item to interact, e.g., a product to click or purchase. Given the known part of a session, next-partial session recommendation aims to recommend all the remaining interactions to complete the current session, e.g., to predict all the subsequent items to complete a basket given the purchased items in it, by mainly modelling intra-session dependencies. Although less studied, this sub-area is even more practical in real-world cases, since a user often does not interact with only one item in the next, but multiple items until she finish the whole session. Given the historical sessions, next session recommendation aims to recommend the next session, e.g., next basket, by mainly modeling inter-session dependencies.

Table 2. A Comparison of Different Sub-Areas in SBRSs

Sub-area	**Input**	**Output**	**Typical research topic**
Next interaction recommendation	Mainly known part of the current session	Next interaction (item)	Next item recommendation, next song/movie recommendation, next POI recommendation, next web page recommendation, next news recommendation, and so on.
Next partial-session recommendation	Mainly known part of the current session	Subsequent part of the session	Next items recommendation, session/basket completion
Next session recommendation	Historical sessions	Next session	Next basket recommendation, next bundle recommendation, and so on.

Sometimes, inter-session dependencies are also incorporated into the first two sub-areas to improve recommendation performance. A comparison of these sub-areas is presented in Table 2.

2.3 Related Surveys

Although many studies have been done in the area of SBRSs, to the best of our knowledge, there are limited comprehensive and systematic reviews to shape this vibrant area and position the existing works as well as the current progress. Although some works have attempted to comprehensively evaluate and compare the performance of existing SBRS algorithms, we have not found any studies that systematically formalize this research field, or comprehensively analyze the unique characteristics of session data and the critical challenges faced by SBRSs. Let alone to provide an in-depth summary of current efforts or detail the open research issues present in the field.

Several surveys focus on the conventional RSs or the emerging deep learning-based RSs. Shi et al. [94] comprehensively analyzed the recently proposed algorithms for collaborative filtering-based RSs and discussed the future challenges in the area. Lops et al. [61] provided an overview of content-based RSs by summarizing the corresponding representative and the state-of-the-art algorithms and discussing the future trends. Burke et al. [7] surveyed the landscape of hybrid RSs. Zhang et al. [143] provided a comprehensive review of recent research efforts on deep learning-based RSs. In addition, there are also several surveys on SRSs. For instance, Quadrana et al. [84] conducted a comprehensive survey on sequence-aware RSs from various aspects, including the recommendation task, the algorithms and evaluations; Fang et al. [24] provided a comprehensive survey on deep learning-based sequential recommendations from the aspects of algorithms, influential factors, and evaluations; and Wang et al. [119] conducted a brief review on the challenges and progress of SRSs.

However, there is a lack of an extensive review on SBRSs. Although SBRSs have been partially discussed in Reference [84], this work mainly focused on sequence-aware RSs and only discussed a small proportion of works on SBRSs built on ordered session data, while ignoring SBRSs based on unordered sessions. More importantly, a variety of recent progress has not been covered. To the best of our knowledge, there are only three formally published review papers (include a short one) particularly focusing on SBRSs. To be specific, Ludewig et al. [63] provided a systematic performance comparison of a number of SBRS algorithms, including **Recurrent Neural Network- (RNN)** based approaches, factorized Markov chain-based approaches, and nearest neighbour-based approaches. Later, Ludewig et al. [64] compared the performance of four neural network-based approaches and five conventional approaches based on rule learning or nearest neighbour, which was further extended into a comprehensive empirical study of SBRS algorithms where 12 algorithmic approaches were compared in terms of their recommendation performance [65]. While focused on the experimental perspective, these studies only covered a few approaches but did not provide a comprehensive review and analysis from the theoretical perspective.

Table 3. Main Notations in SBRSs

Notation	Description	Notation	Description
U	A set of users, i.e., $U = \{u_1, u_2, \ldots, u_{\|U\|}\}$	S	A set of sessions
u_k	The kth user of U, $1 \leq k \leq \|U\|$	$s_{j'}$	The j'−th session of S, which is a list of interactions
V	A set of items, i.e., $V = \{v_1, v_2, \ldots, v_{\|V\|}\}$	c	A session context
v_i	The ith item of V, $1 \leq i \leq \|V\|$	l	A list of interactions
A	A set of actions, i.e., $A = \{a_1, a_2, \ldots, a_{\|A\|}\}$	$\boldsymbol{v}_i$	The representation[1] of v_i
$a_{k'}$	The k'th action of A, corresponding to the k'th type of action, e.g., purchase or click	$\boldsymbol{o}_{i'}$	The representation of $o_{i'}$
O	A set of interactions, i.e., $O = \{o_1, o_2, \ldots, o_{\|O\|}\}$	$\boldsymbol{e}_c$	The representation of c
$o_{i'}$	The i'th interaction of O, which is a tuple of a user, an item and an action	$\boldsymbol{h}_t$	The hidden state[2] at time step t

[1]A representation is often specified as a latent vector; [2]A hidden state is a latent vector from an RNN (cf. Section 8.1.1).

Given the rising popularity and potential of SBRSs and the steady flow of novel research contributions in this area, a comprehensive survey will be of high scientific and practical value. This survey seeks to provide a comprehensive review of the current research on SBRSs to bridge these gaps with the aim of supporting further development of SBRSs. As the first attempt, this article explores the field of SBRSs with an emphasis on the problem statement, analysis of challenges, review of progress and discussion of future prospects.

3 SBRS PROBLEM STATEMENT

An RS can be seen as a system [8, 9], which consists of multiple basic entities including *users*, *items* and their *behaviours*, e.g., user-item interactions. These basic entities and behaviours form the core constituents of a *session*, which is the core entity in an SBRS. Therefore, we first introduce the definitions and properties of these entities and behaviours, and then define the SBRS problem based on them. These definitions and properties will be further used for the characterization and categorization of SBRSs, and so on. The main notations are listed in Table 3.

3.1 User and User Properties

A *user* in an SBRS is the subject who takes actions, e.g., clicks, purchases, on items, e.g., products, and receives the recommendation results. Let u denote a user and each user is associated with a unique ID and a set of attributes to describe her, e.g., the gender of a user, and it has multiple values, e.g., male and female. The attributes of a user could affect the actions she takes on items and further affect the corresponding sessions. For instance, a boy may watch more action movies, leading to more action movies in his watching sessions, while a girl may like to watch more love-story movies. In addition to the explicit attributes that can be obviously observed, some implicit attributes that reflect the user's internal states, e.g., her moods and intentions, may also have a significant impact on her actions. All the users together form the user set, namely $U = \{u_1, u_2, \ldots, u_{|U|}\}$. Note that the user information of a session may not be always available for two reasons: (1) It is not recorded due to the privacy protection and (2) some users do not log in when interacting with online platforms like amazon.com. Consequently, the session becomes anonymous.

3.2 Item and Item Properties

An *item* in an SBRS is an entity to be recommended, such as a product, e.g., a book, or a service, e.g., a course. Let v denote an item, which is associated with a unique ID and a set of attributes to provide the description information of the item, such as the category and the price of an item. All the items in a dataset form the item set, namely $V = \{v_1, v_2, \ldots, v_{|V|}\}$.

Usually, items are different in different domains. For instance, in the news recommendation domain, an item is a news article posted on a news website, e.g., a report on artificial intelligence on abc.com; in the e-commerce domain, an item is a product for sale, e.g., an earphone on amazon.com, and in the service industry, an item is a specific service, e.g., the course "Machine Learning" provided by Coursera (https://www.coursera.org/).

3.3 Action and Action Properties

An *action* is often taken by a user on an item in a session, e.g., clicking an item. Let a denote an action, which is associated with a unique ID and a set of attributes to provide its property information e.g., the type of the action, and has multiple values, e.g., click, view, and purchase. Note that some actions may not be associated with specific items, e.g., a search action or a catalog navigation action. But they may still provide useful information to an SBRS as discussed in Reference [84].

3.4 Interaction and Interaction Properties

Interaction is the most basic unit in sessions. Let o denote an *interaction*, which is a ternary tuple consisting of a user u, an item v and the action a taken by u on v, namely $o = \langle u, v, a \rangle$. In the case where the user information is not available, the interaction become anonymous, i.e., $o = \langle v, a \rangle$. Moreover, in the case, where there is only one type of actions, e.g., clicks, the interaction o can be further simplified as $o = \langle v \rangle$, namely it only consists of an item. All the interactions together form the interaction set O.

3.5 Session and Session Properties

Session is an important entity in an SBRS. Let s denote a session, which is a non-empty bounded list of interactions generated in a period of continuous time that may be connected with some user- (e.g., user ID) or session-specific (e.g., a session-ID or a cookie) information, i.e., $s = \{o_1, o_2, \ldots, o_{|s|}\}$. Note that here we use the concept "list" instead of "set" to indicate that there may be duplicated interactions in one session. For example, a user listens to a song for multiple times in a listening session. Each session is associated with a set of attributes, e.g., the duration of s, which have multiple corresponding values, e.g., 20 minutes or 40 minutes. Some other important attributes of a session include the time and the day when the session happens. Next, we discuss five important properties of sessions that may have a great impact on SBRSs.

Property 1: session length. The length of a session is defined as the total number of interactions contained in it. This is a basic property of sessions, which is taken as one of the statistical indicators of experiment data in most literature [43, 117]. Sessions of different lengths may bring different challenges for SBRSs and thus lead to different recommendation performance. The session characteristics related to session length together with the corresponding challenges for building SBRSs are discussed in detail in Section 4.1.

Property 2: internal order. The internal order of a session refers to the order over interactions within it. Usually, there are different kinds of order flexibility inside different sessions, i.e., no order, flexible order and order. The existence of internal order leads to the sequential dependencies within sessions that can be used for recommendations. The session characteristics related to internal order and the corresponding challenges for building SBRSs are discussed in detail in Section 4.2.

Property 3: action type. In the real world, some sessions contain only one type of actions, e.g., purchase, while other sessions may contain multiple types of actions, e.g., click, purchase (cf. Figure 2(a)). The dependencies over different types of actions are often different. For instance, the items that are clicked together in a session may be similar or competitive while the items purchased together in one session may be complementary. Therefore, the number of action types in a session determines whether the intra-session dependencies are homogeneous (based on a single

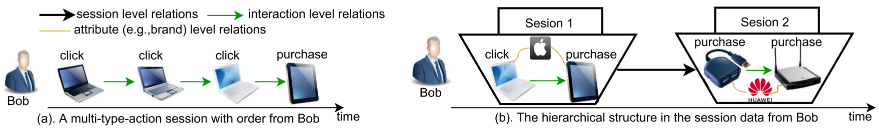
session level relations
interaction level relations
attribute (e.g.,brand) level relations
click
click
click
purchase
Bob
(a). A multi-type-action session with order from Bob
time
Sesion 1
Sesion 2
click
purchase
purchase
purchase
Bob
(b). The hierarchical structure in the session data from Bob
time

Fig. 2. Toy examples for a typical session and session data with hierarchical structure.

type of actions) or heterogeneous (based on multi-type actions), which is important for accurate recommendations. The session characteristics related to action type as well as the corresponding challenges for building SBRSs are discussed in detail in Section 4.3.

Property 4: user information. User information of a session mainly refers to the IDs of the users in the session, and sometimes user attributes are also included. In this article, the property of user information refers to the availability of user information in a session. In the real word, the user information of sessions is given in some cases, while it is not available in other cases (cf. Section 3.1) [38, 107, 129]. User information plays an important role to connect sessions from the same user happening at different time and thus its availability determines the possibility to model the long-term personalized preference across multiple sessions for a specific user. In practice, SBRSs were initially proposed to handle those anonymous sessions where user information is not available [37]. The session characteristics together with the corresponding challenges for building SBRSs are discussed in detail in Section 4.4.

Property 5: session-data structure. Session-data structure refers to the session-related hierarchical structure consisting of multiple levels [85, 116], which intrinsically exists in some session data. For example, the attribute level consists of the attributes of entities (e.g., users, items) in an interaction, the interaction level consists of interactions in each session, and the session level consists of multiple historical sessions from the current user (cf. Figure 2(b)). The interaction level is necessary for a session, while the other levels depend on the specific session data. This is because either the attribute information or the historical session information may not be available in all session data. Usually, the number of levels included in a session dataset determines the information volume that can be used for recommendations. The session characteristics related to session-data structure as well as the corresponding challenges for building SBRSs are discussed in detail in Section 4.5.

3.6 The SBRS Problem

From the system perspective, we formalize SBRS by first illustrating its input, output and work mechanism, and then presenting the problem formalization.

Input. The basic input of an SBRS is the partially known session information that is used for recommendations [14]. According to the specific scenarios, the basic input has three cases: (1) the known part of the current session (i.e., a list of happened interactions), which is the input of the SBRSs modelling intra-session dependencies only for next interaction (item), or next partial-session recommendation (cf. Section 2.2); (2) the list of known historical sessions, which is the input of the SBRSs that mainly model inter-session dependencies for next session (e.g., a basket) recommendation; and (3) the combination of the first two, which is the input of the SBRSs modelling both intra- and inter-session dependencies for the recommendation of next interaction, or next partial-session.

Most of the existing SBRSs mainly take the IDs of users (if not anonymous), items and actions as the input while ignoring their attribute information [38, 58, 105]. Only a minority of studies [39, 46, 103, 117] assume the attribute information is available and take it as part of the input. In a specific case, the input part of the current session or historical sessions may be anonymous or non-anonymous, ordered or unordered, with single- or multi-type actions. From our observation, most of the existing SBRSs assume the input sessions are ordered and with single-type actions. In an SBRS, the input is usually formalized as a *session context* (also called a context in this article) conditioned on which the recommendation is performed.

Output. The goal of an SBRS is to make recommendations according to a given session context, i.e., the known session information. Accordingly, the output of an SBRS is a predicted interaction or a predicted list of interactions that happen subsequently in the current session, or the predicted next session that happens following the given historical sessions (cf. Section 2.2). In an SBRS, the user information of a session is either not available and thus is not predictable (in anonymous sessions) or given by default (in non-anonymous sessions), so it is usually unnecessary to predict the user information. Consequently, each interaction in the output only contains an item and the corresponding action taken on it [102], e.g., an item to purchase or click. Moreover, in most cases where SBRSs are built on single-type-action sessions, the only action type is given by default, e.g., purchase or click, and thus each interaction in the output is further simplified to an item [43, 49]. According to the specific sub-areas, there are three cases for the output (cf. Section 2.2): (1) in next interaction recommendation, the output is a list of alternative interactions (items) [49, 102], ranked by best match as the next interaction (item) in the session; (2) in next partial-session recommendation, the output is a list of interactions (items) to complete the current session; and (3) in next session recommendation, the output is a list of complementary interactions (items) to form the next session [120], e.g., to purchase a basket of complementary products (e.g., milk, bread) to achieve a unified goal (e.g., breakfast). In the last two cases, according to whether the session is ordered or not, the interactions in the list may be ordered or unordered accordingly.

Work mechanism. In principle, the work mechanism of an SBRS is to first learn the comprehensive dependencies among interactions within or/and between sessions and then to utilize the learned dependencies to guide the prediction of the subsequent interactions or sessions to accomplish the recommendation task. Next, we illustrate it in detail in the following problem formalization.

Problem formalization. There are many different types of SBRSs with their own specific characteristics and formalization. Here we give an abstract-level formalization that is suitable for different SBRSs. Let $l = \{o_1, \ldots, o_j, \ldots, o_n\}$ be a list of n interactions, each of which is composed of an item and the corresponding action taken on it. Recall that in SBRSs built on single-type-action sessions, each interaction is simplified to an item and thus the interaction list l is simplified to an item list l_v, i.e., $l_v = \{v_1, \ldots, v_j, \ldots, v_n\} (v_j \in V)$. L is the set of lists, which contains all the possible interaction lists derived from the candidate item set V and action set A. c is the input, i.e., a session context, consisting of all the session information used for the recommendation. All the session contexts together form the session context set C. Similarly to sequence-aware RS [84], let f be a utility function to return the utility score of a candidate interaction list l for a given session context c. An SBRS is to select the recommended interaction list $\hat{l} \in L$ by maximizing the utility score conditioned on the given session context c, i.e.,

$$\hat{l} = arg\ max\ f(c, l), c \in C, l \in L, \tag{1}$$

where the utility function can be specified to multiple forms, e.g., likelihood, conditional probability. The utility function is employed on the interaction list to optimize the candidate list as a whole rather than optimising a single candidate interaction (item). This makes the formalization not only cover all the aforementioned three cases for the output of SBRSs, but it also makes it possible to characterize and evaluate the list as a whole from multiple aspects, e.g., the novelty or diversity of the interactions (items) in the list [84].

4 CHARACTERISTICS AND CHALLENGES

SBRSs are built on session data, and different types of session data are usually associated with different characteristics, which essentially bring different challenges to build SBRSs. Similar to understanding data characteristics and challenges in other data-driven research [10], a deep understanding of the intrinsic characteristics of session data, and the challenges in modelling session data for building SBRSs is fundamental for designing an appropriate SBRS. Therefore, in this section, we systematically illustrate and summarize a variety of characteristics of session data as well as the corresponding challenges caused by each of them in building SBRSs.

According to each of the session properties introduced in Section 3.5, sessions can be divided into different types. For instance, according to the property "session length," sessions can be divided into long sessions, medium sessions and short sessions. Next, we first present different types of sessions categorized by each of their properties, and then discuss the characteristics and challenges associated with each type of sessions.

4.1 Characteristics and Challenges Related to Session Length

According to session length, sessions can be roughly categorized into three types: long sessions, medium sessions and short sessions, while the specific definitions for long, medium and short sessions may vary upon the specific data sets.

Long sessions. A long session contains relatively more interactions, e.g., more than 10. In general, with more interactions, long sessions can provide more contextual information for more accurate recommendations. However, due to the uncertainty of user behaviours, a long session is more likely to contain random interactions [43] that are irrelevant to other interactions in it. This brings noisy information and thus reduces the performance of recommendations [118, 119]. Therefore, the first challenge for SBRSs built on long sessions is *how to effectively reduce the noisy information from the irrelevant interactions*. In addition, there are usually more complex dependencies embedded in a long session, e.g., *long-range dependencies* [141] between two interactions that are far from each other in a session or *high-order dependencies* [119] across multiple interactions in a session. Consequently, another challenge for SBRSs built on long sessions is *how to effectively learn complex dependencies for better recommendation performance.*

Medium sessions. Medium sessions usually contain a medium number of interactions, e.g., 4 to 9. From our observations on session data generated from transaction records in e-commerce industry, medium sessions are the most common case [123]. Compared with long and short sessions, a medium session is less likely to contain too many irrelevant interactions while it usually contains the necessary contextual information for **Session-Based Recommendation (SBR)**. Although relatively less challenging, building SBRSs on medium sessions still faces a general challenge, i.e., *how to effectively extract the relevant and accurate contextual information for accurate recommendations.*

Short sessions. A short session consists of quite limited interactions, e.g., usually less than 4, leading to limited information available for recommendation. For example, in an offline anonymous session consisting of two interactions, the only contextual information that can be utilized to recommend the second interaction (item) is the first interaction in the session. An extreme case is to recommend the first interaction of a session. Consequently, the challenge for SBRSs built on short sessions is *how to effectively make recommendations with quite limited contextual information.*

4.2 Characteristics and Challenges Related to Internal Order

According to whether there is an order over interactions inside a session or not, sessions can be roughly divided into unordered sessions, ordered sessions and flexible-ordered sessions.

Unordered sessions. An unordered session contains interactions without any chronological order between them, namely, whether an interaction happens earlier or later in the session makes no difference [43]. For example, the shopping sessions are sometimes unordered, since users may pick up a basket of items (e.g., {bread, milk, eggs}) without following an explicit order [117]. In unordered sessions, the dependencies among the interactions are based on their co-occurrence rather than the sequences of them, and thus the generally utilized sequence models are not applicable. Compared with sequential dependencies, co-occurrence-based dependencies are usually relatively weak and fuzzy, which are more difficult to learn. Furthermore, most of co-occurrence-based dependencies among interactions are collective dependencies [101, 141], i.e., several contextual interactions in a session collaboratively lead to the occurrence of the next interaction, which are even harder to capture. Consequently, the challenge for SBRSs built on unordered sessions is *how to effectively learn the relatively weak and fuzzy dependencies among interactions, especially those collective dependencies.*

Ordered sessions. An ordered session contains multiple interactions with strict order, and usually strong sequential dependencies exist among them. For example, the session composed of a sequence of online courses taken by a user is often ordered, since some prerequisite courses must be taken first to gain prior-knowledge for the subsequent ones. Although it is relatively easy to learn the strong sequential dependencies within ordered sessions, it is challenging *to effectively learn the cascaded long-term sequential dependencies that decay gradually with time in long ordered sessions.*

Flexibly ordered sessions. A flexibly ordered session is neither totally unordered nor totally ordered, i.e., some parts of the session are ordered while others are not [101]. For example, a tourist generates a session of check-ins at airport, hotel, shopping center, bar, and attraction successively. In the session, the airport, hotel, and attraction are actually sequentially dependent, while the shopping center and bar are randomly inserted without any order. Therefore, the complex dependencies inside flexibly ordered sessions must be carefully considered and precisely learned for accurate recommendation. Consequently, the challenge for SBRSs built on flexibly ordered sessions comes from *how to effectively learn the complex and mixed dependencies, i.e., sequential dependencies among ordered interactions and non-sequential dependencies among unordered ones.*

4.3 Characteristics and Challenges Related to Action Type

According to the number of action types included in a session, sessions can be divided into single-type-action sessions and multi-type-action sessions.

Single-type-action sessions. A single-type-action session includes one type of actions only, e.g., clicks of items, and thus only one type of dependencies comes from the same type of actions, which is relatively easy to learn.

Multi-type-action sessions. A multi-type-action session includes more than one types of actions [54], leading to multiple types of interactions. For example, in a real-world online shopping session, a user usually first clicks several items for comparison and then purchases one or more items. Thus, there are complex dependencies inside a multi-type-action session [66]. Specifically, dependencies not only exist over the interactions from the same type (e.g., clicks of items), but also exist over interactions from different types (e.g., clicks and purchases). As a result, a big challenge for SBRSs built on multi-type-action sessions is *how to effectively and accurately learn both the intra- and inter-action type dependencies for accurate recommendations.*

4.4 Characteristics and Challenges Related to User Information

According to whether the user information is available or not, sessions can be divided into non-anonymous sessions and anonymous sessions.

Non-anonymous sessions. A non-anonymous sessions contains non-anonymous interactions with the associated user information, which enables the connections of different sessions generated by the same user at different time. This makes it possible to learn the user's long-term preference as well as its evolution across sessions. However, due to the relative long time-span and preference dynamics, it is quite challenging *to precisely learn the personalized long-term preference over multiple non-anonymous sessions.*

Anonymous sessions. In anonymous sessions, due to the lack of user information to connect multiple sessions generated by the same user, it is nearly impossible to collect the prior historical sessions for the current session. As a result, only the contextual information from the current session can be used for recommendations. Therefore, it is challenging to *precisely capture the user's personalized preference with limited contextual information to provide accurate recommendation.*

4.5 Characteristics and Challenges Related to Session-data Structure

According to the number of levels of structures, session data can be roughly divided into single-level session data and multi-level session data (cf. Section 3.5). Specifically, the interaction level naturally exists in any session data, and thus single-level session data particularly refers to the data including the interaction level only. Multi-level session data refers to the data including the attribute level or/and the session level in addition to the interaction level.

Single-level session data. A single-level session dataset is usually a set of anonymous sessions where each consists of several interactions without attribute information or historical session information. In such a case, only single-level dependencies, i.e., the inter-interaction dependencies within sessions, can be utilized for recommendations. Hence, due to the lack of auxiliary information from other levels, SBRSs built on single-level session data may easily suffer from the cold-start or data sparsity issue [66]. This leads to the challenge of *how to overcome the cold-start and sparsity issues for accurate recommendations when only the inter-interaction dependencies are available.*

Multi-level session data. Multi-level session data involves a hierarchical structure of at least two levels, i.e., the interaction level plus attribute level and/or session level. In this case, both the dependencies within each level and across different levels would affect the subsequent recommendations. For example, the categories (the attribute level) of several items may have impact on whether these items would be bought together (the interaction level) in one session. Consequently, *how to comprehensively learn the intra- and inter-level dependencies for effective and accurate recommendations* becomes a key challenge for SBRSs built on multi-level session data.

4.6 A Summary of Characteristics and Challenges

In this subsection, we provide a summary of session characteristics and the corresponding challenges. To be specific, a comparison of carefully selected representative and state-of-the-art works on SBRSs regarding their targeted session types, basic model and the application domain is presented in Table 4. Recall that each type of sessions have their own characteristics and challenges, therefore, the session type in Table 4 actually reflects the corresponding session characteristics and challenges for SBRSs that are targeted by each work. As a result, Table 4 provides a comprehensive overall view on existing works from multiple perspectives. For instance, the second row on the left side in Table 4 means that the association rule-based SBRS in Reference [70] mainly targets the Flexible-Ordered (FO), Single-Type-action (ST), Anonymous (A), and Single-Level (SL) sessions with associate rule mining approach for web page recommendations.

5 CLASSIFICATION AND COMPARISON OF SBRS APPROACHES

To provide an overall view of the achieved progress in addressing the challenges introduced in Section 4, we first classify the approaches for SBRSs from the technical perspective, i.e., the involved method or model, in Section 5.1, and then compare different classes of approaches in Section 5.2.

Table 4. Comparison of Representative Works Regarding Targeted Session Type, Basic Model, and Application Domain

Work	Session type	Model	Domain	Work	Session type	Model	Domain
An SBRS based on association rules [70]	FO, ST, A, SL	ARD	Web page	Attention-gated recurrent network (IARN) [78]	L, O, ST, NA, SL	RNN, ATT	Video, movie
Access Pattern Approach (APA) [93]	UO, ST, NA, SL	FPM	Music	KNN-GRU4Rec [45]	O, ST, A, SL	KNN, RNN	Item
Personalized sequential pattern [134]	L, O, ST, NA, SL	SPM	Item	Temporal deep semantic structured model (TDSSM) [98]	O, ST, NA, ML	MLP, RNN	News
Item/session KNN [45, 63]	O/UO, ST, A, SL	NN	Item	List-wise deep neural network [127]	UO, MT, A, SL	MLP	Item
Sequence and Time Aware Neighbourhood (STAN) [29]	O, ST, A, SL	NN	Item	DeepPredict [46]	UO, MT, NA, ML	MLP	Fashion
Temporal-Item-Frequency-based User (TIFU)-KNN [41]	UO, ST, NA, SL	NN	Item	ConvolutionAl Sequence Embedding Recommendation Model (CASER) [101]	FO, ST, NA, SL	CNN	Movie, POI
Page rank and Markov model [21]	O, ST, A, SL	MC	Web page	3D Convolutional Neural Network (3D CNN) [104]	O, ST, A, ML	CNN	Item
FPMC [87]	O, ST, NA, ML	MC, MF	Item	Hierarchical Temporal Convolutional Networks (HierTCN) [137]	L, O, MT, NA, ML	TCN	Item
Dynamic emission and transition model [50]	O, ST, NA, SL	HMM	Music, tweet	SR-GNN [129]	O, ST, A, SL	GNN	Item
PRME [25]	O, ST, NA, SL	MC, ME	POI	GC-SAN [132]	O, ST, A, SL	GNN, ATT	Item
PME [130]	O, ST, NA, SL	MC, ME	Music	TAGNN [138]	O, ST, A, SL	GNN, ATT	Item
LME [12]	O, ST, NA, SL	MC, ME	Music	Multi relational GNN for Session-based Prediction (MGNN-SPred) [125]	O, MT, A, SL	GNN	Item
FPMC-Localized Regions (LR) [16]	O, ST, NA, SL	MF	POI	Full GNN based on Broadly Connected Session graph (FGNN-BCS) [81]	O, ST, A, ML	GNN	Item
Co-factorization (CoFactor) [56]	UO, ST, NA, SL	MF	Item	Full GNN based on Weighted Graph ATtention layer (FGNN-WGAT) [82]	O, ST, A, SL	GNN	Item
Category aware POI recommendation model [59]	O, ST, NA, SL	MF	POI	Hierarchical Attentive Transaction Embedding (HATE) [116]	UO, ST, NA, ML	ATT, DR	Item
Session-based Wide-In-Wide-Out (SWIWO) networks [43]	UO, ST, NA, SL	DR	Item	Dynamic Co-attention Network for SBR (DCN-SR) [14]	L, O & UO, MT, NA, ML	ATT, RNN	Item
Network-based Transaction Embedding Model (NTEM) [117]	UO, ST, A, ML	DR	Item	Encoder-Decoder with attention (EDRec) [62]	O, MT, NA, SL	ED, RNN, ATT	Item
Meta-Prod2Vec [107]	O, ST, A, ML	DR	Music	NARM [52]	L, O, ST, A, SL	ED, RNN ATT	Item
Music Embedding Model (MEM) [110]	O, ST, NA, ML	DR	Music	STAMP model [58]	UO, ST, A, SL	ATT, MLP	Item
Attention-based Transaction Embedding Model (ATEM) [118]	UO, ST, A, SL	DR, ATT	Item	Streaming Session-based Recommendation Machine (SSRM) [31]	O, ST, NA, ML	ATT, MF, RNN	Music, POI
Hierarchical Representation Model (HRM) [114]	UO, ST, NA, ML	DR	Item	Sequential Hierarchical Attention Network (SHAN) [136]	FO, ST, NA, ML	ATT, DR	Item
GRU4Rec [38]	O, ST, A, SL	RNN	Item	Memory-Augmented Neural Network (MANN) [15]	FO, ST, NA, ML	MN	Item
GRU4Rec-BPR [37]	O, ST, A, SL	RNN	Item	Memory Augmented Neural model (MAN) [67]	FO, ST, A, SL	MN	Item
Improved RNN [99]	O, ST, A, SL	RNN	Item	Hierarchical Memory Networks (HMN) [96]	O, ST, A, SL	MN, CNN	Item
Hierarchical RNN (HRNN) [85]	L, O, ST, NA, ML	RNN	Item	Collaborative Session-based Recommendation Machine (CSRM) [112]	O, ST, A, ML	MN, ED	Item
User-based RNN [20]	O, ST, NA, SL	RNN, ATT	Movie, music	Multi-temporal-range Mixture Model (M3) [100]	L, FO, ST, A, SL	MM	Movie
DREAM [139]	UO, ST, NA, ML	RNN	Item	MCPRN [123]	FO, ST, A, SL	MM, RNN	Item
Recurrent Latent Variable network for SBR (ReLaVaR) [11]	O, ST, A, SL	RNN, VI	Item	Intention2basket [121]	FO, ST, A, ML	MM, RNN	Item
RNN-latent cross [5]	O, ST, NA, ML	RNN	Video	VRM [126]	O, ST, A, SL	GM, RNN	Item
Parallel RNN (P-RNN) [39]	O, ST, A, ML	RNN	Video, item	VASER [150]	O, ST, A, SL	GM, RNN	Item
Neural survival recommender [49]	O, ST, NA, SL	RNN	Music, movie	Convolutional Generative Network (NextItNet) [141]	O, ST, A, SL	GM, CNN	Item, music
Recurrent Recommender Networks (RRN) [128]	O, ST, NA, SL	RNN	Video	LIRD [147]	O, MT, NA, SL	RL	Item
Session-aware recommendations with neural network [105]	L, O, MT, A, SL	RNN	Item	DeepPage [146]	O, MT, NA, SL	RL, RNN	Item

Session type S: short, M: medium, L: long; O: ordered, UO: unordered, FO: flexible ordered; ST: single-type-action, MT: multi-type-action; A: anonymous, NA: non-anonymous; SL: single-level, ML: multi-level.
Model ARD: association rule discovery, FPM: frequent pattern mining, SPM: sequential pattern mining, NN: nearest neighbour, MC: Markov chain, MF: matrix-factorization, VI: variational inference, HMM: hidden markov model, ME: metric embedding, DR: distributed representation, RNN: recurrent neural network, CNN: convolution neural network, TCN: temporal convolution network, MLP: multi-layer perceptron, GNN: graph neural network, ATT: attention, MN: memory network, MM: mixture model, GM: generative model, RL: reinforcement learning, ED: encoder-decoder.

[1]Most of the works do not target sessions of a specific length, only a few ones are experimented on long sessions and thus marked "L."

5.1 A Classification of SBRS Approaches

Inspired by [119], we present the taxonomy of approaches for SBRS in Figure 3. According to the employed technique, three super-classes of approaches for SBRSs are identified from the literature, i.e., *conventional SBRS approaches, latent representation approaches*, and *deep neural network*

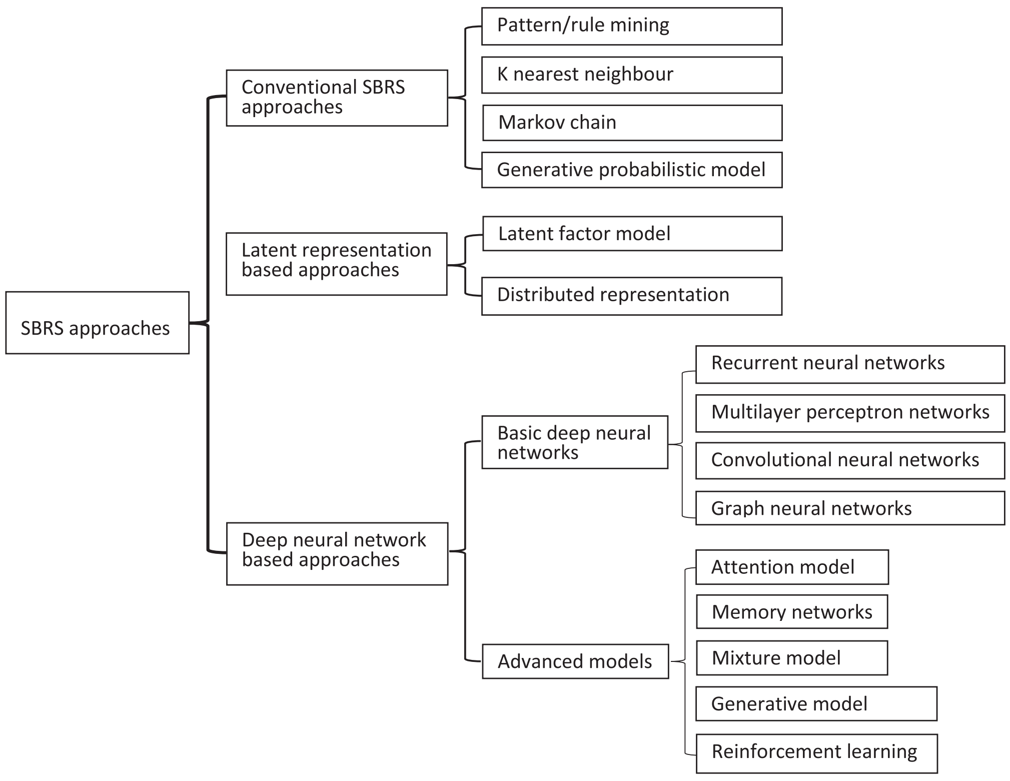
SBRS approaches
Conventional SBRS approaches
Pattern/rule mining
K nearest neighbour
Markov chain
Generative probabilistic model
Latent representation based approaches
Latent factor model
Distributed representation
Deep neural network based approaches
Basic deep neural networks
Recurrent neural networks
Multilayer perceptron networks
Convolutional neural networks
Graph neural networks
Advanced models
Attention model
Memory networks
Mixture model
Generative model
Reinforcement learning

Fig. 3. The categorization of SBRS approaches.

approaches. These three super-classes can be further divided into seven classes in total while each super-class contains multiple classes. To be specific, conventional SBRS approaches contain four classes: *pattern/rule-based approaches*, ***K Nearest Neighbour- (KNN)*** *based approaches*, *Markov chain-based approaches*, and *generative probabilistic model-based approaches*; latent representation approaches contain two classes: *latent factor model* and *distributed representation*; and deep neural network approaches contain two classes: *basic deep neural networks* and *advanced models*. In addition, the class of basic deep neural networks contains four sub-classes while each corresponds to one basic deep neural network architecture, namely, *recurrent neural networks*, ***multi-perceptron layer (MLP)*** *networks, convolution neural networks* and *graph neural neural networks*. Similarly, the class of advanced model contains five sub-classes while each corresponds to one type of models that are usually utilized for building SBRSs, i.e., *attention models, memory networks, mixture models, generative models* and *reinforcement learning*. Consequently, the existing approaches for SBRSs are classified into three super-classes, eight classes, plus nine sub-classes. As a result, 15 atomic classes of SBRS approaches are obtained, i.e., six classes from conventional SBRS approaches and latent representation approaches, and nine sub-classes from deep neural network approaches. In addition to approaches based on a single technique/model, there are some hybrid approaches that combine multiple techniques/models, e.g., an approach for next-basket recommendations combines Markov chain model and latent factor model [87]. Next, first, a comparison of different classes of approaches will be presented in Section 5.2, and then each super-class of approaches will be reviewed in Sections 6, 7, and 8, respectively.

5.2 A Comparison of Different Classes of Approaches

Generally speaking, conventional SBRS approaches are relatively simple, straightforward, and easy to understand and implement. Although simple, they are effective in some cases, especially on those simple datasets where the dependencies within or between sessions are obvious and easy to model and capture. Particularly, in a study by Ludewig et al. [64], KNN-based approaches, e.g., session-KNN, have achieved superior recommendation accuracies even compared with some deep neural network-based approaches, e.g., GRU4Rec, in much less running time on some e-commerce datasets including RETAIL, DIGI.[3] In contrast, deep neural network-based approaches are usually relatively complex, involve complicated and multi-layer network architecture and often require extensive computing. They are generally believed to be more powerful to comprehensively model and capture the complex dependencies, e.g., long-term or high-order dependencies, embedded in complex datasets, e.g., imbalanced or sparse datasets, for more accurate SBR [143]. The superiority of deep neural network-based approaches has been verified by a variety of works in recent years, e.g., References [38, 137, 139]. In general, latent representation-based approaches are a bit more complicated than conventional approaches but less complicated than deep neural network-based approaches. Unlike deep neural network-based approaches, they usually do not involve a deep network architectures, leading to relatively low computation cost. However, benefiting from their efficient and effective representation learning, they sometimes perform very well. In some studies [58, 118], latent representation-based approaches can outperform not only some conventional approaches, e.g., Markov chain-based approaches [87], but also some deep neural network-based ones, e.g., RNN-based approaches [38].

As introduced in Section 3.6, the work mechanism of SBRSs is to learn the comprehensive dependencies to guide the subsequent recommendations. Therefore, learning dependencies in session data is the key computation task in an SBRS. In addition, recall that most of the challenges in SBRSs can be abstracted to learn the various types of dependencies, e.g., high-order dependencies, embedded in different types of session data, e.g., long sessions, as illustrated in Section 4. Therefore, to have a better understanding of how each class of approaches can benefit completing the key computation task and addressing the main challenges in the field of SBRSs, Table 5 compares all the 15 atomic classes of approaches regarding the type of dependencies they can learn. For example, the fourth row in Table 5 means that Markov chain-based approaches mainly capture sequential, short-term, first-order and pointwise dependencies in session data for recommendations. First-order dependency and pointwise dependency refer to the dependency between any two adjacent interactions and that between any pair of interactions, respectively.

In addition, a statistic on the number of publications in each atomic class is presented in Figure 4. This result was achieved by manually retrieving and counting the publications on SBRS using Google scholar on 20 March, 2021. We first used the typical keywords "session, recommendation," "next item/basket/POI/song/news/video recommendation" for searching and then manually counted those relevant publications only.

6 CONVENTIONAL SBRS APPROACHES

Conventional approaches for SBRSs utilize the conventional data mining or machine learning techniques, to capture the dependencies embedded in session data for session-based recommendations. Next, we introduce each of the four classes of conventional approaches, respectively.

[3]https://www.dropbox.com/sh/n281js5mgsvao6s/AADQbYxSFVPCun5DfwtsSxeda?dl=0.

Table 5. A Comparison of Learned Dependencies by Different Classes of Approaches

Approach	Sequential or non-sequential	Short- or long-term	First or high-order	Pointwise or collective
Pattern/rule mining	Both	Both	Both	Both
K nearest neighbour	Mainly non-sequential	Both	Mainly first-order	Both
Markov chain	Sequential	Short-term	First-order	Pointwise
Generative probabilistic model	Sequential	Long-term	Higher-order	Collective
Latent factor model	Sequential	Short-term	First-order	Pointwise
Distributed representation	Mainly non-sequential	Both	Mainly first-order	Collective
Recurrent neural networks	Sequential	Long-term	High-order	Pointwise
Multilayer perceptron networks	Non-sequential	Both	First-order	Collective
Convolutional neural networks	Mainly sequential	Both	Mainly first-order	Collective
Graph neural networks	Both	Both	High-order	Pointwise
Attention models	Mainly non-sequential	Both	First-order	Mainly pointwise
Memory networks	Non-sequential	Both	First-order	Pointwise
Mixture models	Both	Both	Both	Both
Generative models	Either	Either	Either	Either
Reinforcement learning	Sequential	Both	High-order	Pointwise

[1]Non-sequential and sequential dependencies are learned by frequent pattern mining and sequential pattern mining, respectively.
[2]Item-KNN and session-KNN mainly models pointwise and collective dependencies, respectively.
[3]The learned dependencies mainly depend on the employed encoder for encoding the input of the generation model.

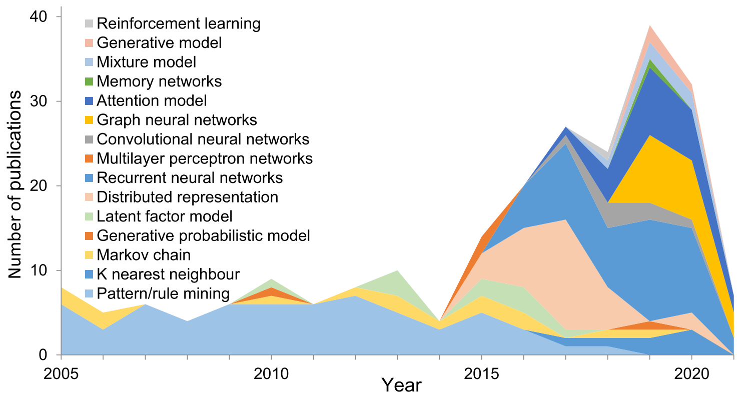

Fig. 4. Number of publications on each class of SBRS per year.

6.1 Pattern/Rule Mining-based SBRSs

Generally speaking, there are two types of pattern/rule mining-based approaches for SBRS: (1) *frequent pattern/association rule mining-based approaches*, which mine the association rules over different interactions within *unordered sessions* to guide the subsequent recommendations, and (2) *sequential pattern mining-based approaches*, which mine the sequential patterns over sequences of sessions or interactions within *ordered sessions* to guide the subsequent recommendations. This class of approaches can handle single-type-action sessions only in which all the actions are the same in a dataset, so each interaction in a given session is simplified into an item.

6.1.1 *Frequent Pattern/Association Rule Mining-based Approaches.*

Frequent pattern/association rule mining-based SBRSs mainly contain three steps: (1) frequent pattern or association rule mining, (2) session matching, and (3) recommendation generation. To be specific, given an item set V and the corresponding session set S over V, first, a set of frequent patterns $FP = \{p_1, p_2, \ldots, p_{|FP|}\}$ are mined by using pattern mining algorithms like FP-Tree [33]. Then, given a partial session $\hat{s}$ (e.g., a list of chosen items in one session), if an item $\hat{v} \in V \setminus \hat{s}$ exists so that $\hat{s} \cup \{\hat{v}\} \in FP$, then $\hat{v}$ is a candidate item for recommendations. Finally, if the conditional probability $P(\hat{v}|\hat{s})$ is greater than a predefined confidence threshold, then $\hat{v}$ is added into the recommendation list [70, 115].

Besides the aforementioned basic framework, there are many variants. For instance, to consider the different significance of different web pages and thus to recommend more useful ones, several methods [27, 133] utilized the page-view duration to weight the significance of each page and then incorporated such weight into association rule mining to build *weighted association rule-based SBRSs.* With regard to the application domain, except for the traditional shopping basket-based product recommendations, frequent pattern/association rule-based SBRSs are also commonly applied in web page recommendations [71], music recommendations [93], and so on.

6.1.2 *Sequential Pattern Mining-based Approaches.*

Following a representative work [134] falling into this sub-class, here we introduce a typical sequential pattern mining-based SBRS built on session level (cf. Section 3.5) for next session recommendations. For sequential pattern-based mining SBRSs built on the interaction level for next interaction recommendations, please refer to Reference [72]. Similarly to frequent pattern/association rule mining-based approaches, sequential pattern mining-based SBRSs also contain three steps: (1) sequential pattern mining, (2) sequence matching, and (3) recommendation generation. Specifically, given a sequence set $Q = \{q_1, q_2, \ldots, q_{|Q|}\}$ where $q = \{s_1, s_2, \ldots, s_{|q|}\}$ is a sequence of sessions from the same user ordered according to the timestamp, first, a set of sequential patterns $SP = \{p_1, p_2, \ldots, p_{|SP|}\}$ is mined on Q. Then, given a user u's sequence $q_u = \{s_1, s_2, \ldots, s_g\}$, for any sequential pattern $p \in SP$, if the last session s_g of q_u belongs to p, i.e., $p = \{s_1, s_2, ..., s_g, s_r ...\}$, then p is a relevant pattern for this specific recommendation and the items after s_g in p, like items in s_r, are candidate items. For each candidate item $\hat{v}$, its support is the sum of the support of all relevant patterns:

$$supp(\hat{v}) = \sum_{s_g \in q_u, s_g \in p, \hat{v} \in s_r, s_r \in p, p \in SP} supp(p). \tag{2}$$

Finally, those candidate items with the top support values are recommended to user u.

Except for the basic framework described above, there are various extensions for sequential pattern mining-based SBRSs. A typical example is to utilize the user-related weighted sequential pattern mining for personalized recommendations, where each sequence is assigned a weight based on its similarity to those past sequences of the target user [97]. Another extension is to build a hybrid RS by combining sequential pattern mining and collaborative filtering to consider both the users' dynamic individual patterns and their general preference [18, 57]. Regarding the application domain, shopping basket-based product recommendations [134] and web page recommendations [72] are two typical applications of sequential pattern-based SBRSs.

6.2 K Nearest Neighbour-based SBRSs

KNN-based approaches for SBRS are proven to be simple but effective [63]. In principle, a KNN-based SBRS first finds out the K interactions or sessions that are most similar to the current interaction or session, respectively, from the session data. Then, it calculates a score for each candidate interaction based on the similarity to indicate its relevance to the current interaction as the guidance of recommendations. For the same reason as mentioned in the first paragraph in Section 6.1, each interaction is simplified as an item in this class of approaches. According to whether the similarity is actually calculated between items or sessions, KNN-based approaches for SBRSs can be divided into *item-KNN* and *session-KNN*.

6.2.1 *Item-KNN.*

Given the current session context, an item-KNN based SBRS recommends those K items most similar to the current item in terms of their co-occurrence in other sessions as the next choice. Technically, each item is encoded into a binary vector where each element indicates whether the item occurs (set to "1") in a specific session or not (set to "0"). Consequently, the similarity between items can be calculated on their vectors with a certain similarity measure, like cosine similarity [63].

6.2.2 *Session-KNN.*

Given the current session context c, a session-KNN-based SBRS first calculates the similarity between c and all other sessions to find the set $N(c)$ of its K neighbour sessions, and then calculates the score of each candidate item $\hat{v}$ w.r.t. c based on the similarity:

$$score(\hat{v}) = \sum_{s_{nb} \in N(c)} sim(c, s_{nb}) \cdot 1_{s_{nb}}(\hat{v}), \tag{3}$$

where sim is a kind of similarity measures and $1_{s_{nb}}(\hat{v})$ is an indicator function that returns 1 if $\hat{v}$ occurs in s_{nb} and 0 otherwise.

Compared with item-KNN, session-KNN considers the whole session context rather than just the current item in the session context, and thus can capture more information for more accurate recommendations. Other similar works include an improved session-KNN that takes into account the readily available sequential and temporal information from sessions [29], and a hybrid approach that combines session-KNN and GRU4Rec (i.e., an RNN-based SBRS) using a weighted combination scheme [45]. In addition, a user-KNN approach built on users' session information was also proposed for next-basket recommendation [41].

6.3 Markov Chain-based SBRSs

Markov chain-based SBRSs adopt Markov chains to model the transitions over interactions within or between sessions to predict the probable next interaction(s) or session given a session context [92]. According to whether the transition probabilities are calculated based on explicit observations or latent space, Markov chain-based approaches can be roughly divided into *basic Markov chain-based approaches* and *latent Markov embedding-based approaches*.

6.3.1 *Basic Markov Chain-based Approaches.*

A basic Markov chain-based SBRS usually contains four steps: (1) calculating the transition probabilities over a sequence of interactions, (2) predicting the transition paths over interactions, (3) matching the session context to the predicted paths, and (4) making recommendations based on the matching result [21]. Note that, in most cases, the interactions here are simplified to items.

To be specific, a Markov chain model is defined as a set of tuples $\{ST, \boldsymbol{P_t}, P_0\}$, where ST is the state space including all the distinct interactions, $\boldsymbol{P_t}$ is the $m * m$ one-step transition probability matrix between m distinct interactions, and P_0 is the initial probability of each state in ST. First, the first-order transitional probability from interaction o_i to o_j is defined as:

$$P_t(i,j) = P(o_i \rightarrow o_j) = \frac{freq(o_i \rightarrow o_j)}{\sum_{o_t} freq(o_i \rightarrow o_t)}. \tag{4}$$

Second, a transition path, e.g., $\{o_1 \rightarrow o_2 \rightarrow o_3\}$, is predicted by estimating its probability by using the first-order Markov chain model:

$$P(o_1 \rightarrow o_2 \rightarrow o_3) = P(o_1) * P(o_2|o_1) * P(o_3|o_2). \tag{5}$$

Then, given a session context consisting of a sequence of interactions, the paths with high probabilities are chosen as the reference paths. Finally, if the session context occurs in a reference path, those items occurring after it in this path are put into the recommendation list.

Except for the basic Markov chain-based SBRS introduced above, there are many variants. For example, Zhang et al. [144] combined first- and second-order Markov model together to make more accurate web page recommendations. Le et al. [50] developed a hidden Markov model-based probabilistic model for next item recommendations. Rendle et al. [87] factorized the transition probability matrix to estimate those unobserved transitions among interactions.

6.3.2 Latent Markov Embedding-based Approaches. Different from the basic Markov chain-based SBRSs that calculate the transition probabilities based on the explicit observations directly, **Latent Markov Embedding- (LME)** based SBRSs first embed the Markov chains into an Euclidean space and then calculate the transition probabilities between interactions based on their Euclidean distance [12]. In this way, they can derive the unobserved transitions and thus solve the data sparsity issue in limited observed data. Formally, each interaction o is represented as a vector $\boldsymbol{o}$ in a d-dimensional Euclidean space, and the transition probability $P(o_i \rightarrow o_j)$ is assumed to be negatively related to the Euclidean distance $||\boldsymbol{o}_i - \boldsymbol{o}_j||_2$ between o_i and o_j. Accordingly, the probability of a transition path $pa = \{o_1 \rightarrow o_2 \rightarrow, \ldots, \rightarrow o_{|pa|}\}$ can be defined based on Markov model:

$$P(\{o_1 \rightarrow o_2 \rightarrow, \ldots, \rightarrow o_{|pa|}\}) = \prod_{i=2}^{|pa|} P(o_{i-1} \rightarrow o_i) = \prod_{i=2}^{|pa|} \frac{e^{-||\boldsymbol{o}_i - \boldsymbol{o}_{i-1}||_2^2}}{\sum_{o_t} e^{-||\boldsymbol{o}_t - \boldsymbol{o}_{i-1}||_2^2}}. \tag{6}$$

To generate personalized recommendations, Wu et al. [130] proposed a **Personalized Markov Embedding (PME)** model that maps both users and items into an Euclidean space where the user-item distance and item-item distance reflect the corresponding pairwise relationship. Further, **Personalized Ranking Metric Embedding (PRME)** was proposed to first project each item into a low-dimensional Euclidean latent space, and then use the Metric Embedding algorithm to effectively compute transitions between items in a Markov chain model. Intuitively, the Euclidean distances measure the probabilities of transitions [25].

6.4 Generative Probabilistic Model-based SBRSs

Generative probabilistic model-based approaches generally first infer the latent taxonomy (e.g., topics or genres) of items (e.g., songs) in sessions and then learn the transitions among these latent taxonomies within or between sessions. Afterwards, they predict the next latent taxonomy using the learned transitions. Finally, they further predict specific items as the next item conditional on the predicted latent taxonomy of items. Usually, the latent topic model is utilized to infer the latent taxonomies and the transitions among them. Representative studies include music recommendation based on latent topic sequential patterns [34], and playlist generation with statistical models on music-listening sessions [149].

6.5 Comparison of Conventional SBRS Approaches

After providing the main idea and key technical details of each class of conventional approaches for SBRSs, we present a comparison and summary of those approaches in this subsection. Specifically, in Table 6, we compare these three classes of approaches in terms of their applicable scenarios, i.e., for which type of session data that an approach is suitable, pros, cons and the typical works. An empirical comparison on prediction accuracy of the first three classes of conventional SBRS approaches was conducted on seven datasets from domains including retail, music and news. The result shows KNN-based approaches especially the session-KNN achieve superior performance [48].

Table 6. A Comparison of Different Classes of Conventional Approaches for SBRSs

Approach	Applicable scenario	Pros	Cons	Typical work
Pattern/rule mining based SBRSs	Simple, balanced and dense, ordered or unordered sessions	Intuitive, simple and effective on session data where dependencies are easy to learn	Information loss, cannot handle complex data (e.g., imbalanced or sparse data)	[27],[70],[71], [72],[93],[134]
KNN-based SBRSs	Simple, ordered or unordered sessions	Intuitive, simple and effective, quick response	Information loss, hard to select K, limited ability for complex sessions (e.g., noisy sessions)	[29],[41],[45], [63]
Markov chain-based SBRSs	Short and ordered sessions with short-term and low-order dependencies	Good at modelling short-term and low-order sequential dependencies	Usually ignore long-term and higher-order dependencies, the rigid order assumption is too strong	[12],[21],[25], [50],[87],[130], [144]
Generative probabilistic SBRSs	Ordered sessions with high-order dependencies	Good at modelling high-order and collective dependencies	Computation cost is relatively high	[34],[149]

7 LATENT REPRESENTATION APPROACHES FOR SBRSs

Latent representation approaches for SBRSs first build a low-dimensional latent representation for each interaction within sessions with shallow models. The learned informative representations encode the dependencies between these interactions and then will be utilized for the subsequent session-based recommendations. According to the utilized techniques, latent representation approaches can be roughly classified into latent factor model-based approaches and distributed representation-based approaches.

7.1 Latent Factor Model-based SBRSs

Latent factor model-based SBRSs first adopt factorization models, e.g., matrix factorization, to factorize the observed transition matrix over interactions (items) into their latent representations, and then utilize the resultant latent representations to estimate the unobserved transitions for the subsequent session-based recommendations. To be specific, first, a transition tensor $\mathcal{B}^{|U|\times|O|\times|O|}$ can be built using the observed session data, where each entry $b_{k,i,j}$ indicates the transition probability from interaction o_i to o_j under user u_k. Then, a general linear factorization model, e.g., Tucker Decomposition, is used to factorize $\mathcal{B}$:

$$\hat{\mathcal{B}} = Co \times \boldsymbol{U} \times \boldsymbol{O}_i \times \boldsymbol{O}_j, \tag{7}$$

where Co is a core tensor, $\boldsymbol{U}$ is the latent representation matrix for users while $\boldsymbol{O}_j$ and $\boldsymbol{O}_k$ are the latent representation matrix for the last interactions and the current interactions, respectively.

To alleviate the negative effect of the sparse transitions observed for $\mathcal{B}$, a special case of Canonical Decomposition [4] is used to transfer Equation (7) into the modelling of pairwise interactions:

$$\hat{b}_{k,i,j} = < \boldsymbol{u}_k, \boldsymbol{o}_i > + < \boldsymbol{o}_i, \boldsymbol{o}_j > + < \boldsymbol{u}_k, \boldsymbol{o}_j >, \tag{8}$$

where $\boldsymbol{u}_k$, $\boldsymbol{o}_i$, and $\boldsymbol{o}_j$ are the latent representation vector of user u_k, the last interaction o_i and the current interaction o_j, respectively [87]. Here, interactions are usually simplified to items.

In addition to the above defined latent factor model-based SBRS, i.e., **Factorized Personalized Markov Chain (FPMC)** model, there are many other variants. For instance, Cheng et al. [16] extended FPMC into FPMC-LR by adding a constraint to limit user movements into a localized region to make it more consistent with the real-world tourism cases for next POI recommendations. A co-factorization model, CoFactor, was proposed to jointly decompose the user-item interaction matrix and the item-item co-occurrence matrix with shared latent factors for items to capture both the users' individual preference and the item transition patterns [56]. Some other similar works

Table 7. A Comparison of Different Classes of Latent Representation Approaches for SBRSs

Approach	Applicable scenario	Pros	Cons	Typical work
Latent factor model	Dense, ordered session data	Relatively simple and effective	Suffer from data sparsity, cannot capture higher-order and long-term dependencies	[16],[55],[56], [59],[87],[92]
Distributed representation	Unordered session data	Simple and efficient, strong encoding capability	Hard to model ordered or heterogeneous sessions (e.g., noisy sessions)	[30],[43],[53],[109], [114],[117],[118]

[55, 59] utilize the matrix factorization model to learn the transitions of preferences from one location category to another to provide location recommendations.

7.2 Distributed Representation-based SBRSs

Distributed representation-based SBRSs generally learn the distributed representations of interactions (usually specified to items, sometimes users are also incorporated) with a shallow neural network structure to map each interaction into a low-dimensional latent space. In most cases, the shallow neural network structure is similar to Skip-gram model [79] or CBOW model [69] in the natural language processing domain. As a result, the intra- or inter-session dependencies are encoded into the distributed representations, which are then used for session-based recommendations.

Specifically, a shallow neural network embeds a user u_k and an item v_i into a latent distributional vector, respectively, using the logistic function $\delta(\cdot)$ for nonlinear transformation [43]:

$$\boldsymbol{u}_k = \delta(\boldsymbol{W}^u_{:,k}), \tag{9}$$

$$\boldsymbol{v}_i = \delta(\boldsymbol{W}^v_{:,i}), \tag{10}$$

where $\boldsymbol{W}^u \in \mathbb{R}^{d\times|U|}$ and $\boldsymbol{W}^v \in \mathbb{R}^{d\times|V|}$ are the user and item embedding matrices, respectively, and the kth column of $\boldsymbol{W}^u$ corresponds to user u_k.

In addition to the basic latent representation-based SBRS introduced above, there are a number of variants. Wang et al. [117] designed a shallow network to embed the ID and features of each item simultaneously to build a compound item representation to tackle the cold-start item issues; some similar works include [107, 110]. To attentively learn the relevance scales of different interactions in the session context w.r.t. the next choice, attention mechanism is incorporated into the representation learning process [118]. In other related works [109, 114], a hierarchical representation of each basket is learned for next-basket recommendations.

7.3 Comparison of Latent Representation based SBRS Approaches

After providing the main idea and the basic technical details of each class of latent representation approaches for SBRSs, we present a comparison and summary of these approaches. Specifically, in Table 7, we compare the two classes of approaches in terms of their applicable scenarios, i.e., for which kind of session data that an approach is suitable, pros, cons and the typical works.

8 DEEP NEURAL NETWORK APPROACHES FOR SBRSs

Deep neural network approaches for SBRSs mainly take advantage of the powerful capabilities of deep neural networks in modelling the complex intra- and inter-session dependencies for recommendations. According to the utilized basic framework, deep neural network approaches can be roughly divided into basic deep neural network approaches, each of which involves one type of a basic neural network architecture, e.g., RNN, and advanced models, each of which involves a certain advanced mechanism or model, e.g., attention model.

8.1 Basic Deep Neural Network-based SBRSs

According to the utilized network architecture, basic deep neural network approaches can be divided into RNN-based approaches, MLP-based approaches, **Convolutional Neural Networks-(CNN)** based approaches and **Graph Neural Networks- (GNN)** based approaches.

8.1.1 Recurrent Neural Networks. Benefiting from their intrinsic advantages for modeling sequential dependencies, RNN-based approaches dominate deep neural network approaches for SBRSs. This is because the order assumption has been applied to the interactions in a majority of session datasets in the literature. Particularly, an RNN-based SBRS first models each ordered session context as a sequence of interactions within the context. In such a way, it takes the last hidden state of the RNN modeling the context as the context representation. Then, the RNN-based SBRS takes the context representation as the input to predict the next interaction to complete the recommendation task. In those RNN-based SBRSs where the inter-session dependencies are considered, the representation of a sequence of historical sessions is first learned in a similar way and then incorporated for recommendations.

We introduce a representative RNN-based SBRS called GRU4Rec that is built on **Gated Recurrent Units (GRU)** [38], as an example to illustrate the work mechanism of RNN-based SBRSs. To be specific, an RNN is built to model the session context consisting of a sequence of interactions. First, the embedding $\boldsymbol{o}_t$ of the tth interaction o_t in the context is taken as the input of the tth time step of the RNN. Then, an RNN unit, i.e., GRU, is used to update the hidden state $\boldsymbol{h}_t$ at the tth time step by absorbing information from both the last hidden state $\boldsymbol{h}_{t-1}$ and the current candidate state $\hat{\boldsymbol{h}}_t$ by using an update gate $\boldsymbol{z}_t$,

$$\boldsymbol{h}_t = (1 - \boldsymbol{z}_t)\boldsymbol{h}_{t-1} + \boldsymbol{z}_t\hat{\boldsymbol{h}}_t, \tag{11}$$

where $\boldsymbol{z}_t$ and $\hat{\boldsymbol{h}}_t$ are computed by Equations (12) and (13) given below, respectively,

$$\boldsymbol{z}_t = \sigma(\boldsymbol{W}_z\boldsymbol{o}_t + \boldsymbol{X}_z\boldsymbol{h}_{t-1}), \tag{12}$$

$$\hat{\boldsymbol{h}}_t = tanh(\boldsymbol{W}_h\boldsymbol{o}_t + \boldsymbol{X}_h(\boldsymbol{r}_t \odot \boldsymbol{h}_{t-1})), \tag{13}$$

where $\odot$ denotes Hadamard product and the reset gate $\boldsymbol{r}_t$ is given below:

$$\boldsymbol{r}_t = \sigma(\boldsymbol{W}_r\boldsymbol{o}_t + \boldsymbol{X}_r\boldsymbol{h}_{t-1}), \tag{14}$$

where σ is the activation function that can be specified to be sigmoid function. $\boldsymbol{W}$ and $\boldsymbol{X}$ are the corresponding weighting matrices.

In this way, a session context c composed of $|c|$ interactions can be modeled by an RNN with $|c|$ units. Finally, the hidden state $\boldsymbol{h}_{|c|}$ from the last time step is used as the representation $\boldsymbol{e}_c$ of c for the prediction of the next interaction [38].

In addition to the basic GRU4Rec, there are also many variants. To improve GRU4Rec, Tan et al. [99] adopted data augmentation via sequence preprocessing and embedding dropout to enhance the training process and reduce overfitting, respectively. Quadrana et al. [85] further improved GRU4Rec by proposing a hierarchical RNN model to capture both intra- and inter-session dependencies for more reliable next item(s) recommendations. Specifically, a two-level GRU-based RNN is designed: the session-level GRU models the sequence of items purchased within each session and generates recommendations for next item(s), while the user-level GRU models the cross-session information transfer and then provides personalized information to the session-level GRU by initializing its hidden state. Another similar work is Inter-Intra RNN proposed by Ruocco et al. [88]. In Reference [20], the authors designed a unique user-based GRU model that incorporates user characteristic to generalize personalized next item recommendations. Furthermore, there are also

RNN-based SBRSs built on basic RNN units, for example, the **Dynamic REcurrent bAsket Model (DREAM)** [139] learns a dynamic representation of a user at each time step using an RNN built on basic RNN units for next basket recommendations.

There are also other variants that incorporate (1) variational inference into RNN to handle the uncertainty in sparse session data and simultaneously enhance the model's scalability on large real-world datasets for recommendations [11, 19]; (2) side information like item features and contextual factors like time and location into RNN to improve the recommendation performance [5, 39]; (3) time decay or attention mechanism into RNN to discriminate the intra-session dependencies and thus achieve more precise recommendations [6, 78]; and (4) traditional models like factorization machines or neighbourhood models to make up the drawbacks of RNN-only models [45, 105]. There are other similar RNN-based SBRSs [28, 37, 49, 128].

8.1.2 *MLP networks.* MLP-based approaches are usually applied to learn an optimized combination of different representations to form a compound representation of session context for the subsequent recommendations. Different from RNN, MLP is mainly suitable for unordered session data due to the lack of capability to model sequence data. Specifically, in the work of Wu et al. [127], an MLP layer is utilized to connect the representations of different parts of a session context to export a unified and compound representation $\boldsymbol{e}_c$ for context c:

$$\boldsymbol{e}_c = \sigma(\boldsymbol{W}_c \boldsymbol{e}_{c_c} + \boldsymbol{W}_v \boldsymbol{e}_{c_v}), \tag{15}$$

where $\boldsymbol{e}_{c_c}$, $\boldsymbol{e}_{c_v}$ are the representation of the sub session context containing "click" actions and the sub session context containing "view" actions, respectively. $\boldsymbol{W}_c$ and $\boldsymbol{W}_v$ are the corresponding weight matrices to fully connect each of the representations to the hidden layer of MLP.

In addition, Jannach et al. [46] applied MLP to learn an optimized combination of different factors like "reminders," "item popularity," and "discount" as a compound session-based feature for next-item recommendations. Song et al. [98] employed an MLP layer to combine both a user's long-term static and short-term temporal preferences for making more accurate next-item recommendations.

8.1.3 *Convolutional Neural Networks.* CNN are another good choice for SBRSs for two reasons: (1) They relax the rigid order assumption over interactions within sessions, which makes the model more robust, and (2) they have high capabilities in learning local features from a certain area and relationships between different areas in a session to effectively capture the union-level collective dependencies embedded in session data. In principle, a CNN-based SBRS first utilizes the filtering and pooling operations to better learn an informative representation for each session context and then uses the learned representation for the subsequent recommendations [140]. To be specific, given a session context c consisting of $|c|$ interactions, an embedding matrix $\boldsymbol{E} \in \mathbb{R}^{d \times |c|}$ of c can be constructed by first mapping each interaction in c into a d-dimensional latent vector and then puting all the vectors together into a matrix. Afterwards, in a horizontal convolutional layer, the mth convolution value α_m^x is achieved by sliding the xth filter $\boldsymbol{F}^x$ from the top to the bottom on $\boldsymbol{E}$ to interact with its horizontal dimensions:

$$\alpha_m^x = \phi_\alpha(\boldsymbol{E}_{m:m+h-1} \odot \boldsymbol{F}^x), \tag{16}$$

where ϕ_α is the activation function for the convolutional layer.

Then the final output $\boldsymbol{e}_c \in \mathbb{R}^z$ from the z filters is obtained by performing the max pooling operation on the convolution result $\boldsymbol{\alpha}^x = [\alpha_1^x, \alpha_2^x, \ldots, \alpha_{|c|-h+1}^x]$ to capture the most significant features in the session context:

$$\boldsymbol{e}_c = max\{max(\boldsymbol{\alpha}^1), max(\boldsymbol{\alpha}^2), ..., max(\boldsymbol{\alpha}^z)\}. \tag{17}$$

Finally, $\boldsymbol{e}_c$ is treated as the representation of the session context c and is used for subsequent recommendations [101].

Some variants include a 3D CNN model [104] built for SBRS, which jointly models the sequential patterns in click session data and the item characteristics from item content features, and a CNN model [76] to accumulate long-term user preferences for generating personalized recommendations. Furthermore, **Temporal Convolutional Networks (TCN)** were utilized to model the interactions within sessions to predict the next interaction [137].

8.1.4 Graph Neural Networks. In recent years, GNN have shown great expressive power in modeling the complex relations embedded in graph structured data by introducing deep neural networks into graph data [113, 122]. To benefit from this power, some researchers have introduced GNN to model the complex transitions within or between sessions to build better-performing SBRSs. First, given a dataset containing multiple sessions, it is transferred to a graph $\mathcal{G}$ by mapping each session into a chain on the graph. Each interaction o in a session serves as a node n in the corresponding chain where an edge e is created to connect each pair of the adjacent interactions in the session. Then, the constructed graph is imported into GNN to learn an informative embedding for each node (interaction) by encoding the complex transitions over the graph into the embeddings. Finally, these learned embeddings are imported into the prediction module for session-based recommendations. According to the specific model architecture of GNN, GNN approaches for SBRSs can be generally divided into three classes: **Gated Graph Neural Networks (GGNN)**, **Graph Convolutional Networks (GCN)**, and **Graph ATtention networks (GAT)**.

Gated Graph Neural Networks for SBRSs. In GGNN-based SBRSs, first, a directed graph is constructed based on all the historical ordered sessions, where the direction of each edge indicates the order of adjacent interactions within sessions. Then, each session graph, i.e., a chain (subgraph) for each session, is processed successively by GGNN to obtain the embedding $\boldsymbol{n}_i$ of node n_i, namely the embedding of the corresponding interaction o_i. Finally, after all the session graphs are processed, the embeddings of all the interactions are obtained, which are then used to construct the embedding of the session context for recommendations. Particularly, in GGNN, a GRU is used to learn the embedding of each node in a session graph by updating the embedding recurrently. Specifically, the embedding (also called hidden state) $\boldsymbol{h}_i^t$ of node n_i at step t is updated by the previous hidden state of itself and its neighbourhood nodes, i.e., $\boldsymbol{h}_i^{(t-1)}$ and $\boldsymbol{h}_j^{(t-1)}$,

$$\boldsymbol{h}_i^t = GRU\left(\boldsymbol{h}_i^{(t-1)}, \sum_{n_j \in N(n_i)} \boldsymbol{h}_j^{(t-1)}, \boldsymbol{A}\right), \tag{18}$$

where $N(n_i)$ is the set of neighbourhood nodes of n_i in the session graph, and $\boldsymbol{A}$ is the adjacency matrix built on the session graph. After multiple iterations until a stable equilibrium is reached, the hidden state at the final step of node n_i is taken as its embedding $\boldsymbol{n}_i$. **Session-based Recommendation with Graph Neural Networks (SR-GNN)** [129] is the pioneering work that introduced GNN into SBRSs and is claimed to have achieved superior performance, compared with non-GNN approaches including **Short-Term Attention/Memory Priority Model (STAMP)** [58], **Neural Attentive Recommendation Machine (NARM)** [52] and an RNN-based approach built on GRU, named GRU4Rec [38]. But it may not always outperform conventional methods, as discussed in Reference [65]. Other representative approaches falling into this stream include (1) **Graph Contextualized Self-Attention Network (GC-SAN)** [132], which utilizes both GNN and self-attention mechanism to learn local dependencies and long-range dependencies, respectively, for session-based recommendations, and (2) **Target Attentive GNN (TAGNN)** [138] that first learns item embedding with GNN and then attentively activates different user interests with respect to varied target items, for session-based recommendations.

Graph Convolutional Networks for SBRSs. Different from GGNN-based SBRSs, GCN-based SBRSs mainly utilize the pooling operation to integrate information from node n_i's neighbourhood node

n_j in the graph to help with the update of the hidden state of n_i as shown below:

$$\hat{\boldsymbol{h}}_i^t = pooling\left(\{\boldsymbol{h}_j^{(t-1)}, n_j \in N(n_i)\}\right), \tag{19}$$

where $N(n_i)$ is the set of neighbourhood nodes of node n_i. Different specific pooling operations including mean pooling and max pooling can be utilized, depending on the specific scenarios. Afterwards, the integrated neighbourhood information can be incorporated into the iterative update of the hidden state of node n_i [125],

$$\boldsymbol{h}_i^t = \boldsymbol{h}_i^{(t-1)} + \hat{\boldsymbol{h}}_i^t. \tag{20}$$

Finally, when a stable equilibrium is reached, the last hidden state of node n_i is taken as its embedding $\boldsymbol{n}_i$.

Graph ATtention networks for SBRSs. GAT-based SBRSs mainly utilize attention mechanism to attentively integrate the information from the neighbourhood nodes of node n_i in a session graph to update its hidden state in each attention layer [82]:

$$\boldsymbol{h}_i^t = attention\left(\{\boldsymbol{h}_j^{(t-1)}, n_j \in N(n_i)\}\right), \tag{21}$$

where $\boldsymbol{h}_i^t$ is the hidden state of node n_i in the tth attention layer. Here *attention* is a general attention module and can be specified to different operations including self attention, multi-head attention, and so on. In principal, the operations in *attention* can be divided into two steps (cf. Section 8.2.1): (1) calculating the importance weights of each neighbourhood node and (2) aggregating the hidden states of neighbourhood nodes according to their importance weights. Finally, once the forward propagation of multiple attention layers is completed, the hidden state of each node n_i in a session graph at the final layer is taken as its embedding $\boldsymbol{n}_i$.

Typical works falling into this class include **Full Graph Neural Network (FGNN)**, which learns the inherent order of the item transition patterns in sessions with a multiple **Weighted Graph ATtention (WGAT)** network [82], another FGNN based on Broadly Connected Session graph to attentively exploit information both within and between sessions [81], and Shortcut Graph ATention to effectively propagate information along shortcut connections with attention mechanism [13].

8.2 Advanced Model-based SBRSs

In addition to the aforementioned four classes of basic deep neural network approaches for SBRSs, there are also advanced approaches that are built on some advanced models or algorithms, including attention models, memory networks, mixture models, generative models and reinforcement learning. Usually, these advanced models or algorithms are combined with some basic approaches like distributed representation learning or RNN to construct more powerful SBRSs.

8.2.1 Attention Models. Attention-based SBRSs introduce the attention mechanism [108] to discriminatively exploit different elements, i.e., interactions or/and sessions, in a session context to build an informative session context representation for accurate recommendations. With the incorporation of attention mechanism, an SBRS is able to emphasize those elements that are more relevant to the next interaction or session and reduce the interference of the irrelevant ones in a session context. Generally, an attention model mainly contains two steps: attention weight calculation and aggregation. Next we introduce how an attention model learns a context representation for next interaction recommendations when the context includes the known part of the current session only (cf. Section 2.2). For contexts including historical sessions, their representations can be learned in a similar way.

Step 1: Given the embedding $\boldsymbol{o}_i$ of interaction o_i in session context c of next interaction o_{tg}, attention model calculates the weight $\beta_{tg,i}$ of o_i to indicate its relevance scale w.r.t o_{tg}, which is usually performed by a softmax function [118]:

$$\beta_{tg,i} = \frac{exp(e(\boldsymbol{o}_i))}{\sum_{o_j \in c} exp(e(\boldsymbol{o}_j))}, \tag{22}$$

where $e(\boldsymbol{o}_i)$ is a utility function, which can be specified as the inner product between a learnable weight vector $\boldsymbol{w}$ and $\boldsymbol{o}_i$. Sometimes, $\boldsymbol{o}_{tg}$ is also taken as an input of the utility function to make the learned weight more sensitive to the target interaction o_{tg}.

Step 2: The embeddings of all interactions in the session context c are aggregated with the learned weights to construct the embeddeding $\mathbf{e}_c$ for c:

$$\mathbf{e}_c = aggregate(\{\boldsymbol{o}_i, \beta_{tg,i},\ o_i \in c\}), \tag{23}$$

where *aggregate* is an aggregation function that is often specified as a weighted sum. The context embedding is then fed into the prediction module for generating recommendations.

In addition to the aforementioned basic attention model, a series of variants have been proposed for improving the performance of session-based recommendations. For example, a hierarchical attention model was proposed to attentively integrate both a user's historical sessions and the current session to capture her long- and short-term preferences for accurate session-based recommendations [116, 136]. Similarly, a co-attention network was designed to better explore the correlations between a user's current interaction and the interactions from historical sessions, for more accurate session-based recommendations [14]. It should be noted that attention models are usually integrated into other basic approaches, including encoder-decoder [62], distributed representation learning [118], RNN [52] and GNN [81], to enhance their capabilities for recommendations. In particular, the attention-enhanced GNN, i.e., GAT, has been introduced in Section 8.1.4. Other representative approaches for SBRSs that utilize attention models include the STAMP model [58], the self attention model [142] and the soft attention model [31].

8.2.2 Memory Networks. A memory network-based SBRS introduces a memory network to capture the dependency between any interaction in the session context and the next interaction directly by introducing an external memory matrix. Such matrix stores and updates the information of each interaction in a secession context more explicitly and dynamically to keep the most relevant and important information for the recommendation task.

To be specific, a memory network-based SBRS mainly consists of two major components: *a memory matrix* that maintains the embeddings of interactions in a session context c, and *a controller* that performs operations (including reading and writing) on the matrix [15]. Suppose $\mathbf{M}^c$ is the memory matrix to store the embeddedings of the recent interactions in c, where each column corresponds to the embedding of one interaction. After an interaction o_i happens in a session and is added into c, $\mathbf{M}^c$ will be updated accordingly to maintain the information of the recent interactions by writing the embedding $\mathbf{o}_i$ of o_i into it:

$$\mathbf{M}^c \leftarrow write(\mathbf{M}^c, \mathbf{o}_i), \tag{24}$$

where *write* stands for the write operation, and it can be specified as one of various writing processes, including the Least Recently Used Access [89].

During the prediction, the relevant information is carefully read from the maintained memory matrix to build the embedding $\mathbf{e}_c$ of the session context c:

$$\mathbf{e}_c = read(\mathbf{M}^c, \mathbf{o}_{tg}), \tag{25}$$

where $\mathbf{o}_{tg}$ is the embedding of the next interaction o_{tg} to be predicted, and it is considered during the reading process to read the information more relevant to o_{tg}. The read operation can be specified to multiple forms, and a typical one is to use the aforementioned attention mechanism (cf. Section 8.2.1) to attentively read the information from the memory matrix.

In addition to the basic memory network-based SBRS introduced above, some advanced variants have been proposed for better modeling sessions and making recommendations. For instance, two parallel memory modules, i.e., an Inner Memory Encoder and an Outer Memory Encoder, were proposed by Wang et al. [112] to model the current session and neighbourhood sessions, respectively, to build more informative embedding for a session context. Song et al. [96] proposed hierarchical memory networks to model a user's item-level and feature-level preferences simultaneously for better preforming SBR. Other typical works include the short-term attention/memory priority model for SBR [58], and the memory augmented neural model for incremental SBR [67].

8.2.3 Mixture Models. A mixture model-based SBRS mainly builds a compound model containing multiple sub-models to take the advantage of each one to comprehensively model the various complex dependencies embedded in session data. Usually, each sub-model excels at modeling a certain type of dependencies, e.g., low-order or higher-order dependencies. In principle, a mixture model-based SBRS performs two main steps: (1) learn different types of dependencies using different sub-models and (2) carefully integrate the learned dependencies for accurate SBR.

Representative mixture model-based SBRSs include neural Multi-temporal range Mixture Model (M3), which combines different kinds of encoders to capture short- and long-term dependencies in a session, respectively, for accurate recommendations [100], and **Mixture-channel Purpose Routing Networks (MCPRN)**, which employs multiple recurrent networks to model the intra-session dependencies under a user's different shopping purposes [123].

8.2.4 Generative Models. Generally speaking, approaches based on generative models for SBRSs make recommendations by generating the next interaction(s) or the next session via a carefully designed generation strategy. In this way, the recommendation procedure better approaches a user's online shopping behaviours in the real word, where items are often picked up step by step to form a shopping basket [121]. To be specific, given a session context c as the prior information, a list of interactions (items) l is generated to serve as the recommendation list:

$$l = generate(c), \tag{26}$$

where *generate* stands for a generation process, which can be specified as one of the various forms including probabilistic generative models [135].

Representative generative model-based SBRSs include *NextItNet* [141] where a probabilistic generative model was devised to generate a probability distribution over the candidate items; *Intention2Basket* model [121] where a utility-based generator was designed to generate a candidate session with the maximum utility to best fulfill a user's shopping intentions; the **Variational Recurrent Model (VRM)** where a stochastic generative process of sessions was specified [126]; and **VAriational SEssion-based Recommendation (VASER)** that utilized a non-linear probabilistic method for Bayesian inference to perform SBR [150].

8.2.5 Reinforcement Learning. **Reinforcement learning (RL)** approaches for SBRSs generally model the interactions between a user and an RS in a session as a **Markov Decision Process (MDP)**. Note that here the interaction particularly refers to the conversation between a user and an RS. For instance, first, an RS recommends an item to a user who provides some feedback on it, and then the RS recommends the subsequent item according to the user's feedback to better fit her preference. An RL-based SBRS aims to learn the optimal recommendation strategies via recommending trial-and-error items and receive reinforcements for these items from users' feedback [147]. In this way, an RL-based SBRS is able to continuously update its strategies during the interactions with users until reaching the optimal one that best fits the users' dynamic preferences. Moreover, the expected long-term cumulative reward from users is considered during the optimization of the strategies.

Table 8. A Comparison of Different Classes of Deep Neural Network Approaches for SBRSs

Approach		Applicable scenario	Pros	Cons	Typical work
Basic deep neural networ-ks	RNN	Long and rigidly ordered sessions	Model long-term and high-order sequential dependencies	The rigid order assumption is too strong for session data	[6],[11],[38], [39],[78],[85], [88],[99],[139]
	MLP	Unordered sessions, sessions with multi-aspects (e.g., static and dynamic features) to be combined	A simple structure, project sparse features to dense ones, learn the combination of different parts	Cannot model complex sessions, e.g., ordered, heterogeneous sessions	[17],[46],[98], [127]
	CNN	Flexible-ordered, heterogeneous or noisy sessions	Robust, no rigid order assumption, capture the union-level collective dependency	Relatively high complexity	[76],[101],[104], [137],[140]
	GNN	Complex sessions with complex transitions, e.g., repeat interactions	Model the complex transitions among interactions	Complex and costly	[81],[82],[125], [129],[132],[138]
Advanced models	Atten-tion	Heterogeneous, noisy, or long sessions	Identify and highlight important information	Cannot capture sequential information	[31],[52],[58], [62],[116],[118], [136],[142]
	Memo-ry	Long, incremental or noisy sessions	Dynamically store the latest information	Cannot capture sequential information	[15],[67],[89], [96],[112]
	Mixtu-re	Heterogeneous, noisy sessions	Model different types of dependencies, e.g., long and short term dependencies	Relatively complex and costly	[100],[123]
	Gener-ative	Dynamic, incremental sessions	Close to the practical session formation	Complex	[121],[126],[141]
	RL	Dynamic, incremental sessions	Interactive process, consider the future effect of actions	Hard to simulate the interactive environment	[40],[146],[147]

Following the work of Zhao et al. [147], we formalize a basic RL-based SBRS. First, the following five key concepts are defined in an RL-based SBRS. *State space* Sa, where a state $sa_t = \{sa_t^1, \ldots, sa_t^{m'}\} \in Sa$ is defined as the previous m' items with which a user interacted before time t. *Action space* Ac, where an action $ac_t = \{ac_t^1, \ldots, ac_t^{n'}\} \in Ac$ is to recommend a list of n' items to a user at time t based on the current state sa_t. *Reward* Re: After the RS takes an action ac_t at the state sa_t, it receives immediate reward Re_t according to the user's feedback. *Transition probability* $Tp(sa_{t+1}|sa_t, ac_t)$ defines the probability of the state transition from sa_t to sa_{t+1} when the RS takes an action ac_t. *Discount factor* df: $df \in [0,1]$ defines the discount factor when we measure the present value of the future reward. Therefore, SBR can be formalized to find a recommendation policy $\pi : Sa \to Ac$ to maximize the cumulative reward for an RS given the historical MDP, i.e., (Sa, Ac, Re, Tp, df).

There are three main steps in an RL-based SBRS. The first step is to calculate the state-specific weight parameters by mapping state sa_t to a weight matrix $\boldsymbol{W}_t$:

$$f_t : sa_t \to \boldsymbol{W}_t. \tag{27}$$

The second step is to calculate the score of each candidate item using the score function f_s and then select items with the highest score for recommendations:

$$score(v_i) = f_s(\boldsymbol{v}_i, \boldsymbol{W}_t). \tag{28}$$

The final step is to calculate the action value $E(sa_t, ac_t)$ of the potential action ac_t, i.e., to recommend the selected item, to judge whether ac_t matches the current state sa_t or not [147]. Usually, the following optimal action-value function $E^*(sa_t, ac_t)$, namely the maximum expected return achievable by the optimal policy [146], is used:

$$E^*(sa_t, ac_t) = \mathbb{E}_{sa_{t+1}}[Re_t + df\ max_{ac_{t+1}} E^*(sa_{t+1}, ac_{t+1})|sa_t, ac_t]. \tag{29}$$

Subsequently, the recommendation strategies are optimized by minimizing the error between the action value of the ground truth action and that of trialed actions.

Typical works on RL-based SBRSs include **LIst-wise Recommendation framework based on Deep reinforcement learning (LIRD)** [147] that learns recommendation strategies for list-wise recommendations; a similar work called DeepPage for page-wise recommendations [146]; and Reinforcement Learning with Window for Recommendation [40] where a state compression method was proposed to capture an enormous state space for play-list recommendations.

8.3 Comparison of Deep Neural Network-based SBRS Approaches

After introducing the main ideas and the key technical details of deep neural network approaches for SBRSs, we present a comparison and summary of these approaches. In particular, in Table 8, we compare the two classes of deep neural network approaches, including nine sub-classes, in terms of their applicable scenarios, the typical example of pros and cons, and the typical works.

9 SBRS APPLICATIONS, ALGORITHMS, AND DATASETS

9.1 SBRS Applications

SBRSs are widely applied in a variety of real-word domains and scenarios to benefit both customers and businesses. A summary of SBRS applications is presented in Table 9. Generally speaking, these applications can be grouped into (1) *conventional applications*, e.g., next-item recommendation in E-commerce, and (2) *emerging applications*, e.g., next-treatment recommendation in healthcare.

According to whether to recommend a product, a content or a service, we can organize SBR into product recommendation, content recommendation and service recommendation. The conventional applications involve all these three classes while the emerging applications mainly involve service recommendation (cf. Table 9). From our observation, most of the existing works on SBRS focus on the conventional applications, especially the E-commerce domain, e.g., to recommend the next item [134] or next basket of items [121] on an online shopping platform (e.g., amazon.com). In addition, it is not uncommon that SBRSs are applied to other conventional domains, e.g., next news/web-page recommendation in media domain [98], next song/movie/video recommendation in entertainment domain [147], and next-POI recommendation in tourism domain [59].

Compared with the prosper of conventional applications of SBRS, the emerging applications of SBRS are just in their early stage. However, the applications of SBRS in emerging domains including finance and healthcare are promising and deserve to be further explored. For example, SBRS is of great potential to recommend next trading strategies or portfolios to an investor per her investment goals and context in a financial market and to suggest personalized treatments [131] on a patient according to her health conditions, past treatments and medical treatment protocols [32].

9.2 Algorithms and Datasets for SBRSs

The source code of most of the representative SBRS algorithms is publicly accessible. We summarize the open-source code of algorithms for SBRSs built on different models for various tasks in Table 1 in the supplemental material to facilitate the access for empirical analysis. The supplemental material is online only and can be accessed via the official website of the journal.

Datasets are necessary for evaluating SBRS algorithms. We summarize a collection of 13 publicly accessible real-world datasets that are commonly used for SBRS evaluations in Table 2 in the

Table 9. A Summary of SBRS Applications

Category		Application domain	Application scenario	Typical work
Conventional application	Product recommendation	E-commerce	Next-item/basket recommendation	[38],[43],[45],[63], [114],[121],[129],[134]
	Content recommendation	Media, entertainment	Next news/web-page/song/movie /video recommendation	[21],[49],[70],[78], [98],[100],[107],[147]
	Service recommendation	Tourism	Next-POI recommendation	[16],[59]
Emerging application	Service recommendation	Finance	Next-trading recommendation	[26],[131]
		Healthcare	Next-treatment recommendation	[32]

supplemental material. These datasets cover a wide range of application domains from e-commerce to POI and are with various characteristics. They can provide challenging test-beds for SBRS algorithms. These algorithms and datasets are also available in the arXiv version[4] of this article.

10 PROSPECTS AND FUTURE DIRECTIONS

Our comprehensive review of the literature has revealed the significant challenges facing SBRS research and the enormous opportunities SBRS presents. In this section, we outline several promising prospective research directions, which we believe are critical to the further development of the filed.

10.1 Session-based Recommendations with General User Preference

Significance. SBRSs usually ignore users' long-term general preferences that can be well captured by conventional RSs like collaborative-filtering-based RSs. This may lead to unreliable recommendations, since users with different general preferences and consumption habits may chose different items even under the same session context. In this case, how to effectively incorporate users' general preferences into an SBRS is critical yet challenging.

Open issues. Here, we discuss two major issues w.r.t. the general preference learning as well as its incorporation into SBRSs and sketch several critical future research directions.

— How to incorporate users' explicit general preference into an SBRS? In this case, it is assumed that the explicit user-item preference data, e.g., a user-item rating matrix, is available. An intuitive way is to first learn users' general preferences from the explicit preference data using conventional RS approaches, e.g., **Matrix Factorization (MF)**, and then take the learned preference as an indicator to fine-tune the ranking of candidate items in an SBRS, e.g., to put those items more preferred by a user to the front of the recommendation list. Another way is to combine both users' long-term general preferences and short-term preferences together when ranking the candidate items. For instance, a Generative Adversarial Network framework was proposed by Zhao et al. [145] to build a hybrid model for movie recommendations. In this model, MF and RNN are utilized to learn users' long-term preference and short-term preference, respectively. However, the efforts to address this issue are still limited and more efforts are needed.

— How to incorporate a user's implicit general preference into an SBRS? In the real word, the explicit preference data may not be always available, since users may or may not provide explicit feedback, e.g., ratings, on everything they bought. In such a case, the implicit preference data, i.e., users' transaction behaviour data including view, click, add to cart and purchase, can be leveraged to learn users' implicit general preference [35, 36]. In practice, such implicit preference data is often available in the session-based recommdation scenario [91]. Although a variety of works [3, 80] have explored how to learn a users' general preference from such implicit preference data

[4]https://arxiv.org/abs/1902.04864.

in conventional RSs, e.g., collaborative filtering, the efforts in SBRS field are still limited. Therefore, how to simultaneously learn a user's implicit general preference and her short-term preference and effecetively integrate them for accurate SBR is a challenging issue that requires more efforts.

10.2 Session-based Recommendations Considering More Contextual Factors

Significance. A context refers to the specific internal and external environment when a user makes choices on items [9]. Accordingly, contextual factors refer to context-related aspects that may affect a user's choices, such as weather, season, location, time and recent popularity trend. Taking these contextual factors into account may make a huge difference on recommendation performance [47], as demonstrated by researchers including Adomavicius et al. [1] and Pagano et al. [73]. In practice, an SBRS can be seen as a simplified context-aware RS whose context is simplified to a session context [105]. Although contextual information has been incorporated into other types of RSs including context-aware RSs [1, 106], most of the contextual factors are rarely exploited in SBRSs.

Open issues. How to incorporate more contextual factors into SBRSs? A few works have provided some initial solutions to this issue. A Contextual RNN for Recommendation was proposed to incorporate contextual factors including the time gap between different interactions and the time of a day when an interaction happens, into an RNN-based SBRS [95]. In several works [44, 51], the recent popularity trend, a user's recently viewed items, and items with discount in shopping mall are taken as contextual factors for SBR. However, these solutions are just a starting point, and more explorations are still necessary to address issues such as how to collect more contextual information and how to more effectively incorporate it for more accurate SBR.

10.3 Session-based Recommendations with Cross-domain Information

Significance. Cross domains mean different but relevant domains [42, 152], e.g., movie domain and song domain. Usually, a user's purchased items come from multiple domains rather than a single one to meet her demand [151]. In addition, a user's choices on items from different domains are often dependent. For example, after Alice watches the movie "Titanic," she may listen to the movie's theme song "My heart will go on." Such an example shows that items from different domains may not only be dependent but can even form a sequential session, such as {"Titanic," "My heart will go on"}. The recommendations based on such type of session data are interesting but quite challenging. Such recommendations not only cover more aspects of our daily lives but also provide a solution to the data sparsity issue when only one domain is considered. However, it is hard to collect a user's consumed items from various domains together, and also the relations between items from different domains are much more complex than that from a single domain.

Open issues. According to whether the items from different domains can form a session or not, there are two main open issues to be explored.

— How to borrow knowledge from other domains to benefit the SBR in the target domain? When no sessions can be built across different domains, a target-auxiliary framework can be employed to benefit SBR from other domains. To be specific, the framework takes the target domain where the recommendations are made as the main information source while taking other domains as a supplementary one. An intuitive instance of the framework could utilize transfer learning [23, 74] that transfers knowledge from source domains to help with the tasks in the target one. Although transfer learning has been well explored in conventional RSs like collaborative filtering [60, 75], it is rarely explored in SBRSs.

— How to perform SBR on multiple domains? This case happens when sessions can be built from different domains. Different from the aforementioned target-auxiliary framework, in this case, RSs treat items from different domains equally and each domain can serve as the target domain to make recommendations. This is much more interesting yet challenging than the first case. Consequently, how to develop advanced models to effectively capture the complex and heterogeneous dependencies [8] among different domains for accurate SBR requires more explorations.

10.4 Session-based Recommendations by Considering More User Behaviour Patterns

Significance. In addition to the basic co-occurrence or sequential behaviour patterns hidden in session data, there are actually more types of user behaviour patterns that can be leveraged to benefit SBR, such as repeat consumption [86] and periodic consumption [41]. Such kinds of behaviour patterns are not uncommon in the real world, but are usually ignored by existing works on SBRSs.

Open issues. How to effectively discover and leverage more types of user behaviour patterns to improve SBR? On the one hand, the other types of patterns are less frequent and obvious than the basic co-occurrence or sequential patterns that are thus more difficult to be precisely identified. On the other hand, how these patterns can influence the users' final choices on items is not very clear. Hence, it is difficult to effectively incorporate the useful information from these patterns while reducing noise to benefit the recommendations. Such issues are of practical significance in real-world applications, especially in the e-commence industry, and thus require more efforts.

10.5 Session-based Recommendations with Constraints

Significance. In reality, it is not uncommon that there are some types of underlying constraints over the items in one session. For example, in some cases, the items purchased in one session may not be identical or similar, instead, they may complement each other [146], e.g., milk and bread, to form a coherent package to satisfy a user's certain goal, e.g., breakfast. In other cases, there may be duplicated items within a session, since a user may buy multiple copies of items in one session. Such kinds of constraints are often ignored by existing works on SBRSs.

Open issues. How to generate a session with some constraints on it to better satisfy a user's purchase goal? On the one hand, different users may have different interaction patterns and requirements to be satisfied with different constraints. Therefore, it is difficult to determine which constraints should be added to a specific user to best accomplish her goal. On the other hand, it is also challenging to jointly optimize both the added constraints and the prediction accuracy. This open issue is challenging but of practical concern, with only limited attention from the community [84]. A straightforward solution is to incorporate some semantic relations between items by employing a knowledge graph [111] into SBRSs.

10.6 Interactive Session-based Recommendations

Significance. In the real-word cases, particularly in the online shopping scenario, a session is often generated from a continuous interactive process between a user and the shopping platform. For example, a user may first click an item to start a session, followed by different actions on this item, e.g., view it, add it to cart, or just skip it. By taking these different actions as different feedback from the user, the platform can then accordingly adjust recommendation strategies for the subsequent items. Such interactive process proceeds until to the end of the session. Though such intrinsic nature embedded in session generation process is quite important for precisely learning the user's dynamic preference for accurate SBR, it has been overlooked by most of the existing works on SBRSs.

Open issues. How to effectively model the continuous interactive process between users and the platform for interactive SBR? This issue is critical but challenging in the area of SBRSs. In such a case, a user's preference is revealed by her limited real-time interactions with the platforms and usually changes over time. So, it is difficult to precisely capture the user's dynamic preferences from limited interactions in a timely manner. Although, some researchers have proposed approaches based on reinforcement learning [153] to address it, the studies are still in early stage while more and deeper explorations are needed.

10.7 Online or Streaming Session-based Recommendations

Significance. On a real-world online shopping platform, session data usually comes incrementally in a streaming scenario. This leads to the continuous, large-volume, high-velocity nature of session data [31, 148]. However, most of the existing studies on SBRSs work on the offline and static data, which may be inconsistent with the real-world application scenario.

Open issues. How to effectively learn users' dynamic preferences in an online and streaming scenario for better SBR? This is much more challenging than the SBRSs based on offline data, but of greater significance from the practical perspective. Specifically, it is quite challenging to develop accurate and highly-efficient recommendation algorithms to effectively model large-volume streaming data and generate timely recommendations. Researchers tried to address this issue with adaptively distilled exemplar replay strategy [68], a continuously queried and updated non-parametric memory mechanism [67], or a reservoir-based streaming model [31, 83]. More efforts are still required for this open issue.

11 CONCLUSIONS

In this article, we have conducted a systematic and extensive review of the most notable works to date on SBRSs. We have proposed a unified framework to organise the existing works in this area into three sub-areas and provided a unified problem statement for SBRS to reduce some confusions and inconsistencies in the field. We have thoroughly analyzed the characteristics of session data and the corresponding challenges they bring for SBRSs. We have also proposed a classification scheme for the organization and clustering of existing approaches for SBRSs, and highlighted some critical technical details for each class of approaches. In addition, we have discussed some of the most pressing open issues and promising directions. The research in SBRS field is flourishing and a number of newly developed techniques and emerging approaches keep coming. It is our hope that this survey can provide readers with a comprehensive understanding of the key aspects, main challenges, notable progress in this area, and shed some light on future studies.

ACKNOWLEDGMENTS

The authors thank Dr. Liang Hu and Dr. Yan Zhao for their constructive suggestions on this work.

REFERENCES

[1] Gediminas Adomavicius and Alexander Tuzhilin. 2015. Context-aware recommender systems. In *Recommender Systems Handbook*. Springer, 191–226.

[2] Charu C. Aggarwal. 2016. Content-based recommender systems. In *Recommender Systems*. Springer, 139–166.

[3] Susan C. Anyosa, João Vinagre, and Alípio M. Jorge. 2018. Incremental matrix co-factorization for recommender systems with implicit feedback. In *Companion of the Web Conference*. 1413–1418.

[4] Hans-Jürgen Bandelt and Andreas W. M. Dress. 1992. A canonical decomposition theory for metrics on a finite set. *Adv. Math.* 92, 1 (1992), 47–105.

[5] Alex Beutel, Paul Covington, Sagar Jain, et al. 2018. Latent cross: Making use of context in recurrent recommender systems. In *WSDM*. ACM, 46–54.

[6] Veronika Bogina and Tsvi Kuflik. 2017. Incorporating dwell time in session-based recommendations with recurrent neural networks. In *Proceedings of the RecTemp Workshop co-located with ACM RecSys'17*. 57–59.

[7] Robin Burke. 2002. Hybrid recommender systems: Survey and experiments. *User Model. User-Adapt. Interact.* 12, 4 (2002), 331–370.

[8] Longbing Cao. 2015. Coupling learning of complex interactions. *Inf. Process. Manage.* 51, 2 (2015), 167–186.

[9] Longbing Cao. 2016. Non-IID recommender systems: A review and framework of recommendation paradigm shifting. *Engineering* 2, 2 (2016), 212–224.

[10] Longbing Cao. 2018. *Data Science Thinking: The Next Scientific, Technological and Economic Revolution.* Springer.

[11] Sotirios P. Chatzis, Panayiotis Christodoulou, et al. 2017. Recurrent latent variable networks for session-based recommendation. In *DLRS.* 38–45.

[12] Shuo Chen, Josh L. Moore, et al. 2012. Playlist prediction via metric embedding. In *SIGKDD.* ACM, 714–722.

[13] Tianwen Chen and Raymond Chi-Wing Wong. 2020. Handling information loss of graph neural networks for session-based recommendation. In *SIGKDD.* 1172–1180.

[14] Wanyu Chen, Fei Cai, et al. 2019. A dynamic co-attention network for session-based recommendation. In *CIKM.* 1461–1470.

[15] Xu Chen, Hongteng Xu, Yongfeng Zhang, et al. 2018. Sequential recommendation with user memory networks. In *WSDM.* 108–116.

[16] Chen Cheng, Haiqin Yang, Michael R. Lyu, and Irwin King. 2013. Where you like to go next: Successive point-of-interest recommendation. In *IJCAI.* 2605–2611.

[17] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, et al. 2016. Wide & deep learning for recommender systems. In *DLRS.* ACM, 7–10.

[18] Keunho Choi, Donghee Yoo, Gunwoo Kim, and Yongmoo Suh. 2012. A hybrid online-product recommendation system: combining implicit rating-based collaborative filtering and sequential pattern analysis. *Electr. Commerce Res. Appl.* 11, 4 (2012), 309–317.

[19] Panayiotis Christodoulou, Sotirios P. Chatzis, et al. 2017. A variational recurrent neural network for session-based recommendations using bayesian personalized ranking. In *ISD.* 1–9.

[20] Tim Donkers, Benedikt Loepp, and Jürgen Ziegler. 2017. Sequential user-based recurrent neural network recommendations. In *RecSys.* ACM, 152–160.

[21] Magdalini Eirinaki, Michalis Vazirgiannis, et al. 2005. Web path recommendations based on page ranking and markov models. In *WIDM.* ACM, 2–9.

[22] Michael D. Ekstrand, John T. Riedl, Joseph A. Konstan, et al. 2011. Collaborative filtering recommender systems. *Found. Trends Hum.–Comput. Interact.* 4, 2 (2011), 81–173.

[23] Ali Mamdouh Elkahky, Yang Song, et al. 2015. A multi-view deep learning approach for cross domain user modeling in recommendation systems. In *WWW.* 278–288.

[24] Hui Fang, Danning Zhang, Yiheng Shu, and Guibing Guo. 2020. Deep learning for sequential recommendation: Algorithms, influential factors, and evaluations. *Trans. Inf. Syst.* 39, 1 (2020), 1–42.

[25] Shanshan Feng, Xutao Li, Yifeng Zeng, et al. 2015. Personalized ranking metric embedding for next new POI recommendation. In *IJCAI.* 2069–2075.

[26] Dušan Fister, Matjaž Perc, and Timotej Jagrič. 2021. Two robust long short-term memory frameworks for trading stocks. *Appl Intell* (2021). https://doi.org/10.1007/s10489-021-02249-x

[27] R. Forsati, M. R. Meybodi, and A. Ghari Neiat. 2009. Web page personalization based on weighted association rules. In *ICECT.* IEEE, 130–135.

[28] P. Moreira Gabriel De Souza, Dietmar Jannach, and Adilson Marques Da Cunha. 2019. Contextual hybrid session-based news recommendation with recurrent neural networks. *IEEE Access* 7 (2019), 169185–169203.

[29] Diksha Garg, Priyanka Gupta, Pankaj Malhotra, Lovekesh Vig, and Gautam Shroff. 2019. Sequence and time aware neighborhood for session-based recommendations: Stan. In *SIGIR.* 1069–1072.

[30] Asnat Greenstein-Messica, Lior Rokach, and Michael Friedman. 2017. Session-based recommendations using item embedding. In *IUI.* ACM, 629–633.

[31] Lei Guo, Hongzhi Yin, Qinyong Wang, et al. 2019. Streaming session-based recommendation. In *SIGKDD.* 1569–1577.

[32] Kyle Haas, Stuart Morton, et al. 2019. Using similarity metrics on real world data and patient treatment pathways to recommend the next treatment. In *AMIA Summits on Translational Science Proceedings*, 398.

[33] Jiawei Han, Jian Pei, and Yiwen Yin. 2000. Mining frequent patterns without candidate generation. In *ACM Sigmod Record*, Vol. 29. ACM, 1–12.

[34] Negar Hariri, Bamshad Mobasher, and Robin Burke. 2012. Context-aware music recommendation based on latent topic sequential patterns. In *RecSys.* 131–138.

[35] Ruining He and Julian McAuley. 2016. VBPR: Visual bayesian personalized ranking from implicit feedback. In *AAAI.* 144–150.

[36] Xiangnan He, Hanwang Zhang, Min-Yen Kan, and Tat-Seng Chua. 2016. Fast matrix factorization for online recommendation with implicit feedback. In *SIGIR.* ACM, 549–558.

[37] Balázs Hidasi and Alexandros Karatzoglou. 2018. Recurrent neural networks with top-k gains for session-based recommendations. In *CIKM.* 843–852.

[38] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2016. Session-based recommendations with recurrent neural networks. In *ICLR*. 1–10.

[39] Balázs Hidasi, Massimo Quadrana, et al. 2016. Parallel recurrent neural network architectures for feature-rich session-based recommendations. In *RecSys*. ACM, 241–248.

[40] Binbin Hu, Chuan Shi, and Jian Liu. 2017. Playlist recommendation based on reinforcement learning. In *ICIS*. Springer, 172–182.

[41] Haoji Hu, Xiangnan He, Jinyang Gao, and Zhi-Li Zhang. 2020. Modeling personalized item frequency information for next-basket recommendation. In *SIGIR*. 1–10.

[42] Liang Hu, Jian Cao, Guandong Xu, et al. 2013. Cross-domain collaborative filtering via bilinear multilevel analysis. In *IJCAI*. AAAI Press, 2626–2632.

[43] Liang Hu, Longbing Cao, Shoujin Wang, et al. 2017. Diversifying personalized recommendation with user-session context. In *IJCAI*. 1858–1864.

[44] Dietmar Jannach and Malte Ludewig. 2017. Determining characteristics of successful recommendations from log data: A case study. In *SAC*. ACM, 1643–1648.

[45] Dietmar Jannach and Malte Ludewig. 2017. When recurrent neural networks meet the neighborhood for session-based recommendation. In *RecSys*. ACM, 306–310.

[46] Dietmar Jannach, Malte Ludewig, et al. 2017. Session-based item recommendation in e-commerce: on short-term intents, reminders, trends and discounts. *User Model. User-Adapt. Interact.* 27, 3-5 (2017), 351–392.

[47] Dietmar Jannach, Bamshad Mobasher, and Shlomo Berkovsky. 2020. Research directions in session-based and sequential recommendation. *User Model. User-Adapt. Interact.* 30, 4 (2020), 609–616.

[48] Priit Järv. 2019. Predictability limits in session-based next item recommendation. In *RecSys*. 146–150.

[49] How Jing and Alexander J. Smola. 2017. Neural survival recommender. In *WSDM*. ACM, 515–524.

[50] Duc-Trong Le, Yuan Fang, and Hady W Lauw. 2016. Modeling sequential preferences with dynamic user and context factors. In *ECML-PKDD*. Springer, 145–161.

[51] Lukas Lerche, Dietmar Jannach, and Malte Ludewig. 2016. On the value of reminders within e-commerce recommendations. In *UMAP*. ACM, 27–35.

[52] Jing Li, Pengjie Ren, Zhumin Chen, Zhaochun Ren, et al. 2017. Neural attentive session-based recommendation. In *CIKM*. ACM, 1419–1428.

[53] Yuqi Li et al. 2017. Learning graph-based embedding for time-aware product recommendation. In *CIKM*. ACM, 2163–2166.

[54] Zhi Li, Hongke Zhao, Qi Liu, Zhenya Huang, Tao Mei, and Enhong Chen. 2018. Learning from history and present: Next-item recommendation via discriminatively exploiting user behaviors. In *SIGKDD*. 1734–1743.

[55] Defu Lian, Vincent W. Zheng, and Xing Xie. 2013. Collaborative filtering meets next check-in location prediction. In *WWW*. ACM, 231–232.

[56] Dawen Liang, Jaan Altosaar, et al. 2016. Factorization meets the item embedding: Regularizing matrix factorization with item co-occurrence. In *RecSys*. ACM, 59–66.

[57] Duen-Ren Liu, Chin-Hui Lai, and Wang-Jung Lee. 2009. A hybrid of sequential rules and collaborative filtering for product recommendation. *Inf. Sci.* 179, 20 (2009), 3505–3519.

[58] Qiao Liu, Yifu Zeng, Refuoe Mokhosi, and Haibin Zhang. 2018. STAMP: Short-Term attention/memory priority model for session-based recommendation. In *SIGKDD*. ACM, 1831–1839.

[59] Xin Liu, Yong Liu, Karl Aberer, and Chunyan Miao. 2013. Personalized point-of-interest recommendation by mining users' preference transition. In *CIKM*. ACM, 733–738.

[60] Babak Loni, Yue Shi, Martha Larson, and Alan Hanjalic. 2014. Cross-domain collaborative filtering with factorization machines. In *ECIR*. Springer, 656–661.

[61] Pasquale Lops, Marco De Gemmis, and Giovanni Semeraro. 2011. Content-based recommender systems: State of the art and trends. In *Recommender Systems Handbook*. Springer, 73–105.

[62] Pablo Loyola, Chen Liu, and Yu Hirate. 2017. Modeling user session and intent with an attention-based encoder-decoder architecture. In *RecSys*. ACM, 147–151.

[63] Malte Ludewig and Dietmar Jannach. 2018. Evaluation of session-based recommendation algorithms. *User Model. User-Adapt. Interact.* 28, 4-5 (2018), 331–390.

[64] Malte Ludewig, Noemi Mauro, et al. 2019. Performance comparison of neural and non-neural approaches to session-based recommendation. In *RecSys*. 462–466.

[65] Malte Ludewig, Noemi Mauro, Sara Latifi, and Dietmar Jannach. 2021. Empirical analysis of session-based recommendation algorithms. *User Model. User-Adapt. Interact.* 31, 1 (2021), 149–181.

[66] Wenjing Meng, Deqing Yang, and Yanghua Xiao. 2020. Incorporating user micro-behaviors and item knowledge into multi-task learning for session-based recommendation. In *SIGIR*. 1–10.

[67] Fei Mi and Boi Faltings. 2020. Memory augmented neural model for incremental session-based recommendation. In *IJCAI*. 1–7.

[68] Fei Mi, Xiaoyu Lin, and Boi Faltings. 2020. Ader: Adaptively distilled exemplar replay towards continual learning for session-based recommendation. In *RecSys*. 408–413.

[69] Tomas Mikolov, Quoc V Le, and Ilya Sutskever. 2013. Exploiting similarities among languages for machine translation. arXiv:1309.4168. Retrieved from https://arxiv.org/abs/1309.4168.

[70] Bamshad Mobasher et al. 2001. Effective personalization based on association rule discovery from web usage data. In *WIDM*. ACM, 9–15.

[71] María N. Moreno, Francisco J. García, et al. 2004. Using association analysis of web data in recommender systems. In *EC-Web*. Springer, 11–20.

[72] Utpala Niranjan, R. B. V. Subramanyam, and V. Khanaa. 2010. Developing a web recommendation system based on closed sequential patterns. In *ICT*. Springer, 171–179.

[73] Roberto Pagano, Paolo Cremonesi, et al. 2016. The contextual turn: From context-aware to context-driven recommender systems. In *RecSys*. ACM, 249–252.

[74] Sinno Jialin Pan and Qiang Yang. 2010. A survey on transfer learning. *TKDE* 22, 10 (2010), 1345–1359.

[75] Weike Pan and Qiang Yang. 2013. Transfer learning in heterogeneous collaborative filtering domains. *Artif. Intell.* 197 (2013), 39–55.

[76] Keunchan Park, Jisoo Lee, and Jaeho Choi. 2017. Deep neural networks for news recommendations. In *CIKM*. ACM, 2255–2258.

[77] Michael J. Pazzani and Daniel Billsus. 2007. Content-based recommendation systems. In *The Adaptive Web*. Springer, 325–341.

[78] Wenjie Pei, Jie Yang, Zhu Sun, et al. 2017. Interacting attention-gated recurrent networks for recommendation. In *CIKM*. ACM, 1459–1468.

[79] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*. 1532–1543.

[80] Ladislav Peska and Peter Vojtas. 2017. Using implicit preference relations to improve recommender systems. *J. Data Semant.* 6, 1 (2017), 15–30.

[81] Ruihong Qiu, Zi Huang, Jingjing Li, and Hongzhi Yin. 2020. Exploiting cross-session information for session-based recommendation with graph neural networks. *Trans. Inf. Syst.* 38, 3 (2020), 1–23.

[82] Ruihong Qiu, Jingjing Li, Zi Huang, and Hongzhi Yin. 2019. Rethinking the item order in session-based recommendation with graph neural networks. In *CIKM*. 579–588.

[83] Ruihong Qiu, Hongzhi Yin, Zi Huang, and Tong Chen. 2020. Gag: Global attributed graph neural network for streaming session-based recommendation. In *SIGIR*. 669–678.

[84] Massimo Quadrana, Paolo Cremonesi, and Dietmar Jannach. 2018. Sequence-aware recommender systems. *Comput. Surv.* 51, 4 (2018), 1–36.

[85] Massimo Quadrana et al. 2017. Personalizing session-based recommendations with hierarchical recurrent neural networks. In *RecSys*. ACM, 130–137.

[86] Pengjie Ren, Zhumin Chen, Jing Li, et al. 2019. RepeatNet: a repeat aware neural recommendation machine for session-based recommendation. In *AAAI*, Vol. 33. 4806–4813.

[87] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2010. Factorizing personalized markov chains for next-basket recommendation. In *WWW*. ACM, 811–820.

[88] Massimiliano Ruocco, Ole Steinar Lillestøl Skrede, and Helge Langseth. 2017. Inter-session modeling for session-based recommendation. In *DLRS*. ACM, 24–31.

[89] Adam Santoro, Sergey Bartunov, Matthew Botvinick, et al. 2016. Meta-learning with memory-augmented neural networks. In *ICML*. 1842–1850.

[90] J. Ben Schafer, Dan Frankowski, Jon Herlocker, and Shilad Sen. 2007. Collaborative filtering recommender systems. In *The Adaptive Web*. Springer, 291–324.

[91] Tobias Schnabel, Paul N. Bennett, Susan T. Dumais, and Thorsten Joachims. 2018. Short-term satisfaction and long-term coverage: Understanding how users tolerate algorithmic exploration. In *WSDM*. ACM, 513–521.

[92] Guy Shani, David Heckerman, and Ronen I. Brafman. 2005. An MDP-based recommender system. *J. Mach. Learn. Res.* 6, (Sep. 2005), 1265–1295.

[93] Bo Shao, Dingding Wang, Tao Li, and Mitsunori Ogihara. 2009. Music recommendation based on acoustic features and user access patterns. *IEEE Trans. Aud. Speech Lang. Process.* 17, 8 (2009), 1602–1611.

[94] Yue Shi, Martha Larson, and Alan Hanjalic. 2014. Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges. *Comput. Surv.* 47, 1 (2014), 3.

[95] Elena Smirnova and Flavian Vasile. 2017. Contextual sequence modeling for recommendation with recurrent neural networks. In *DLRS*. 2–9.

[96] Bo Song, Yi Cao, et al. 2019. Session-based recommendation with hierarchical memory networks. In *CIKM*. 2181–2184.

[97] Wei Song and Kai Yang. 2014. Personalized recommendation based on weighted sequence similarity. In *Practical Applications of Intelligent Systems*. Springer, 657–666.

[98] Yang Song et al. 2016. Multi-rate deep learning for temporal recommendation. In *SIGIR*. ACM, 909–912.

[99] Yong Kiam Tan, Xinxing Xu, and Yong Liu. 2016. Improved recurrent neural networks for session-based recommendations. In *DLRS*. ACM, 17–22.

[100] Jiaxi Tang, Francois Belletti, Sagar Jain, Minmin Chen, et al. 2019. Towards neural mixture recommender for long range dependent user sequences. In *WWW*. 1782–1793.

[101] Jiaxi Tang and Ke Wang. 2018. Personalized top-n sequential recommendation via convolutional sequence embedding. In *WSDM*. 565–573.

[102] Md Mehrab Tanjim, Congzhe Su, Ethan Benjamin, et al. 2020. Attentive sequential models of latent intent for next item recommendation. In *The Web Conference*. 2528–2534.

[103] Maryam Tavakol and Ulf Brefeld. 2014. Factored MDPs for detecting topics of user sessions. In *RecSys*. 33–40.

[104] Trinh Xuan Tuan and Tu Minh Phuong. 2017. 3D convolutional networks for session-based recommendation with content features. In *RecSys*. ACM, 138–146.

[105] Bartłomiej Twardowski. 2016. Modelling contextual information in session-aware recommender systems with neural networks. In *RecSys*. ACM, 273–276.

[106] Moshe Unger. 2015. Latent context-aware recommender systems. In *RecSys*. ACM, 383–386.

[107] Flavian Vasile, Elena Smirnova, and Alexis Conneau. 2016. Meta-Prod2Vec: Product embeddings using side-information for recommendation. In *RecSys*. ACM, 225–232.

[108] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*. 5998–6008.

[109] Shengxian Wan, Yanyan Lan, Pengfei Wang, Jiafeng Guo, Jun Xu, and Xueqi Cheng. 2015. Next basket recommendation with neural networks. In *RecSys*. ACM, 1–2.

[110] Dongjing Wang, Shuiguang Deng, et al. 2016. Learning music embedding with metadata for context aware recommendation. In *ICMR*. ACM, 249–253.

[111] Hongwei Wang, Fuzheng Zhang, et al. 2018. Ripplenet: Propagating user preferences on the knowledge graph for recommender systems. In *CIKM*. 417–426.

[112] Meirui Wang, Pengjie Ren, Lei Mei, Zhumin Chen, Jun Ma, and Maarten de Rijke. 2019. A collaborative session-based recommendation approach with parallel memory modules. In *SIGIR*. 345–354.

[113] Nan Wang, Shoujin Wang, Yan Wang, et al. 2020. Modelling local and global dependencies for next-item recommendations. In *WISE*. Springer, 285–300.

[114] Pengfei Wang, Jiafeng Guo, Yanyan Lan, Jun Xu, Shengxian Wan, and Xueqi Cheng. 2015. Learning hierarchical representation model for next basket recommendation. In *SIGIR*. ACM, 403–412.

[115] Shoujin Wang and Longbing Cao. 2017. Inferring implicit rules by learning explicit and hidden item dependency. *IEEE Trans. Syst. Man Cybernet. Syst.* 50, 3 (2017), 935–946.

[116] Shoujin Wang, Longbing Cao, Liang Hu, Shlomo Berkovsky, Xiaoshui Huang, Lin Xiao, and Wenpeng Lu. 2020. Jointly modeling intra- and inter-transaction dependencies with hierarchical attentive transaction embeddings for next-item recommendation. *IEEE Intell. Syst.* (2020), 1–7. https://doi.org/10.1109/MIS.2020.2997362

[117] Shoujin Wang, Liang Hu, and Longbing Cao. 2017. Perceiving the next choice with comprehensive transaction embeddings for online recommendation. In *ECML-PKDD*. Springer, 285–302.

[118] Shoujin Wang, Liang Hu, Longbing Cao, et al. 2018. Attention-based transactional context embedding for next-item recommendation. In *AAAI*. 2532–2539.

[119] Shoujin Wang, Liang Hu, Yan Wang, et al. 2019. Sequential recommender systems: challenges, progress and prospects. In *IJCAI*. AAAI Press, 6332–6338.

[120] Shoujin Wang, Liang Hu, Yan Wang, et al. 2020. Intention nets: Psychology-inspired user choice behavior modeling for next-basket prediction. In *AAAI*. 6259–6266.

[121] Shoujin Wang, Liang Hu, Yan Wang, et al. 2020. Intention2Basket: A neural intention-driven approach for dynamic next-basket planning. In *IJCAI*. 2333–2339.

[122] Shoujin Wang, Liang Hu, Yan Wang, Xiangnan He, et al. 2021. Graph learning based recommender systems: A review. In *IJCAI*. 1–9.

[123] Shoujin Wang, Liang Hu, Yan Wang, Quan Z. Sheng, Mehmet Orgun, and Longbing Cao. 2019. Modeling multi-purpose sessions for next-item recommendations via mixture-channel purpose routing networks. In *IJCAI*. AAAI Press, 3771–3777.

[124] Shoujin Wang, Gabriella Pasi, Liang Hu, and Longbing Cao. 2020. The era of intelligent recommendation: Editorial on intelligent recommendation with advanced AI and learning. *IEEE Intell. Syst.* 35, 5 (2020), 3–6.

[125] Wen Wang, Wei Zhang, Shukai Liu, et al. 2020. Beyond clicks: Modeling multi-relational item graph for session-based target behavior prediction. In *The Web Conference*. 3056–3062.

[126] Zhitao Wang, Chengyao Chen, et al. 2018. Variational recurrent model for session-based recommendation. In *CIKM*. 1839–1842.
[127] Chen Wu and Ming Yan. 2017. Session-aware information embedding for e-commerce product recommendation. In *CIKM*. ACM, 2379–2382.
[128] Chao-Yuan Wu, Amr Ahmed, Alex Beutel, Alexander J Smola, and How Jing. 2017. Recurrent recommender networks. In *WSDM*. ACM, 495–503.
[129] Shu Wu, Yuyuan Tang, Yanqiao Zhu, et al. 2019. Session-based recommendation with graph neural networks. In *AAAI*. 346–353.
[130] Xiang Wu, Qi Liu, Enhong Chen, Liang He, et al. 2013. Personalized next-song recommendation in online karaokes. In *RecSys*. ACM, 137–140.
[131] Zhuoran Xiong, Xiao-Yang Liu, Shan Zhong, Hongyang Yang, and Anwar Walid. 2018. Practical deep reinforcement learning approach for stock trading. arXiv:1811.07522. Retrieved from https://arxiv.org/abs/1811.075229059.
[132] Chengfeng Xu, Pengpeng Zhao, Yanchi Liu, et al. 2019. Graph contextualized self-attention network for session-based recommendation. In *IJCAI*. 3940–3946.
[133] Liang Yan and Chunping Li. 2006. Incorporating pageview weight into an association-rule-based web recommendation system. In *AI*. Springer, 577–586.
[134] Ghim-Eng Yap, Xiao-Li Li, and S. Yu Philip. 2012. Effective next-items recommendation via personalized sequential pattern mining. In *DASFAA*. Springer, 48–64.
[135] Mao Ye, Xingjie Liu, and Wang-Chien Lee. 2012. Exploring social influence for recommendation: a generative model approach. In *SIGIR*. 671–680.
[136] Haochao Ying, Fuzhen Zhuang, Fuzheng Zhang, et al. 2018. Sequential recommender system based on hierarchical attention network. In *IJCAI*. 3926–3932.
[137] Jiaxuan You, Yichen Wang, Aditya Pal, Pong Eksombatchai, et al. 2019. Hierarchical temporal convolutional networks for dynamic recommender systems. In *WWW*. 2236–2246.
[138] Feng Yu et al. 2020. TAGNN: Target attentive graph neural networks for session-based recommendation. In *SIGIR*. 1–5.
[139] Feng Yu, Qiang Liu, et al. 2016. A dynamic recurrent model for next basket recommendation. In *SIGIR*. ACM, 729–732.
[140] Fajie Yuan, Xiangnan He, Haochuan Jiang, Guibing Guo, et al. 2020. Future data helps training: modeling future contexts for session-based recommendation. In *The Web Conference*. 303–313.
[141] Fajie Yuan, Alexandros Karatzoglou, et al. 2019. A simple convolutional generative network for next item recommendation. In *WSDM*. 582–590.
[142] Shuai Zhang, Yi Tay, Lina Yao, Aixin Sun, and Jake An. 2019. Next item recommendation with self-attentive metric learning. In *RecNLP*. 1–9.
[143] Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. 2019. Deep learning based recommender system: A survey and new perspectives. *Comput. Surv.* 52, 1 (2019), 1–38.
[144] Zhiyong Zhang and Olfa Nasraoui. 2007. Efficient hybrid Web recommendations based on Markov click stream models and implicit search. In *WI*. 621–627.
[145] Wei Zhao, Wenyou Wang, Jianbo Ye, Yongqiang Gao, et al. 2017. Leveraging long and short-term information in content-aware movie recommendation. arXiv:1712.09059. Retrieved from https://arxiv.org/abs/1712.09059.
[146] Xiangyu Zhao, Long Xia, Liang Zhang, Zhuoye Ding, Dawei Yin, and Jiliang Tang. 2018. Deep reinforcement learning for page-wise recommendations. In *RecSys*. 95–103.
[147] Xiangyu Zhao, Liang Zhang, Long Xia, Zhuoye Ding, Dawei Yin, and Jiliang Tang. 2017. Deep reinforcement learning for list-wise recommendations. arXiv:1801.00209. Retrieved from https://arxiv.org/abs/1801.00209.
[148] Yan Zhao, Shoujin Wang, Yan Wang, Hongwei Liu, and Weizhe Zhang. 2020. Double-wing mixture of experts for streaming recommendations. In *WISE*. Springer, 269–284.
[149] Elena Zheleva, John Guiver, Eduarda Mendes Rodrigues, et al. 2010. Statistical models of music-listening sessions in social media. In *WWW*. 1019–1028.
[150] Fan Zhou, Zijing Wen, Kunpeng Zhang, Goce Trajcevski, and Ting Zhong. 2019. Variational session-based recommendation using normalizing flows. In *WWW*. 3476–3482.
[151] Feng Zhu, Chaochao Chen, et al. 2019. DTCDR: a framework for dual-target cross-domain recommendation. In *CIKM*. 1533–1542.
[152] Feng Zhu, Yan Wang, Chaochao Chen, et al. 2021. Cross-domain recommendation: challenges, progress, and prospects. arXiv:2103.01696. Retrived from https://arxiv.org/abs/2103.01696.
[153] Lixin Zou, Long Xia, et al. 2020. Pseudo dyna-Q: A reinforcement learning framework for interactive recommendation. In *WSDM*. 816–824.

Received August 2020; revised March 2021; accepted May 2021