A Survey on Session-based Recommender Systems

SHOUJIN WANG, University of Technology Sydney; Macquarie University, Australia
LONGBING CAO, University of Technology Sydney, Australia
YAN WANG, Macquarie University, Australia

Session-based recommender systems (SBRS) are an emerging topic in the recommendation domain and have attracted much attention from both academia and industry in recent years. Most of existing works only work on modelling the general item-level dependency for recommendation tasks. However, there are many more other challenges at different levels, e.g., item feature level and session level, and from various perspectives, e.g., item heterogeneity and intra- and inter-item feature coupling relations, associated with SBRS. In this paper, we provide a systematic and comprehensive review on SBRS and create a hierarchical and in-depth understanding of a variety of challenges in SBRS. To be specific, we first illustrate the value and significance of SBRS, followed by a hierarchical framework to categorize the related research issues and methods of SBRS and to reveal its intrinsic challenges and complexities. Further, a summary together with a detailed introduction of the research progress is provided. Lastly, we share some prospects in this research area.

CCS Concepts: • **Surveys and overviews**;

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

ACM Reference Format:
Shoujin Wang, Longbing Cao, and Yan Wang. 2018. A Survey on Session-based Recommender Systems. 1, 1 (February 2018), 34 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

Recommender systems (RS) have evolved into a fundamental tool for helping users make informed decisions and choices, especially in the era of big data in which customers have to make choices from a large number of products and services. A lot of RS models and techniques have been proposed and most of them have achieved great success. Among them, the content-based RS [6, 107] and collaborative filtering RS [37, 115] are two representative ones. Their efficacy has been demonstrated by both research and industry communities.

However, these aforementioned conventional RS still have some drawbacks. A critical one is that they only focus on a user's long-term static preference while ignoring his or her short-term transactional patterns, which results in missing the user's preference shift through the time. In this case, the user's intent at a certain time point may be easily submerged by his or her historical shopping behaviours, which leads to unreliable recommendations. This is because these RS usually break down a basic transaction unit (e.g., a session) into multiple records at smaller granularity (e.g., user-item interaction pairs) and then mix all these records. Such splitting actually has destroyed the intrinsic nature of a transactional behavior, in which a user's preference shift is embedded. For

Authors' addresses: Shoujin Wang, University of Technology Sydney; Macquarie University , Advanced Analytics Institute, 2 Blackfriars St, Chippendale, NSW, 2008, Australia, shoujinwang@foxmail.com; Longbing Cao, University of Technology Sydney, Sydney, Australia, Longbing.cao@uts.edu.au; Yan Wang, Macquarie University, Sydney, Australia, yan.wang@mq.edu.au.

XXXX-XXXX/2018/2-ART $15.00
https://doi.org/10.1145/1122445.1122456

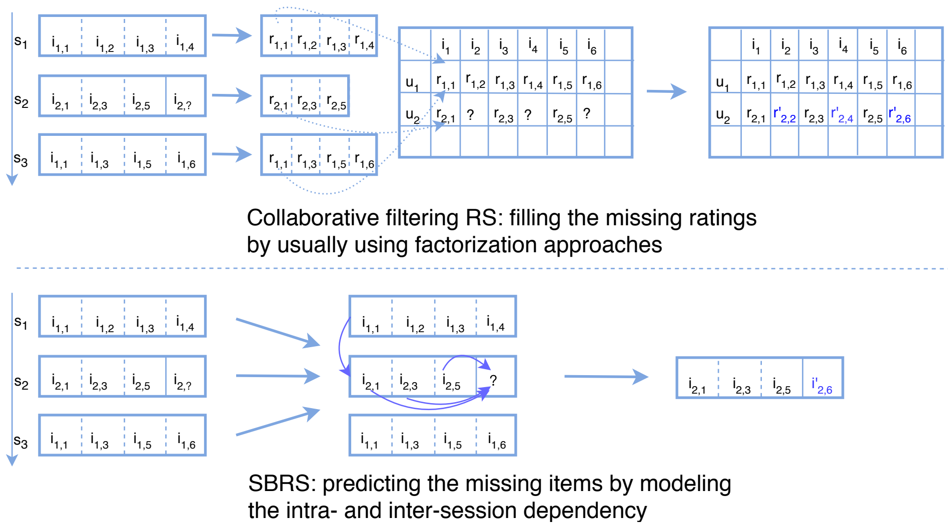

Fig. 1. Comparison between collaborative filtering RS and SBRS

example, in the matrix factorization [49, 71], which is a representative model for collaborative filtering RS, the items a user bought in all transactions are put into one row of a matrix as shown in the top half of Figure 1. Another typical drawback is that user IDs are not always available due to privacy issues in some cases, therefore the conventional RS which require user information are not applicable. In such cases, the recommendations can be only made on the transactional data without specific user information.

To address the above issues, it is necessary to take the transactional structure into account to capture richer information in recommendation. In other words, it is necessary to learn user transactional behaviour patterns and the user preference shift from one transaction to another. To this end, SBRS were proposed recently. Here, a *session* can be regarded as a transaction with multiple purchased items in one shopping event. Different from content-based and collaborative filtering RS, SBRS comprehensively consider the information embedded from one session to another and take a session as the basic unit for recommendation, as shown in the bottom half of Figure 1. In this way, SBRS aim to maximally reduce the information loss caused by ignoring or breaking down session structures as in some of existing approaches.

Besides the transactional domain, SBRS are widely applied in other domains like next web page recommendations, next POI recommendations, tourism recommendations, next song recommendations, next movie recommendations, and so on. To cover these various domains, in this work, the concept "session" is not limited to a transaction, instead, it refers to a collection of consumed or visited elements at one time or in a certain period of time. For instance, the web pages visited by a user in one internet surf can be gathered into a session, and the songs listened by a user in an hour or in a day can also be grouped into a session.

Except for the difference illustrated in the second and third paragraphs, SBRS also differ from other RS at multiple aspects. To have a substantial understanding of such differences, we present a comprehensive comparison between SBRS and other typical ones in Table 1.

In this survey, we give a comprehensive and systematic overview of the session-based recommendation scenarios, which take a session as the basic unit for recommendation, and the corresponding technique: SBRS. This is actually a relative novel recommendation paradigm investigated in recent years in the relevant communities. The survey provides a basic foundation and a comprehensive view of SBRS with a rich list of relevant resources for the community.

The contributions of this work are multifold.

- We systematically formalize the issues of SBRS and the corresponding work mechanisms, which provides a in-depth and comprehensive understanding of this new recommendation paradigm. In addition, we thoroughly demonstrate the value and significance of SBRS from both theoretical and practical perspectives.
- We introduce a hierarchical framework for the modeling of SBRS, which is believed to cover nearly all the components and their relations associated with a session structure, to the best of our understanding. Further, the special data characteristics, the critical problem complexities and the key challenges in SBRS have been thoroughly revealed and illustrated. To the best of our knowledge, this is the first work to systematically categorize the above aspects of SBRS.
- We review the current research progress of SBRS by providing a systematic categorization of the existing work from two dimensions: the research issue perspective and the technical perspective. Following the technical categorization, a detailed introduction to the representative approaches in each category is created. This brings about not only an overview of the progress made so far but also the necessary technical details of solutions.
- By comparing the research progress to the SBRS problem complexities and challenges, we offer several prospects of future SBRS research.

The rest of this survey is organized as follows. We first define the research scenarios and key concepts, and then formalize the research issues in Section 2. We then present the significance, complexities and key challenges of SBRS in Section 3, followed by an overview of the SBRS research evolution and the attention paid by the related research communities to SBRS in Section 4. Further, in Section 5, we categorize SBRS from the research issue and technical perspectives respectively. Following the technical categorization, we review two main categories of approaches in the subsequent Sections 6 and 7 respectively. Lastly, we offer several future prospects of SBRS in Section 8 and conclude the work in Section 9.

2 FORMALIZATION AND NOTATIONS

In this section, we first define the core concepts *session* and *SBRS*. Then, we formalize the SBRS task and accordingly give some critical relevant definitions and notations that will be used in this paper.

The English oxford living dictionary defines *session* as "a meeting of an official body, especially a legislature, council, or court of law, to conduct its business" [34]. In this work, we have expanded the concept of session to garner more general and specific meaning related to recommendation.

Definition 2.1 (Session). A *session* is a set of items (e.g., referring to any objects, e.g., products, songs or movies) that are collected or consumed in one event (e.g., a transaction) or in a certain period of time or a collection of actions or events (e.g., listening to a song) that happened in a period of time (e.g., one hour).

For instance, both a set of items purchased in one transaction and a list of songs listened by a user in one hour can be viewed as a session. In addition, the web pages that a user successively clicked in one hour can also be regarded as a session.

An *SBRS* is a RS that is built on the base of sessions and takes a session as the basic data organization unit for analyzing the data and making recommendation. Here session differentiates SBRS from other representative recommendation models like content-based and collaborative filtering RS, in which the content in a session is usually split into smaller units, e.g., a single item or a user-item pair, forming their basic data organization units. Accordingly, we formally define the learning task of SBRS below.

Definition 2.2 (Session-based recommender systems (SBRS)). Given partially known session information, e.g., part of a session or recent historical sessions, an SBRS aims to predict the unknown part of a session or the future sessions based on modelling the complex relations embedded within a session or between sessions.

Table 1. Comparison Between SBRS and Other RS

RS	Input	Core assumption	Work mechanism	Pros	Cons
Content-based filtering (CBF) [125]	User, item content information	A user likes what he/she liked	Matching up user profile against item content	Simple and straightforward, can handle cold-start issues	The assumption may not fit real-world cases well
Collaborative filtering (CF) [115]	User-item interaction data	A user likes what he/she liked	Modeling user-item interactions	Effective and relatively simple	Easily suffering from sparsity and cold-start issues
Context-aware RS [5]	Users, items, context and user-item interaction data	A user may have different preferences under different contexts	Modeling user-item-context interactions	Incorporating more information and fitting the real-world cases better	Data availability and sparsity issues
SBRS [125]	Session data	User preference changes along with the corresponding session context	Recommending items that have occurred in a similar context	Considering the user preference evolution, which fits the real-world cases better	Ignoring user's general and long-term preference

Accordingly, SBRS can be grouped into two research directions: next-item(s) recommendations that recommend a part of the current session, and next-session (e.g., next-basket) recommendations that recommend partial or whole of future sessions. Both will be formally defined later in this work.

In general, a recommendation application, e.g., shopping-basket-based transactions, involves two basic objects and concepts: *user* u and *item* i, and all the users and items in a dataset constitute the user universal set $U = \{u_1, u_2, ..., u_{|U|}\}$ and item universal set $I = \{i_1, i_2, ..., i_{|I|}\}$ respectively. The interactions between users and items, e.g., "click" or "buy", form another important component in RS. The user-item interactions within a period of time form the foundation of a session. For instance, the clicks made by a user on all items in an online shopping transaction form a click session; the items purchased by a user in a shopping event form a transactional session, which is also called a *transaction*. In general, a collection of items that are interacted by a certain user during a certain period (e.g., one day) or in a certain event (e.g., one shopping visit) constitute a session $s = \{i_1, i_2, ..., i_{|s|}\}$. All the sessions together in one dataset form the session universal set $S = \{s_1, s_2, ..., s_{|S|}\}$. Generally, an SBRS makes the unknown session information as its target $\boldsymbol{t}$ to be predicted by taking the prior session information as the context and condition, which is called a *session context* C in this work. A session context can be either an *intra-session context* C^{Ia} or an *inter-session context* C^{Ie} according to whether the session context comes only from one session or across multiple sessions.

Definition 2.3 (Intra-session context). Taking session s_n as the current session for making recommendations (recommending unknown item i_t in s_n), an intra-session context C^{Ia} is the set of items that are already known in s_n, namely $C^{Ia} = \{i | i \in s_n, i \neq i_t\}$.

Definition 2.4 (Inter-session context). Taking session s_n as the current session for recommendations, an inter-session context C^{Ie} is the set of recent sessions happened before s_n, namely $C^{Ie} = \{s_{n-1}, s_{n-2}, ..., s_{|C^{Ie}|}\}$.

We differentiate the above two kinds of session contexts because they actually involve different kinds of relations for the recommendation task: the intra-session context embeds the intra-session dependency while the inter-session context mainly conveys the inter-session dependency.

Now we specify and formulate the session-based recommendation task as follows.

Definition 2.5 (Session-based recommendation task). Given a session context C, the session-based recommendation is to learn a function f which maps C to the recommendation target $\boldsymbol{t}$: $\boldsymbol{t} \Leftarrow f(C)$. Note that, a session context is the dominant information for session-based recommendations; sometimes, additional information like item features and user features may also be incorporated into session-based recommendations as complementary information.

As discussed above, SBRS can be generally categorized into two main branches: next-item recommendations, and next-session recommendations.

Definition 2.6 (Next-item(s) recommendations). Given an intra-session context C^{Ia} over the current session s_n, the next-item(s) recommendations predict the next item(s) i_t in s_n conditional on C^{Ia}. It should be noted that most next-item(s) recommendations only take the current session into account [31, 51], a minority of work incorporates recent sessions into the context as an addition [111].

Definition 2.7 (Next-session (next-basket) recommendations). Given a inter session context C^{Ie} for current session s_n, the next-session recommendations predict those items possibly occurring in session s_n.

The notations used in this paper are listed in Table 2 below.

3 SIGNIFICANCE, COMPLEXITY AND KEY CHALLENGES

3.1 Values and Significance

SBRS are of great significance from both research and business perspectives.

From the research perspective, an SBRS can effectively keep the natural characteristic (i.e., session structure) of session data and avoid local information loss to the maximum extent, as demonstrated in Section 1. For example, the item co-occurrence information may be lost if session structures are not considered; and the user short-term preferences, usually indicated by item occurrence in a session, are lost as well. Such local transaction information is quite critical for reliable recommendations in a specific transactional event. The reasons are varied, without session information, (a) it is easy to generate duplicated recommendations, namely to recommend items similar or identical to those already in hand; (b) the user shopping behaviour patterns are lost and thus it is hard to generate personalized recommendations that match a user's specific preference well; (c) the user preference shift is lost, and thus it is impossible to capture a user's current preferences to make timely recommendations; user preferences are often dynamic rather than static and may be different from one session to another; and (d) it is difficult to capture local and short-term user preferences for reliable recommendations, which are shown only in very recent sessions rather than in all sessions. In practice, the aforementioned RS usually only capture the global and long-term user preferences.

By remaining the intrinsic structure of session data and taking it as a basic data unit, an SBRS keeps all the local information, which is ignored by other RS including collaborative filtering. As a result, SBRS can easily fix the

Table 2. Notation List

Notation	Meaning
f	a feature
i	an item
I	an itemset
t	a transaction
T	a transaction set
s	a session
S	a session set
q	a sequence
Q	a sequence set
u	an user
U	an user set
C	a context
p	a pattern
PT	a pattern set
P	a probability
v	a value (represented by a lowercase letter)
$\mathbf{v}$	a vector (represented by a bold lowercase letter)
$\mathbf{W}$	a matrix (represented by a bold uppercase letter)

defects of other RS and thus provide more reliable recommendations by focusing more on the dynamic and local aspects of session data. As the items already chosen in the current or recent sessions are considered, SBRS more easily avoid recommending items that are similar or identical to them; as a result, the duplicated recommendations are avoided. Furthermore, as each user's purchased items are organized into sessions, it becomes easy to know his or her transactional patterns within and between sessions by analyzing item distributions within and across sessions. In addition, only the current or recent sessions, rather than the whole item pool from all sessions, are imported into an SBRS, which enables the possibility of paying attention to a short-term and local user preferences at each specific time point. This actually makes the recommendation results more timely, focused and reliable. More importantly, it is easier to capture a user's preference shift by analyzing the item changes from one session to another in an SBRS. In practice, this treats user preferences as dynamic rather than static, to more closely match the real-world scenarios.

From the business perspective, SBRS are even more important than other RS. Essentially, the availability of session data is much stronger than the data (e.g., item features or ratings) required by other RS. This provides SBRS with a much wider application range. In practice, SBRS can be applied to any applications where transactional data are recorded.

3.2 Data Characteristics and Complexity

Though extremely valuable and significant, SBRS are yet quite challenging and complex. In practice, there is a hierarchical structure consisting of five main levels in a typical session dataset, from the feature value level till to the domain level, as shown in Fig. 2. Out of these five levels, the three levels in the middle (i.e., feature level, item level and session level) are the core in most of session learning models. The logical relations between different components of this structure are as follows: items are the key concept and the most important component. The

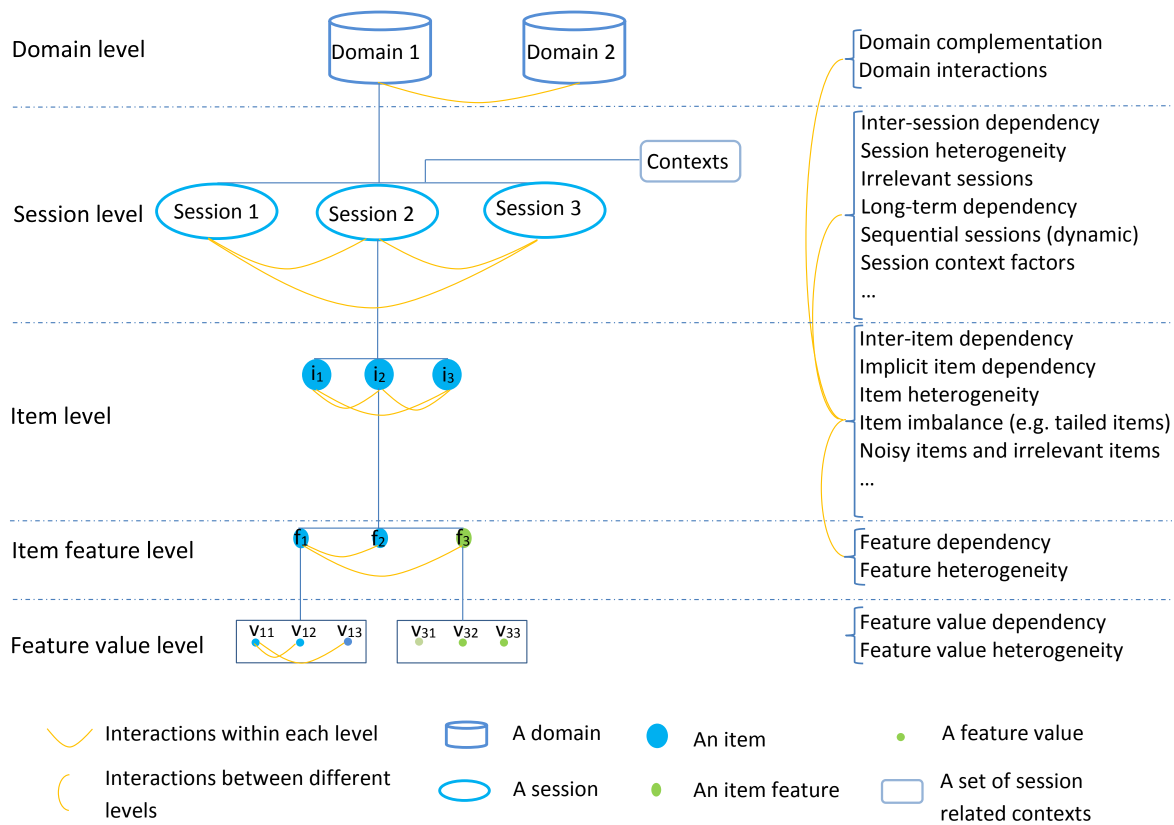

Fig. 2. A hierarchical structure of session data and the specific challenges in SBRS

reasons are twofold: on one hand, an item is the atomic granularity in a session dataset, other components related to items are either to describe items (e.g., features) or to organize items (e.g., sessions). On the other hand, items usually play the main role in most of session-related models including SBRS. For instance, the goal of an SBRS is to recommend the next item or items. It should be noticed that we use item here because most SBRS focus on transactional domain, an item can be easily replaced by other entity, such as a song in the music domain, or a web-page browse in the internet domain.

Each item is often described by multiple heterogeneous features, such as item category, price and place of production. Each feature usually contains multiple values, and some values may be frequent, while others may be not. Furthermore, a collection of relevant items typically forms a session (e.g., a transaction consisting of multiple items purchased together). In most cases, the relevance between items is built on their co-occurrences. Generally, a dataset collected from one domain (e.g., food) contains multiple sessions, for instance, a transactional dataset contains thousands of transaction records. It should be noticed that in some minority cases, a session may contain items from multiple domains to form a mixed-domain session.

3.3 Key Challenges

Corresponding to the aforementioned hierarchical structure of session data, there are a variety of typical complexity and critical challenges in each level of the structure as listed on the right hand side of Fig. 2. We first provide an overview of these challenges and then categorize the low-level specific challenges in terms of the intrinsic nature of the SBRS problem.

3.3.1 An Overview of Challenges in SBRS. Driven by the own characteristics of session data, the challenges in SBRS mainly come from three overall aspects: the inner-session aspect (i.e., internal aspect of a session), the inter-session aspect, and the outer-session aspect (i.e, external aspect of a session). In addition, these aspects interact with each other, as shown in Fig. 3. We illustrate these three aspects of challenges one by one below.

(a) *Inner-session challenges.* Inner-session challenges refer to the challenges inside a session. As shown in Section 3.2 and Fig. 2, there is a quite complex structure within a session. To be specific, a hierarchical structure consists of multiple levels, from items contained in this session to the corresponding features of the items, and to the corresponding values of these features. Accordingly, inner-session challenges include all the challenges on the item level, the feature level, the feature value level and the interactions between these levels, which are listed on the right side of Fig. 2. The specific challenges on each level will be demonstrated in detail in Section 3.3.2.

(b) *Inter-session challenges.* Inter-session challenges mean the challenges associated with the interactions between sessions, containing the challenges on the session level listed on the right side of Fig. 2. Some typical challenges include the session heterogeneity, the sequential dependency between sessions, and the dynamics of sessions. The detailed illustrations of these challenges are provided in Section 3.3.2.

(c) *Outer-session challenges.* Outer-session challenges mainly contain three classes: the challenges on the domain level, the challenges of modelling the context associated with a session, and the interactions between the outer-session challenges and the inter-session challenges, as illustrated in Fig. 3. Specific challenges in each class are detailed in Fig. 2 and Section 3.3.2.

The *session-associated context* refers to the circumstance where a session happens. It mainly contains the contextual factors such as time, venue, weather, season and users. Context should be considered in an SBRS because different contexts can bring uncertainty and dynamics to the session evolution [73] and thus make a huge difference on a user's real-time preferences [4]. The corresponding challenges lie in the heterogeneity of different contextual factors and the complex coupling relationships between them [17]. For instance, it is quite challenging to appropriately model the collaborative effect between venue and season since much uncertainty exists here.

3.3.2 A Categorization of Challenges in SBRS. Generally, all the specific challenges on all levels in SBRS can be divided into the following four categories: heterogeneity, coupling, other complexities, and their interactions [17, 19, 21].

(a) *The heterogeneity within each level*: refers to different elements on each level that has different characteristics and thus cannot be treated equally or be modelled in the same way.

- Value heterogeneity: the feature values from one feature are often distributed differently and thus cannot be treated equally. For instance, some values may be more frequent than others.
- Feature heterogeneity: an item usually contains different types of features, including categorical and numerical ones, these heterogeneous features cannot be modelled in the same way.
- Item heterogeneity: items in a session also have different distributions (e.g., some are quite popular while most of others may rarely occur). Particularly, the contextual items in an SBRS are usually heterogeneous due to their different relevance scales to the next item(s).

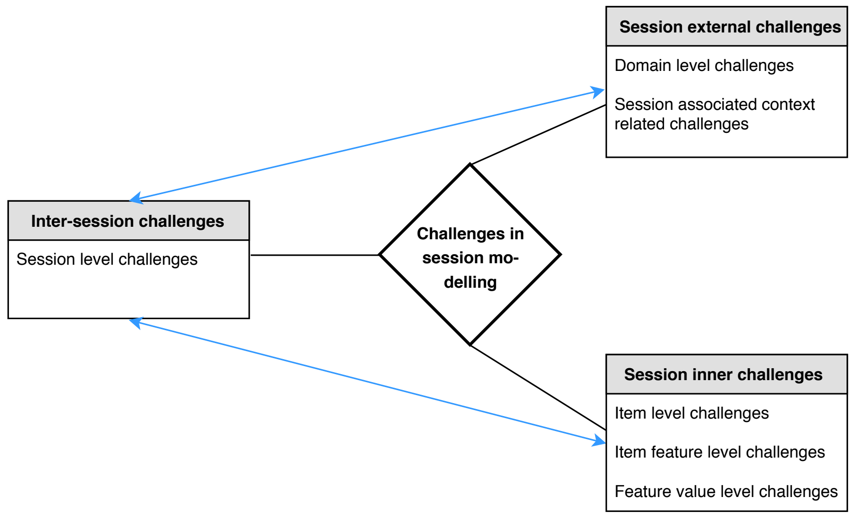

Fig. 3. An overview of challenges in SBRS

- Session heterogeneity: similarly, the session heterogeneity mainly refers to that in a sequence of sessions, different contextual sessions relate to the current session with different scales; some are even irrelevant or noisy sessions.
- Context heterogeneity: the various types of contextual factors, e.g., time, venue, season, and user, contribute to the session evolution collaboratively but differently; they cannot be modelled with a unified method.

(b) *The couplings within each level*, which are indicated by the yellow lines within each level. Here *couplings* [17] refer to the interactions that the different elements on each level are somehow dependent on each other, and there are diverse interactions between them.

- Value-level coupling: it contains two different types of value couplings: intra-feature value coupling and inter-feature value coupling. The former refers to the interactions between different values from the same feature (e.g., values with similar frequencies are regarded similar w.r.t. their importance [41, 133]) while the latter means the interactions between values from different features (e.g., the category of an item may have some effect on its price).
- Feature-level coupling: it indicates one feature has influence on another, or the combination of several features infers a particular pattern, revealing a special semantic meaning; some examples can be found in [103].
- Item-level coupling: this indicates the common interactions over items within a session. For instance, some items (e.g., bread and milk) in a supermarket are usually bought together.
- Session-level coupling: it means the interactions between different sessions. In real-world transactions, the recent sessions usually have some impact on the current one. For instance, a user who has bought a car a week ago may have a car insurance purchased in the current transaction.
- Domain-level coupling: it refers to the interactions between different domains. Suppose a user likes the movie "Titanic", then he may love the topic song "My heart will go on" in this movie. In this case, the two domains (movie and song) are interactive.

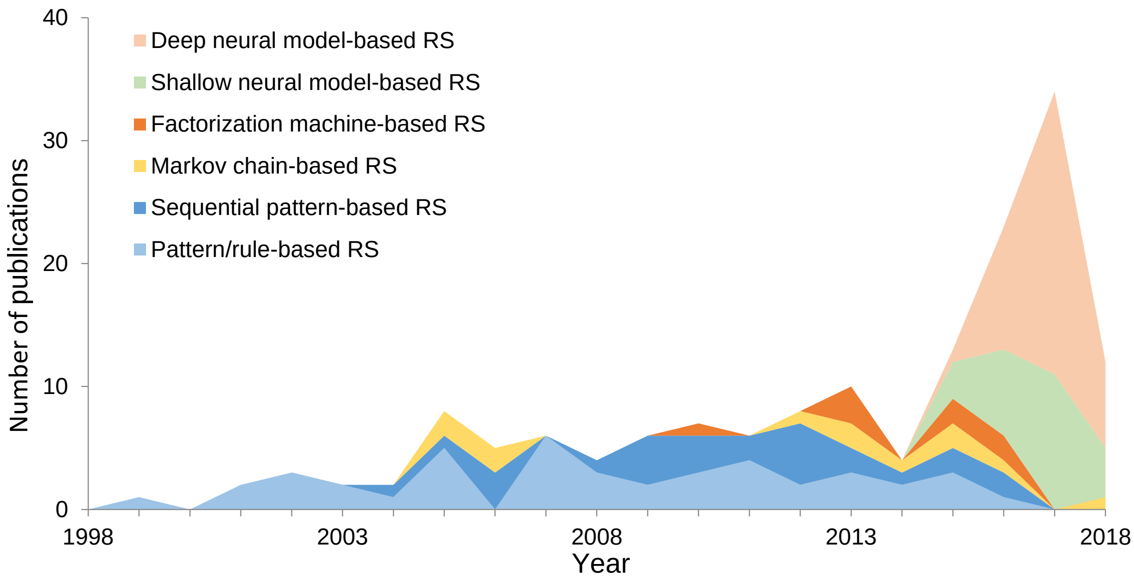

Fig. 4. Number of publications of SBRS integrated from different categories of approaches in each year [1]

- Contextual coupling: it indicates the interactions between different contextual factors. For instance, a user may prefer different items in a shopping event happened on a weekend in the winter from those in his/her shopping list on a weekday in the summer.

(c) *Other complexities within each level*: except for the typical heterogeneity and coupling issues on each level, there are other complexities [20, 21]. For instance, on the item level, the complexities may include the implicit item dependency, the item imbalance, and so on; on the session level, there may be challenges including the long-term dependency over sessions, and the session modeling under certain context factors like venue, time and users.

(d) *The interactions between different levels*: various interactions may exist across different levels. Here we list the two most typical ones.

- Feature-item interaction: the features of an item usually influences its occurrences in a transaction. For instance, items belonging to the same category are more likely to occur together. The good modelling of this interaction could benefit the cold-start item issues, in which item occurrence data is quite limited.
- Session-item interaction: other sessions may lead to whether an item occur or not in the current session. For instance, a user who has bought a house recently may buy some furniture in the current transaction.

4 AN OVERVIEW OF SBRS

In this section, a comprehensive overview of SBRS is given. Specifically, we first briefly review the evolutionary history of SBRS, and then discuss about the attention attracted from the community.

4.1 A Brief Evolutionary History of SBRS

[1]The data was retrieved by using Google scholar and updated till July 2018.

The research on SBRS has attracted much attention since 1990s [8], and was labelled under different names: pattern-based RS, using patterns for recommendations, rule-based RS, sequence-based RS, sequential recommendations, transaction-based RS, session-aware RS, next-item recommendations, next-basket reocmmendations, next-song recommendations, next-movie recommendations, and next-POI recommendations, etc.

We treat the above topics as part of SBRS, and group the relevant work on SBRS into two clearly different stages: the model-free stage from the late of 1990s to the early of 2010s, and the model-based stage from the early of 2010s till now. The first stage was driven by the development of data mining techniques, including pattern mining, association rule discovery, and sequence mining. Subsequently, it was dominated by pattern/rule-based and sequence-based recommendation research. According to the literature review, we found the middle of 2000s witnessed the peak of this stage, and many relevant works were published during this period. The second stage was driven by the development of statistics and machine learning techniques, especially some time-series-related models like Markov chain models, recurrent neural network (RNN) models, etc. Thanks to the fast development of deep learning techniques in recent years, model-based RS have reached its peak since 2017. Many researchers rushed into this area and developed various neural models for the next-item or next-basket recommendations in the past two years. A statistic of publications on SBRS by year is given in Fig. 4.

4.2 Attention in the Research Community

Today, work related to SBRS appears widely in several top conferences on data mining like KDD, CIKM and WSDM, on machine learning and artificial intelligence like AAAI, IJCAI and ECML, and on their applications like Recsys, WWW and SIGIR.

In addition to the publications on general RS topics that were scattered in a variety of top conferences, there are also special events which particularly focus on SBRS. For example, there is a separate session called SBRS in ACM RecSys 2017 [2] and related keynote speeches on SBRS[3]. SBRS also accounts for a large portion of a workshop called deep learning in RS [70] held in RecSys 2016 and another workshop [52] together with a tutorial [69] on the same topic in the same conference in 2017.

5 CATEGORIZATION AND SUMMARIZATION

We categorize SBRS research from different perspectives to build an in-depth and comprehensive understanding. To be specific, we first categorize SBRS from the research issue perspective, followed by the categorization from the technique perspective. In each category, we briefly summarize the the corresponding progress.

5.1 Categorization from the Research Issue Perspective

Generally speaking, the SBRS research often focuses on what to recommend and how to recommend. What to recommend actually talks about the research tasks and scenarios, which are usually set in priority. More efforts are devoted into the second one – how to recommend, which is closely connected to the challenges demonstrated in Section 3.3.

5.1.1 What to Recommend: A categorization of recommendation scenarios and settings. Usually, the session data acquired by an SBRS can be categorized into two types. The first one is the shopping basket-like data (e.g., the Tmall dataset[4]), which has a clear intrinsic session structure. For instance, each shopping basket has clear boundaries which distinguish it from another. In such a case, a shopping basket is a natural session. The recommendation tasks on such a type of data can be divided into next-item(s) recommendations and next-basket recommendations, according to whether recommend items for the current session or for the next one.

[2]https://recsys.acm.org/recsys17/session-4/

[3]e.g., at the complexRec workshop at ACM RecSys 2017 [60] and at the 41st German conference on artificial intelligence [61]

[4]https://ijcai-15.org/index.php/repeat-buyers-prediction-competition

The other is the event history-like data, in which the original data is usually organized as a collection of event records, such as the movie data (e.g., MovieLens dataset[5]), and POI data (e.g., Foursquare dataset[6]). Different from the shopping basket data, the event history data usually does not have a natural session structure; in other words, there is no clear boundary to split events into different sessions. For example, different from a shopping basket, which usually consists of multiple items purchased by a user at one time, a user usually only watches one movie each time. Therefore, the session characteristic is fuzzier and the dependency in such data may be not so strong as that in the shopping basket-like data. This kind of data is usually split into multiple sessions manually by using certain techniques like time sliding window in order to make session-based recommendations. Accordingly, The recommendation tasks on such kind of data are summarized as next-event/action recommendations. Next, we first illustrate these three recommendation scenarios in detail, followed by a summary of the up-to-date progress in each scenario.

(a) *Next-Item(s) Recommendations.* As defined in Section 2, next-item(s) recommendations are to recommend the next one or several items within a session (usually a shopping-basket) by mainly modelling the intra-session dependency. Next-item(s) recommendations are the main stream and the most common setting of session-based recommendations. Recently, next-item(s) recommendations have attracted much attention and much progress has been achieved as summarized in Table 3.

(b) *Next-Basket Recommendations.* As defined in Section 2, next-basket recommendations are to recommend the items that would occur in the next session (e.g., a shopping-basket) by mainly modelling the inter-session dependency (sometimes the intra-session dependency is also incorporated). Compared to next-item(s) recommendations, next-basket recommendations have received less attention and some progress has been achieved in this scenario as summarized in Table 3.

(c) *Next-Event/Action Recommendations.* Next-event/action recommendations are to recommend the next event or action (e.g., to watch a movie or to listen to a song) by modelling the intra- or inter-session dependency, or both. These are another significant case of session-based recommendations which are usually built on datasets without natural session structures such as the movie or POI data. Many researchers have paid attention to different applications of next-event/action recommendations, including next-song recommendations, next-movie recommendations, and next-POI recommendations, and much progress has been achieved as summarized in Table 3.

5.1.2 How to Recommend: A Categorization of recommendation approaches.

A major driving force in SBRS is *couplings*, or more specifically *dependency* [139, 143, 147]. This is because of the particular session-based recommendation setting: given a session context, recommendations are made by finding those item(s) or events that are most dependent on the context as the next choice. Naturally, the answer to the critical question "how to recommend" is to model the dependency embedded in the session data. Essentially, there is quite complex dependency within each level (cf. Fig. 2) or between different levels in the session data. Correspondingly, the approaches to session-based recommendations generally model the complex dependency. Based on the hierarchical structure of session data illustrated in Section 3.3, the relevant approaches can be divided into five branches. Each branch corresponds to the modelling of dependency on one level.

(a) *Item-level Dependency Modeling.* The item-level dependency modeling refers to the modelling of inter-item or inter-event dependency within a session for next item(s) or next event recommendations. This is also called intra-session dependency modelling. Currently, most existing work in SBRS falls into this branch.

There are several typical issues in the item-level dependency modelling which could make a huge difference on the quality of dependency modelling and the performance of recommendations.

[5]https://movielens.org/
[6]https://sites.google.com/site/yangdingqi/home/foursquare-dataset

Table 3. Different Session-based Recommendation Scenarios

Scenarios	Research progress	Related work
Next-item(s) recommendations	Much progress has been made in this scenario, including intra- and inter-session dependency modelling and feature-to-item interaction modelling with various approaches including pattern-based approaches, factorization machines, Markov chain models, and neural models.	[14, 25, 32, 36, 40, 45, 50, 51, 53, 57, 63, 64, 82, 108, 111, 113, 124, 140, 141, 149, 150]
Next-basket recommendations	Some progress has been made to model inter-session dependency with typical approaches including factorization machines and neural models.	[112, 132, 138, 151]
Next-event/action recommendations	Some progress has been made to model sequential or non-sequential inter-event dependency with typical approaches including factorization machines, latent Markov models (also called metric embeddings), and neural models.	[24, 27, 31, 39, 81, 84, 146]

–*Ordered vs. unordered items.* The real-world session data usually falls into two types – ordered data and unordered ones, according to whether there is order over the items in a session or not [141]. For instance, in the sequence data of medical treatment or in the gene expression data, the order of treatment activities or genes are quite strict and cannot be destroyed. While in the grocery shopping-basket data, the order over items within a shopping basket may not make much sense since many customers may just randomly pick up and put items into the basket. In addition, usually the original order over which the items are put into the cart cannot be kept well in the transaction recording data in real-world cases.

To this end, some approaches working on the item-level dependency for session-based recommendations assume a strict order over items within a session, and they mainly rely on the sequential dependency over items. A naive approach is the sequential pattern mining-based RS, which explicitly mine frequent sequential patterns to guide the recommendations. The details will be given in Sections 5.2 and 6.2. To capture more implicit sequential dependency over items within sessions and keep the original information as much as possible, the Markov Chain-based approaches and the neural model-based approaches, especially the RNN-based approaches, are proposed. The Markov Chain models and RNN have the natural advantages to capture the comprehensive sequential dependency due to their particular model designs for the time series data, more details will be given in Sections 5.2 and 7.

Other approaches try to relax or avoid the ordering assumption to make them more flexible and practical. In such a case, these approaches do not have a ordering assumption over items. The pattern/rule-based approaches are a simple and straightforward one. They explicitly discover the frequent patterns based on item co-occurrences to guide recommendations and will be detailed in Sections 5.2 and 6.1. To keep both frequent and non-frequent items in the datasets and to capture more comprehensive implicit dependency over items, more advanced approaches are employed. They include the factorization machine-based approaches, the shallow neural networks (also called as embedding)-based ones, the naive deep neural network (DNN)-based ones, and the convolutional neural network (CNN)-based ones. They will be detailed in Sections 5.2 and 7.

–*First-order dependency vs. higher-order dependency.* Some dependency in the session data is first-order while others may be higher-order [57]. Accordingly, some approaches are built on the first-order dependency assumption and only capture the first-order dependency over items; in other words, they predict the next item(s) by only considering the immediately prior item. Typical approaches belonging to this category include the first-order Markov chain-based approaches and the factorization-based approaches. Details on these approaches will be given in Sections 7.1 and 7.2. However, most session data not only involves the first-order dependency but also the higher-order dependency. In this case, more advanced approaches such as neural model-based approaches are necessary to capture higher-order dependency for more sound recommendations. The neural model-based approaches, including both shallow and deep neural networks, can comprehensively capture both first-order and higher-order dependencies by utilizing the complex network structures. Details on such kind of approaches are given in Section 7.3.

(b) *Session-level Dependency Modelling.* The session-level dependency modeling refers to the modelling of inter-session dependency for session-based recommendations [113]. Since a session is built on items, the session-level dependency modeling is often accompanied by the item-level dependency modeling. From the learning task perspective, the session-level dependency modelling approaches are divided into next-item(s) recommendations and next-basket recommendations. In next-item(s) recommendations, the session-level dependency modelling is employed to incorporate the inter-session dependency and thus take the effect of previous sessions into account [111, 150]. As a result, this can enhance the prior information for next-item(s) recommendations compared to those ones only modelling the intra-session dependency. In next-basket recommendations, the session-level dependency modelling is necessary to capture the inter-session dependency to recommend what would be bought in the next basket by finding the connections between the candidate items and the items in recent baskets [151].

According to how to model the inter-session dependency, the session-level dependency modelling approaches can be mainly divided into two classes: item dependency modelling and Collective dependency modelling.

–*Item dependency modelling.* The item dependency modelling models the item-to-item transitions from one basket to another separately. In other words, it models the inter-item dependency where the items are from different sessions. This kind of approaches first explore the transitions of items between sessions, and then predict the items that may occur in the next basket based on those items that have been put in the current baskets. Factorization machine has been employed to generate next basket recommendations by modelling the item-to-item dependency [112].

–*Collective dependency modelling.* Different from the above mentioned item dependency modelling, the collective dependency modelling models the inter-session dependency by taking each session as a whole. A representation of the whole session can be learned by a hierarchical neural model which first learns the item representation and then integrates the representation of each item in the session. During the training of the model, the next session is usually used as the output to guide the representation learning of recent sessions, therefore the learned representations can actually reflect the dependency between historical sessions and next one [111, 132, 138, 151].

(c) *Feature-level Dependency Modelling.* The feature-level dependency modeling refers to the approaches of modelling the inter-feature dependency [134, 135] and the feature-to-item dependency [140] for session-based recommendations. The inter-feature dependency means one feature of an item may somehow influence its another features, for instance, apples produced in different countries usually have different prices. The feature-to-item dependency indicates the occurrences of items in a session are usually affected by their features, for example, the items belonging to the same category (e.g., food) but with complementary functions (e.g., eating and drinking) are often bought together and thus occur in the same session (i.e., transaction). The involvement of item features enables us to understand the item intrinsic characteristics from a low level perspective. As a result, the feature-level dependency modeling can often greatly benefit the recommendations of cold-start items which have quite limited observed data on the item level and the session level [68].

Much progress has been made on modelling the feature-level dependency for cold-start item recommendations in conventional RS including content-based ones [13, 131] and collaborative filtering ones [42, 158]. However, the feature-level dependency modelling has been less studied in SBRS and is at its early research stage. Only few researchers start to address the cold start issues by incorporating the content features into a neural model or a factorization model for next-item [140] or next-song [31] recommendations.

Table 4. Different Approaches to SBRS

Approaches	Research progress	Gaps
Item-level dependency modeling	Much progress has been made to generally model explicit inter-item dependency within sessions with various approaches including pattern-based approaches [40, 83, 94] and factorization machines [82] to particularly capture implicit inter-item relations with approaches like implicit rule learning [139], to particularly model the sequential inter-item dependency with approaches including sequential pattern-based models [149], Markov chain models [36, 117], RNN models [14, 33, 50, 51, 63, 84, 111, 120, 124], to particularly address the item heterogeneity issue with the attention mechanism [78, 87, 141].	Item imbalance issue (e.g., how to handle tailed items), noise and irrelevant items
Session-level dependency modelling	Much progress has been made in areas such as modelling general inter-session transitions with factorization machines [112], neural representation learning methods [132, 138], modelling sequential inter-session dependency with RNN [87, 111, 113, 150, 151].	Long-term dependency, context-aware inter-session dependency
Feature-level dependency modelling	Some progress has been achieved by incorporating item features into a factorization-based model [31], or a neural model (e.g., RNN or CNN) [53, 127, 140] to model feature-to-item interactions for cold-start item recommendations.	Feature heterogeneity and dependency
Feature value-level dependency modelling	No work focuses on this for session-based recommendations so far.	Feature value heterogeneity and dependency, value-to-feature-to-item interactions
Domain-level dependency modelling	No work focuses on this for session-based recommendations so far.	Domain complementarity and interactions

(d) *Feature Value-level Dependency Modelling & Domain-Level Dependency Modelling.* The feature value-level dependency modeling captures the intra-feature dependency [133] and the feature value-to-feature-to-item interactions for session-based recommendations. The domain-level dependency modeling captures the dependency between different domains to help with the session-based recommendations in one domain or in both domains, which will be discussed in Section 8.6.

To the best of our knowledge, there is no work exploring the feature value-level dependency or domain-level dependency in SBRS. Some work has been done to explore the feature value-level dependency for other tasks including object representation learning [133] and outlier detection [103, 105]. The significance of domain-level dependency has been demonstrated by the great success of transfer learning [100]. Some work has focused on modelling the domain-level dependency with the aim of borrowing some prior information from source domains to help with the recommendations in the target domain [126]. However, most of such work falls into cross-domain collaborative filtering, which builds a bridge between the source domain and the target domain to address the data sparsity issue in collaborative filtering RS [55, 101]. In SBRS, there is no existing work focusing on domain-level dependency modeling to the best of our knowledge. However, based on our observations, this would be a very promising direction in the future, as will be discussed in Section 8.

5.2 A Categorization from the Technical Perspective

In this section, we try to categorize SBRS from the technical perspective. Specifically, all existing works are divided into two branches: model-free approaches and model-based approaches. Each branch contains several types of approaches.

5.2.1 Model-Free Approaches. Model-free approaches are mainly built on data mining techniques and they usually do not involve complex mathematical models. Two typical approaches in this branch are pattern/rule-based RS for unordered session data and sequential pattern-based RS for ordinal session data.

(a) *Pattern/Rule-based Approaches.* Pattern/rule-based RS first mine frequent patterns or association rules and then use these patterns and rules to guide the subsequent recommendations. This is based on the assumption that most customers would follow the common shopping patterns [1, 83, 94]. For instance, customers usually bought milk and bread together when they go shopping, therefore $\{milk, bread\}$ can be treated as a frequent pattern to recommend bread to those why have bought milk. It should be noted that pattern/rule-based RS are applied in unordered data.

(b) *Sequential Pattern-based Approaches.* To handle those data having a strict order over items or involving time-factor based effect, sequential pattern-based RS are proposed. Similar to pattern-based RS, they first mine a collection of sequential patterns and then recommend the remaining items after the occurrence of the prior items [95, 97, 149].

5.2.2 Model-based Approaches. Different from the model-free RS, model-based RS are usually built on strict assumptions like ordering over items and complex models such as Markov chain models. Existing model-based approaches can be mainly categorized into three classes according to the models they involved: Markov Chain-based approaches, factorization-based approaches, and neural model-based approaches.

(a) *Markov Chain-based Approaches.* Markov Chain-based RS models the first-order (sometimes higher order) dependency over a sequence of items by using transitional probabilities and then generates the recommendations of the following items by utilizing such dependency [36, 112, 117]. Different from sequential pattern-based approaches which are easy to filter out those infrequent items and patterns and thus lead to information loss, Markov Chain-based RS take all items into consideration and thus decrease the information loss greatly.

(b) *Factorization-based Approaches.* These approaches first factorize the item co-occurrence matrix or the item-to-item transitional matrix into a latent representation vector of each item and then predict the following items by using these latent representations [82, 112]. Such approaches should be distinguished from the commonly used factorization machine (like matrix factorization) in collaborative filtering-based RS, which usually factorize the user-item interaction matrix (like rating matrix) into latent factors of users and items respectively [85, 123].

(c) *Neural Model-based Approaches.* Neural model-based approaches take advantage of the neural network to learn the complex relationships and interactions over items within or between sessions and then generate recommendations based on such interactions. Based on the model structure, neural model-based approaches can be divided into shallow neural model-based approaches, which sometimes are also called embedding models or representation learning models [45, 57, 72, 141] and deep neural model-based approaches like RNN [51, 122].

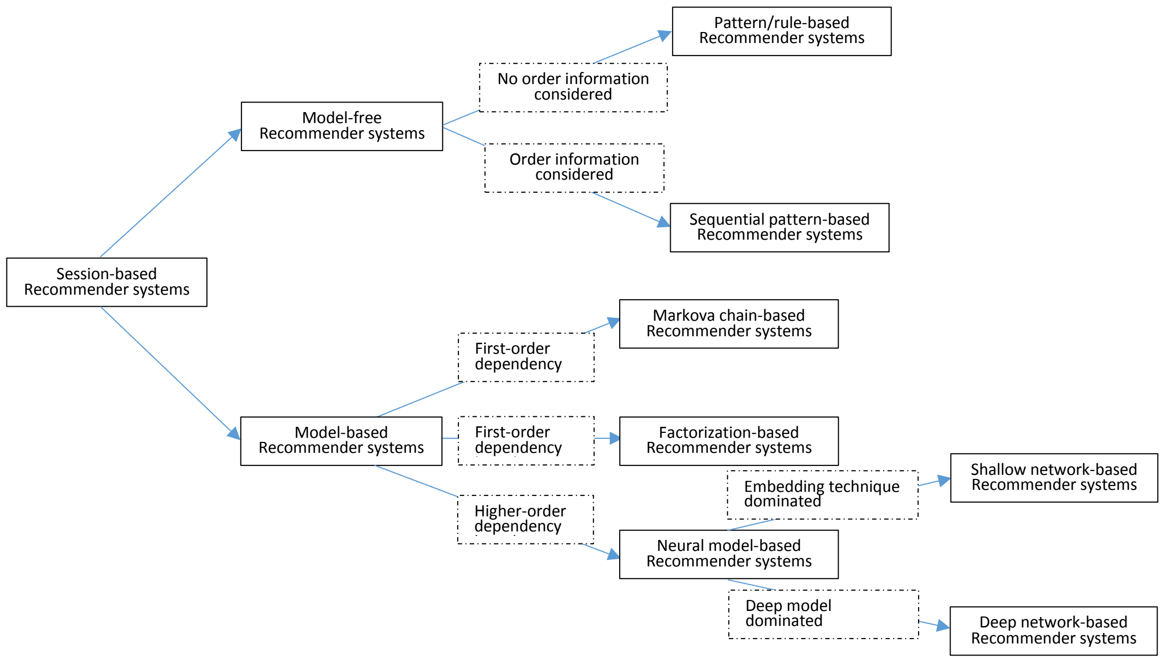

Fig. 5. The categorization of SBRS

5.2.3 Comparisons between Different Technical Approaches. After having a brief understanding of each technical approach for session-based recommendations, we present a comprehensive comparison between different approaches in this section.

Generally speaking, model-free approaches are simple, straightforward and easy to implement. Since both pattern/rule-based and sequential pattern-based approaches are frequency-based only, they are easy to filter out infrequent but significant items or patterns. As a result, model-free approaches are suitable for the relation discovery between frequent items in simple datasets. However, they may easily fail to model complex dependency in complex datasets for session-based recommendations. In contrast, model-based approaches are much more complicated but powerful in complex datasets. On one hand, they do not explicitly filter out items or patterns from the datasets, but retain the information to the maximum extent. On the other hand, thanks to the complex working mechanisms such as deep neural networks [142], these model-based approaches are able to model the complex and implicit relations hidden in the data, which results in more solid session-based recommendations. More detailed comparisons between different specific approaches are provided in Table 5.

Table 5. Comparisons between Different Technical Approaches for Session-based Recommendations

Approaches	Working mechanism	Applicable scenarios	Target issues	Pros	Cons
Pattern/rule-based approaches	Mine frequent patterns or association rules to guide recommendations	Simple, balanced and dense session data without order	Capture explicit co-occurrence-based dependency between items	Intuitive, simple and effective on simple data	Information loss, cannot handle complex data (e.g., imbalanced data, long tailed data)
Sequential pattern-based approaches	Mine sequential patterns to guide recommendations	Simple, balanced and dense session data with order	Capture explicit co-occurrence-based inter-item sequential dependency	Intuitive, simple and effective to capture sequential relations on simple data	Information loss, cannot handle complex data (e.g., imbalanced data, long tailed data)
Markov chain-based approaches	Use Markov chain to model the transitions between items or sessions for recommendations	Relative simple sequential data mainly with short-term and low-order dependency	Capture explicit or implicit inter-item sequential dependency	Reduced information loss, flexible, good at modelling short-term sequential dependency	Usually ignore long-term and higher-order dependency
Factorization-based approaches	Factorize item transitions into latent representations of items for recommendations	Relative simple data mainly with short-term and low-order dependency	Learn latent item representations to fit for item-to-item transitions	Reduced information loss, good at modelling low-order dependency	Easy to suffer data sparsity issues, cannot capture higher-order dependency
Neural model-based approaches	Model the complex dependency in a neural network and embed this dependency into the learned latent representations for recommendations	Complex data with sequential or non-sequential dependency	Encode the complex dependency into latent representations of items or sessions	Powerful, can capture both long term and short term, higher-order and low-order dependency	Hard to implement, computationally costly

6 MODEL-FREE APPROACHES

Model-free SBRS are mainly based on data mining especially pattern mining techniques. The general idea is to find out the common and explicit regularities by mining the patterns from session data and then generate recommendations by using these regularities. Two branches of typical approaches are pattern/rule-based RS and sequential pattern-based RS.

6.1 Pattern/Rule-based Approaches

Pattern/rule-based RS mainly contain three stages: frequent pattern mining, session matching, and item recommendations. To be specific, given a item universal set I and the corresponding session set S over I as defined in Definition 2.2, a set of frequent patterns $PT = \{p_1, p_2, ..., p_{|PT|}\}$ are mined by using pattern mining algorithms like Apriori [7] or FP-Tree [46]. For a given active partial session $\hat{s}$ (e.g., a collection of chosen items in one transaction), if an item $\hat{i}$ exists so that $\hat{s} \cup \hat{i}(\hat{i} \in I \setminus \hat{s})$ is a frequent pattern, namely $\{\hat{s} \cup \hat{i}\} \in PT$, then item $\hat{i}$ is a candidate for recommendations. Further, if the conditional probability $P(\hat{i}|\hat{s})$ is greater than a predifined *confidence* threshold β, then $\hat{i}$ is added into the recommendation list [94].

Besides the aforementioned basic pattern/rule-based RS framework, there are many variants. Lin etc. [83] incorporated association rule mining into collaborative filtering. They replaced the minimum support constraint required in pattern mining with a constraint on the number of rules for each end user to improve the efficiency by avoiding mining too many rules irrelevant to a specific end user. To consider the different significance of different items and thus to recommend more useful ones, the methods in [148] and [40] 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 RS. By mining user behavior patterns, e.g., web navigation patterns, association rule mining is applied to capture the preference of a specific user or a group of users and thus help generate personalized recommendations [2, 40, 74, 152]. Some other works combined pattern mining into collaborative filtering to address some particular issues like sparsity, robustness and personalizition [59, 74, 114]. With regard to the application, except for the traditional shopping basket-based recommendations, pattern-based RS are commonly applied in web recommendations [96], music recommendations [118], and so on.

6.2 Sequential Pattern-based Approaches

Although similar to pattern-based RS, sequential pattern-based RS is different from the former mainly from two aspects: (1). It mainly captures the inter-session dependencies for cross session recommendations (e.g., next-basket recommendations), which is different from pattern-based RS capturing intra-session dependencies for inner session recommendations (e.g., next item recommendations); (2) It takes the order over sessions into account and thus it is more suitable for sequrntial data. Similar to pattern/rule-based RS, sequential pattern-based RS also contain three stages: sequential pattern mining, sequence matching, and recommendation generation. Next, we give a formulation of sequential pattern-based RS.

Given a sequence set $Q = \{q_1, q_2, ..., q_{|Q|}\}$, in which a sequence $q = \{s_1, s_2, ..., s_{|q|}\}(q \in Q)$ is a set of sessions order by their unique timestamps and a user u's past sequence q_u, a sequential pattern-based RS generates recommendations by matching q_u to those sequential patterns mined on Q [149]. Specifically, let $q_u = \{s_1, s_2, ..., s_h\}$ be a sequence of u and $SP = \{p_1, p_2, ..., p_{|SP|}\}$ be the set of sequential patterns mined on Q. For any pattern $p \in SP$, if the last session of q_u belongs to p, i.e., $p = \{s_1, s_2, ..., s_h, s_l ...\}$, then pattern p is a relevant pattern for this specific recommendation and the items after s_h in p, like items in s_l, are candidate items. For each candidate item i_c, its support is the sum of the support of all relevant patterns, formally:

$$supp(i_c) = \sum_{s_h \in q_u, s_h \in p, i_c \in s_l, s_l \in p, p \in SP} supp(p) \tag{1}$$

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

Except for the basic framework described above, there are various extensions to make a more reliable sequential pattern-based RS. A typical example is to utilize the user-related weighted sequential pattern mining for personalized recommendations. Each sequence is assigned a weight based on its similarity to those past sequences of the target user to personalize recommendations for specific users [121, 149, 153]. Another extension is to build a hybrid RS by combining sequential pattern mining and collaborative filtering [30, 58, 86, 153]. Thanks to the combination, both the dynamic individual patterns captured by the sequential pattern mining and the general preference modeled by the collaborative filtering are considered. From the application perspective, item recommendations in transactional data and web page recommendations in web access log data are two typical application domains of sequential pattern-based RS [97, 157].

7 MODEL-BASED APPROACHES

Different from model-free approaches, model-based ones are usually based on special models (e.g., Markov chain model). In this section, three representative model-based RS are discussed, i.e., Markov chain-based ones, factorization machine-based ones, and neural model-based ones.

7.1 Markov Chain-based Approaches

Markov chain-based RS adopt Markov chains to model the transitions over items within sessions to predict the probable next item given a sequence of prior items in a session. To decrease the model complexity, most RS are built on first-order Markov chain, like [15, 39, 117, 146]. Next, we give the formulation of a basic Markov chain-based RS, followed by some extensions.

7.1.1 Basic Markov Chain-based Approaches. Generally speaking, the process of a basic Markov chain-based RS is simple: first calculating the transition probability over a sequence of items from the training data, and then matching a user's shopping sequence to the sequence with calculated transition probability for prediction and recommendations. Those candidate items with high probability are put into the recommendation list [36].

Given a session set S as defined in Definition 2.2, each session $s \in S$ is a sequence of items with strict order. A Markov chain synopsis is built to encode all sessions into a directed graph G by taking each item as a node and the co-occurrence between each pair of items as an edge. The frequency of each item and the co-occurrence times of each item pair correspond to the node weight and edge weight respectively. Therefore, each session is described by a path in G.

The Markov chain model is defined as a set of tuples $\{ST, \boldsymbol{P_t}, P_0\}$, where ST is the state space including all distinct nodes in G, $\boldsymbol{P_t}$ is the $m * m$ one-step transition probability matrix between m distinct items, and P_0 is the initial probability of each state in ST. The first-order transitional probability from item i_j to i_k is defined as:

$$\boldsymbol{P_t}(j,k) = P(i_j \rightarrow i_k) = \frac{freq(i_j \rightarrow i_k)}{\sum_{i_t \in I} freq(i_j \rightarrow i_t)} \tag{2}$$

Then we can estimate the probability of a shopping path $\{i_1 \rightarrow i_2 \rightarrow i_3\}$ by using the aforementioned first-order Markov chain model:

$$P(i_1 \rightarrow i_2 \rightarrow i_3) = P(i_1) * P(i_2|i_1) * P(i_3|i_2) \tag{3}$$

Given a sequence of chosen items, we chose the shopping paths with high probabilities and take the given sequence as the prefix. Those items occurred in these paths and after the given sequence are put into the recommendation list.

Except for the basic Markov chain-based RS defined above, there are many variants like [47]. For example, Zhang etc. combined first- and second-order Markov model together to make more accurate web recommendations [154], while Le etc. developed a hidden Markov model-based probabilistic model for next item recommendations by incorporating additional factors like context features to leverage the recommendation accuracy [73]. Another important variant is to adopt a factorization method on the transition probability to estimate those unobserved transitions [112], which will be introduced in Section 7.2.

7.1.2 Latent Markov Embedding-based Approaches. Different from the basic Markov chain-based RS which calculate the transition probability based on the explicit observations directly, latent Markov Embedding (LME)-based RS first embed the Markov chains into a Euclidean space and then calculate the transition probabilities between items based on their Euclidean distance [26]. In this way, it can drive the unobserved transitions and thus solve the sparsity issue in limited observed data. Formally, each item i is represented as a vector $\boldsymbol{v}_i$ in a d-dimensional Euclidean space, and the transition probability $P(i_{(j-1)} \rightarrow i_j)$ is assumed to be negatively related to the Euclidean distance $||\boldsymbol{v}_{i_j} - \boldsymbol{v}_{i_{j-1}}||_2$ between items i_j and i_{j-1}. Accordingly, the probability of an shopping path $q' = \{i_1 \rightarrow i_2 \rightarrow, ..., \rightarrow i_l\}$ can be defined based on a Markov model:

$$P(q') = \prod_{j=2}^{l} P(i_{j-1} \rightarrow i_j) = \prod_{j=2}^{l} \frac{e^{-||\boldsymbol{v}_{i_j} - \boldsymbol{v}_{i_{j-1}}||_2^2}}{\sum_{i_k \in I} e^{-||\boldsymbol{v}_{i_k} - \boldsymbol{v}_{i_{j-1}}||_2^2}} \tag{4}$$

To generate personalized recommendations, Wu etc. [146] proposed a personalized Markov embedding (PME) model which maps both users and items into an Euclidean space where the user-item distance and item-item distance reflects the corresponding pairwise relationship respectively.

Personalized ranking metric embedding is proposed in [39] to learn the embeddings by fitting the ranking of the POI transitions to make use of the unobserved data. Specifically, the observed next POI is assumed to be more related to the current one than those unobserved ones. For instance, if transition $i_j \rightarrow i_k$ is observed while $i_j \rightarrow i_t$ is not, i_k should be ranked higher than i_t in terms of i_j, this is modeled as a ranking ">" over POIs. Therefore, the whole model is transferred into:

$$P(i_j \rightarrow i_k) > P(i_j \rightarrow i_t) \Rightarrow ||\boldsymbol{v}_{i_j} - \boldsymbol{v}_{i_k}||_2 - ||\boldsymbol{v}_{i_j} - \boldsymbol{v}_{i_t}||_2 > 0 \tag{5}$$

7.2 Factorization Machine-based Approaches

Recently, factorization machines are adopted to factorize the observed transitions from the current item to the next one into latent representations of items or users. The resultant latent representations are used to estimate the unobserved transitions for recommendations [54, 81, 117]. Next, we first formalize the basic factorization machine-based SBRS and then review some variants.

Once the personalized transition probability is achieved from the observed data, a transition matrix $\boldsymbol{A}^u$ is built for each user u. Therefore, for all users a transition tensor $\mathcal{A}$ is built like $\mathcal{A}^{|U|\times|I|\times|I|}$, where each entry a_{u,i_j,i_k} indicates the transition probability from item i_j to i_k under user u. A general linear factorization model, the Tucker Decomposition, is used to factorize the transition cube and estimate the unobserved transitions.

$$\hat{\mathcal{A}} = C \times \boldsymbol{V}_U \times \boldsymbol{V}_{I_j} \times \boldsymbol{V}_{I_k} \tag{6}$$

where C is a core tensor, $\boldsymbol{V}_U$ is the feature matrix for users while $\boldsymbol{V}_{I_j}$ and $\boldsymbol{V}_{I_k}$ are the feature matrix for last items and the following items respectively.

The Tucker Decomposition subsumes factorization models like the Canonical Decomposition [10] aka parallel factor analysis (PARAFAC). A special case of Canonical Decomposition is used to model the pairwise interactions because of the very spares transitions observed for $\mathcal{A}$:

$$\hat{a}_{u,i_j,i_k} =< \boldsymbol{v}_u, \boldsymbol{v}_{i_j} > + < \boldsymbol{v}_{i_j}, \boldsymbol{v}_{i_k} > + < \boldsymbol{v}_u, \boldsymbol{v}_{i_k} > \tag{7}$$

where $\boldsymbol{v}_u$, $\boldsymbol{v}_{i_j}$ and $\boldsymbol{v}_{i_k}$ is the latent factor of user u, the current item i_j and the next item i_k respectively. Subsequently, a factorized personalized Markov chain (FPMC) model [112] is built for SBRS.

Cheng etc. [27] extended FPMC into FPMC-LR model by adding a constraint to limit user movements into a localized region to be more consistent with the real-world tourism cases for next POI recommendations.

The factorization models only built on item co-occurrence transition matrix actually only captures the sequential patterns over items while ignoring the user's individual preference which is captured by other RS like collaborative filtering. In [82], the authors combines the matrix factorization used in collaborative filtering and the matrix factorization used in SBRS together to capture both the individual preference and the item transition patterns. Specifically, the authors proposed a co-factorization model, CoFactor, which jointly decomposes the user-item interaction matrix and the item-item co-occurrence matrix with shared latent factors for items. Other similar works include [88] which utilizes the matrix factorization model to learn the preference transitions from one location category to another and thus provides location recommendations.

7.3 Neural Model-based Approaches

Derived by the powerful generalization and representation capability of neural networks, a series of neural model-based approaches have been developed to model the comprehensive relations between item features, items and sessions for recommendations. Generally, neural model-based SBRS can be divided into shallow neural models and deep neural models according to the number of layers incorporated into the neural networks. Next, we introduce these two kinds of recommendation models respectively.

7.3.1 Shallow Neural Models. Shallow neural models, sometimes called embedding models, usually contain a shallow network structure [44] which maps the items within sessions into a latent space and then capture inter-item relations more easily in this latent space. The mechanism behind is that, when mapping items into a multi-dimensional latent space, their positions in such a space reflect their relations. For instance, the items within a shorter distance have higher similarity and are more relevant to each other. To this end, the latent numerical vector representation of each item contain much richer information than the original item ID. This is mainly inspired by the great success of the recently proposed word embedding techniques including Skip-gram [109] and the CBOW model [92] in natural language processing [91, 93].

A representative shallow neural model-based SBRS is [57]. It designed a shallow network with a wide-in-wide-out structure to firstly map user IDs and the corresponding item IDs into latent vector representations and then combine them together as the given context representation, which is lastly fed into the output layer (a softmax layer) to predict the corresponding next item.

To be specific, the shallow network embeds a user u and an item i into a latent vector using the logistic function $\delta(\cdot)$ for nonlinear transformation:

$$\boldsymbol{v}_u = \delta(\boldsymbol{W}^1_{:,u}) \tag{8}$$

$$\boldsymbol{v}_i = \delta(\boldsymbol{W}^2_{:,i}) \tag{9}$$

where $\boldsymbol{W}^1 \in \mathbb{R}^{K\times|U|}$ and $\boldsymbol{W}^2 \in \mathbb{R}^{L\times|I|}$ are the weight matrix to connect user input and session context input to the hidden layer respectively. In this way, each user and each item are represented by a numerical vector $\boldsymbol{v}_u \in [0,1]^K$ and $\boldsymbol{v}_i \in [0,1]^L$ respectively, similar to the vector representation of each word in the Word2Vec model [43].

In addition, Wang etc. incorporated item features into the network to embed an item ID and its features simultaneously to tackle the cold-start recommendation issues [140]. The work in [130, 137] also incorporated side information to help with building a more informative item embedding. To attentively learn the really relevance scale of different contextual items to the target item, the attention mechanism is adapted into the embedding process of items [141]. The work in [138] and [132] learned a hierarchical representation for the next basket recommendations. Furthermore, there are more similar SBRS built on item embedding inspired by the Word2Vec model like [11, 45, 72, 80, 98, 136, 156].

7.3.2 Deep Neural Models. It is generally believed that the deep neural models for SBRS started in 2016 when a gated recurrent unit (GRU)-based RNN model (GRU4Rec) was designed for SBRS in [51]. Following this work, a series of deep neural models have been proposed for SBRS, like [50, 53, 111, 124]. Out of these, RNN-based models dominated this area due to its intrinsic advantages for modeling sequential dependency, since a majority of session data is assumed to be sequentially dependent. DNN-based models are usually applied to learn an optimized combination of different representations for the following recommendations. CNN-based models are often adopted to extract more informative local features from sessions and other contents for better recommendations. Next, we introduce the representative models for SBRS under different network architectures respectively.

(a) *RNN-based Models.* We first introduce the basic GRU4Rec model [51], then present some improved versions and lastly show some variants of RNN-based SBRS.

Given a session set S as defined in 2.2, where each session $s \in S$ corresponds to a sequence of items bought consecutively in a transaction. GRU4Rec models each session as a sequence, in which it tries to predict a probability distribution over the next elements of the sequence conditioned on the current hidden state. A GRU is an advanced RNN unit aiming at handling gradient vanishing problems by learning when and by how much to update the hidden state [29]. Specifically, its hidden state $\boldsymbol{h}_t$ is updated as follows:

$$\boldsymbol{h}_t = (1 - z_t)\boldsymbol{h}_{t-1} + z_t\hat{\boldsymbol{h}}_t \tag{10}$$

where the update gate z_t and the candidate hidden state $\hat{\boldsymbol{h}}_t$ are computed by the following equations respectively. z_t actually decides how much the unit updates its hidden state from the last state.

$$\boldsymbol{z}_t = \sigma(\boldsymbol{W}_z\boldsymbol{x}_t + \boldsymbol{U}_z\boldsymbol{h}_{t-1}) \tag{11}$$

$$\hat{\boldsymbol{h}}_t = tanh(\boldsymbol{W}\boldsymbol{x}_t + \boldsymbol{U}(\boldsymbol{r}_t \odot \boldsymbol{h}_{t-1})) \tag{12}$$

where σ is the logistic sigmoid function and the reset gate $\boldsymbol{r}_t$ is given by:

$$\boldsymbol{r}_t = \sigma(\boldsymbol{W}_r\boldsymbol{x}_t + \boldsymbol{U}_r\boldsymbol{h}_{t-1}) \tag{13}$$

Each GRU unit represents one hidden state and a GRU layer consisting of a sequence of connected GRU units forms a hidden layer of a RNN. In this case, a l-length sequence can be modeled by a hidden layer with l GRU units. The input $\boldsymbol{x}_t$ is the embedding of item i_t in a session s. Therefore, given the past items in s, GRU4Rec is built to model the intra-session dependency over sequential items and thus predict the probability distribution on the next item.

To improve GRU4Rec, Tan etc. [124] adopted data augmentation via sequence preprocessing and embedding dropout to enhance training and reduce overfitting respectively. In addition, a model pre-training approach is proposed to consider the possible temporal shift in the distribution of input data. Quadrana etc. [111] further improved GRU4Rec by proposing a hierarchical RNN model to incorporate cross-session information for recommendations. Specifically, a two-level GRU-based RNN is designed: the session-level GRU models the sequence of items purchased by a user within a 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. To this end, both the intra- and inter-session dependencies are captured to generate more reliable next item(s) recommendations. Another quite similar work is Inter-Intra RNN (II-RNN) proposed in [113]. In [35], the authors designed a unique user-based GRU model which incorporates user characteristic to generalize personalized next item recommendations.

Except for the GRU-like RNN-based RS mentioned above, there are many other variants for RNN-based SBRS. For example, the Dynamic REcurrent bAsket Model (DREAM) [151] learns a dynamic representation of a user by utilizing the hidden state $\boldsymbol{h}_t^u$ of user u at time step t as the representation of u at time t while the input for each state is the embedding of each session aggregated from the embeddings of all the items contained in it. In this case, the user representation is different from each time point, as new sessions are added into the model along with the time.

Other extensions include the incorporation of (1) variational inference into RNN to handle the uncertainty in sparse transaction data and simultaneously leverage the model's scalability on large real-world datasets [25, 32]; (2) other side information like item features and other contextual factors like time, location and interfaces into RNN-based models to enhance the recommendations [12, 53]; (3) time decay or attention mechanism into RNN-based models to discriminate the intra-session dependency over items and thus achieve more precise recommendations [14, 108]; and (4) traditional models like factorization machine or neighbourhood models to make up the drawbacks of RNN-only models [63, 128]. Other similar RNN-based models include [50, 67, 145].

(b) *DNN-based Models.* Except for RNN, DNN is an alternative solution for SBRS, especially when there is no strict order over items within a session. Here, DNN particularly refers to the naive deep neural network structure consisting of multi-layer perceptron (MLP). Next, we introduce a typical DNN-based SBRS and then discuss some variants.

In [144], a DNN is applied to learn a comprehensive session representation for recommendations by learning a optimized combination of the embeddings of different components under a session context with a MLP layer as below:

$$\boldsymbol{h}_1 = \sigma(\boldsymbol{W}_c \boldsymbol{v}_{s_c} + \boldsymbol{W}_v \boldsymbol{v}_{s_v} + \boldsymbol{W}_t \boldsymbol{v}_{i_t} + \boldsymbol{W}_u \boldsymbol{v}_u) \tag{14}$$

where $\boldsymbol{v}_{s_c}$, $\boldsymbol{v}_{s_v}$ are the embeddings of two sub sessions (i.e., click sub session and view sub session) which both serve as the context for target item prediction, while $\boldsymbol{v}_{i_t}$ and $\boldsymbol{v}_u$ are the embeddings of the target item and the corresponding user respectively. $\boldsymbol{W}_c$, $\boldsymbol{W}_v$, $\boldsymbol{W}_t$ and $\boldsymbol{W}_u$ are the corresponding weight matrices to fully connect each of the embeddings to the first hidden layer of DNN.

Similar to the above approach, Jannach etc. applied DNN to learn an optimized combination of different factors like "reminders", "item popularity" and "discount" as a compound session-based features for the next-item prediction [64]. Another example is the implementation of DNN to transfer and generalize the sparse user-item interactions to dense and informative session features for prediction. A wide liner models combined with a deep neural network is applied on Google Play to improve the mobile app recommendation performance [28]. In [122], a DNN-based architecture was proposed to model both the long-term static and short-term temporal user preferences simultaneously for better recommendations.

(c) *CNN-based Models.* CNN is another good choice for SBRS for two reasons: (1) It relaxes the rigid order assumption over items within a session commonly used by RNN, which makes the model more robust; (2) It has high capacity in learning local features from a certain area and relationships between different areas, which can effectively capture the union-level collective dependency that are usually ignored by other models. Next, we specifically introduce a representative CNN-based SBRS [125] and then demonstrate some other similar models.

The model proposed in [125] mainly contains an embedding layer, convolutional layers and fully-connected layers. Particularly, the convolutional layers include a horizontal and a vertical convolutional layer respectively. Specifically, a filtering and a pooling operations are conducted in the horizontal convolutional layer successively.

In the horizontal convolutional layer, the i_{th} convolution value is achived by sliding one filter $\boldsymbol{F}^k$ from top to bottom on the embedding matrix $\boldsymbol{E}$ to interacts with its horizontal dimensions:

$$c_i^k = \phi_c(\boldsymbol{E}_{i:i+h-1} \odot \boldsymbol{F}^k) \tag{15}$$

where ϕ_c is the activation function for the convolutional layer.

Then the final output $\boldsymbol{o} \in \mathbb{R}^n$ from the n filters is obtained by performing the max pooling on the convolution result $\boldsymbol{c}^k = [c_1^k\ c_2^k, ..., c_{l-h+1}^k]$ to capture the most significant features:

$$\boldsymbol{o} = max\{max(\boldsymbol{c}^1), max(\boldsymbol{c}^2), ..., max(\boldsymbol{c}^n)\} \tag{16}$$

Similarly, in the vertical convolutional layer, the convolution result is achieved by sliding the vertical filters on the embedding matrix $\boldsymbol{E}$ to interact with its columns. Finally, the outputs of the two convolutional filters are concatenated as the input of a fully-connected layer to get higher-level features for final predictions.

Another similar work is a 3D CNN [127] built for SBRS, which jointly models the sequential patterns in session click data and the item characteristics from item content features. Furthermore, in [106], the authors proposed a CNN model to capture the preference of individual users to generate personalized recommendations.

8 PROSPECTS AND FUTURE DIRECTIONS

Base on the above summary of the challenges in session modelling and SBRS, the corresponding research progress, and the various techniques applied in SBRS, here we analyze the gaps that have not been well studied and the potential future directions in SBRS.

Although RS have received intensive attention in recent years, SBRS is a relatively emerging area in RS and is still in its early stage. According to our systematic review and the explorations in this area, we have identified a collection of challenges that SBRS research still face, which may suggest some potential future research directions. We divide these challenges into seven branches w.r.t. the information sources and scenario settings. Next, we discuss each branch by firstly illustrating its significance and then presenting the specific open issues.

8.1 Session-based Recommendations With General User Preference

Significance. SBRS usually ignore the general user preference which can be well captured by other conventional RS like collaborative-filtering. This may lead to unreliable recommendations as users with different shopping preferences and consumption habits may chose different next item(s) even under the same session context. In this case, how to learn the general user preference from session data and then incorporate it into an SBRS model is critical yet challenging.

Open issues. Here, we discuss two major issues w.r.t the general preference learning and its incorporation into SBRS and sketch the possibilities for future research directions.

- How to incorporate a user's explicit preference into an SBRS? In this case, it is assumed that user preference on their purchased items is available in the form of a user-item preference matrix. An intuitive way is to first predict a user's preference on all candidate items by utilizing conventional approaches like matrix factorization (MF) and then use the predicted preference data as a constraint to tune the candidate ranking and selection in an SBRS. For instance, if two candidate items have similar possibilities to be chosen in the next choice according to the session context, the one with higher preference to a particular user can be put to the front of the others in the recommendation list. Another way is to combine the above two factors together in ranking the candidate items. In [155], the authors proposed a generative adversarial network (GAN) framework to leverage the MF and RNN-based hybrid model for movie recommendations, which jointly learns long-term preference and short-term shopping patterns.

- How to incorporate a user's experience into an SBRS without explicit preference data? In the real-word cases, the explicit preference data may not be always available as customers may not rate everything they bought. In this case, the shopping-basket-based transactional data is usually treated as a kind of implicit feedback [48, 49] to indicate user preference. Some existing work [9, 110] has explored how to learn a user's preference from such implicit feedback data in collaborative filtering. In practice, such data like clicks or views of items is often available in session-based recommdation scenarios [116]. Note that such implicit preference data is even the same as session data used in SBRS in some cases. Therefore, how to simultaneously learn the a user's implicit preference and the short-term patterns from an identical data source but avoid information duplication is a challenging issue which has not got effective solutions in SBRS.

8.2 Session-based Recommendations Considering More Contextual Factors

Significance. Recommendation context refers to the practical and specific situation when a user is making choices on items. Accordingly, contextual factors refer to different context-related aspects that may affect a user's choice, such as weather, season, location, time and recent popularity trend. Taking these contextual factors into account may make a huge difference on recommendation performance. The significance of context in RS is also emphasized by other researchers like Gediminas el al. [3, 5], Shi el al. [119] and Pagano el al. [99]. Actually, an SBRS can be seen as a simplified context-aware RS whose context is simplified as a session context [128].

Open issues. Although contextual information has been incorporated into other RS or even context-aware RS have been proposed as a new paradigm [5, 129], context is rarely exploited in SBRS.

- How to incorporate more contextual factors into SBRS? A Contextual RNN for Recommendation (CRNN) is proposed to incorporate contextual factors including the type of user-item interactions, the time gap between different events in a session and the time of a day into an RNN-based SBRS [120]. In [62, 75], the recent popularity trend, user's recently viewed items, and items with discount in shopping mall are taken as contextual factors for session-based recommendations. However, these methods are just a starting point, more explorations are still necessary on how to collect more contextual information and how to develop models to more effectively incorporate such information into SBRS.

8.3 Session-based Recommendations With Noisy and Irrelevant Items

Significance. Currently, sequential SBRS like RNN-based ones and Markov chain-based ones always assume strong dependency over successive items. However, this may not be the case in the real-world transactional data because a user may just randomly pick up some items he/she likes into the cart, as the case shown in Section 7.3.2. These randomly picked items may be irrelevant to both the chosen items and the next choice. If we ignore such cases, the recommendation results may be easily misled by these noisy items.

Open issues. How to make reliable recommendations with noisy sessions? Although some mechanisms like attention [141] and pooling [125] have been applied to SBRS to emphasize those really relevant and important items for the next choice, but the relevant research is still quite limited. More efforts are deserved to develop more robust and resilient models to handle noisy sessions.

8.4 Session-based Recommendations for Multi-Step Recommendations

Significance. Usually, a shopping event contains multiple steps rather than just one. For example, when a user bought a bread, he may buy milk later, followed by purchasing cheese. Given the partial session consisting of a bread, most current SBRS only make one forward-step recommendations, namely they only make prediction on milk rather than predict milk and then cheese continuously. This may reduce the utility of RS greatly.

Open issues. How to generate multi-step recommendations given a partial session as the context? This issue is practical and critical but extremely challenging. Considering the advantage in multi-step modeling, the encoder-decoder framework [79, 90] may be an intuitive choice to address this issue.

8.5 Session-based Recommendations With Cross-session Information

Significance. Actually, a user's choice on the next item may not depend only on items in the same session, but also be influenced by items from other sessions. Suppose a user bought a cellphone, then he may would like to buy a cellphone cover or an accessory in his/her next shopping. To this end, taking cross-session information into account may make it practical in some session-based recommendations.

Open issues. How to incorporate cross-session information into SBRS? Explorations on this topic are quite limited. The work in [111] proposed a hierarchical RNN architecture to model both intra- and inter-session dependencies for session-based recommendations. But it just applied a very intuitive way to incorporate multiple sessions into a RNN structure.

8.6 Session-based Recommendations with Cross-domain Information

Significance. Cross-domain means another domain that is different from but relevant to the target domain in which the recommendations are made [56]. Usually, a user's purchased items involve multiple product categories (domains) rather than one domain to meet her/his compound demand. In addition, the choices on items from different domains may not be independent from each other. For instance, a user may have seen a movie "Titanic" first and enjoyed it greatly and then he may listen to the movie's theme song "My heart will go on" and likes the singer "Céline" very much, as a result, he may listen to other songs such as "I'm Your Angel" by "Céline". In another case, young girls who like the protagonist "Rose" in Titanic are likely to buy the same dress as Rose's. Such examples show that the items from different domains may not only be dependent but can even form a sequential session, such as {Titanic, My heart will go on, Céline, I'm Your Angel} or {Titanic, Rose, Rose's style dress}. The recommendations based on such type of sessions are interesting but quite challenging. On one hand, such recommendations not only cover more aspects of our daily living activities but also provide a solution to the data sparsity issue when only one domain is considered. On the other hand, it is hard to collect a user's consumed products or services from various domains together, and the relations between items from different domains are much more complex than that from one domain.

Open issues. According to whether the products or services from different domains can form a tight session or not, there are two open issues to be further explored.

- How to borrow knowledge from other domains to help with the session-based recommendations in the target domain? When no sessions can be built over products or services from different domains, the way to make use of other domains is a target-auxiliary framework. To be specific, it takes the target domain where the recommendations are made as the main information source while taking other domains as a supplementary one to leverage the recommendation performance. An intuitive choice is transfer learning [38, 100] which transfers knowledge from source domains to help with the tasks in the target one. Although transfer learning has been well applied in other RS like collaborative filtering [76, 89, 101, 102], it is rarely explored in SBRS.

- How to make session-based recommendations from multiple domains? This case happens when a session can be built on elements from different domains, like the aforementioned example {Titanic, My heart will go on, Céline, I'm Your Angel}. Different from the aforementioned target-auxiliary framework, this case treats all elements from different domains equally and each domain can serve as the target domain. It is much more interesting yet challenging than the above case. It can incorporate nearly all of a user's daily activities into a unified RS framework. However, the challenges come from two aspects: session-building and model development. Different from session-based recommendations in one domain, where the natural sessions are usually already there, almost no obvious session on cross-domains are available. Therefore, how to build a reasonable session from multiple heterogeneous data sources is the first challenge. Once the cross-domain sessions are ready, how to develop a model to effectively capture the complex and heterogeneous coupling relationships [17] between elements from different domains is another challenge.

8.7 Session-based Recommendations from a Non-IID Perspective

Significance. Non-IID characteristics refer to the non-independent and identically-distributed fact (i.e., the non-IIDness) embedded in objects and their attributes [16, 19, 23], which are quite obvious in session data and session modelling as illustrated in Section 3. In general, any real-life recommendation is non-IID, and non-IID RS apply to any recommendation data, problems and tasks [19]. In particular, the two main types of non-IIDness [16, 19, 23] are couplings and heterogeneities, which are reflected within and between the elements in each level and between different levels in the hierarchical structure of session data (cf. Fig. 2).

Open issues. According to the two typical types of non-IIDness, two sets of open issues in this direction are summarized below.

- How to model the heterogeneities of users, items, user/item features, feature values and sessions in an SBRS? Here 'heterogeneity' corresponds to certain objects, e.g., users, items or sessions, which are different from each other and thus cannot be treated homogeneously in the modelling stage. It should be noted that heterogeneity here somehow relates to but is not limited to the distributions. Some work has begun to explore the item heterogeneity within a session for more specific recommendations, such as [141, 150]. But the progress is quite limited and more explorations are deserved, especially the heterogeneities of different but relevant sessions.

- How to model the diverse couplings between different levels, e.g., the item-level and session-level, in an SBRS? As demonstrated in Section 3.2, different levels in a session structure are comprehensively interacted with each other. To this end, it is necessary to consider the influence from other levels when generating recommendations on the item level. Most of existing work just focuses on the intra-item-level dependency modelling while ignoring the inter-level dependency. Initial progress has been made on learning user couplings, item couplings, and user-item couplings by coupled matrix factorization [77].

However, little attention has been paid to the above issues in the community. Much progress has been made on non-IID learning for other areas, including similarity and metric learning [65, 133–135], coupling learning for objects, attributes and hierarchical representations [66], heterogeneity learning for objects and attributes [159], feature selection [104], and their applications to behavior analysis, outlier detection, clustering, classification, and multi-view, multimodal, and multi-source data analysis [22, 23]. These will be useful for building a deep understanding of non-IID RS and developing non-IID models for capturing the non-IID characteristics in complex recommendation data, problems, and applications.

9 CONCLUSIONS

In this survey, we have conducted a systematic review of the SBRS area. We have firstly motivated the necessity of SBRS research with a comprehensive comparison between SBRS and other RS, followed by the formations and related definitions of SBRS. Then, we have shown the complexities and challenges of SBRS, followed by a brief overview of research activities in SBRS. The up-to-date research progress and specific techniques have been summarized. Lastly, the prospects of SBRS research have been given.

This review suggests further deeper thinking and research into several directions. First, the mathematical nature of any recommendation problems and applications are non-IID, involving hierarchical couplings and heterogeneities, while existing work either assumes users/items and their attributes as IID. Research on non-IID RS address the nature and some intrinsic complexities and challenges of recommendation. Second, any complex RS are complex systems [18], more efforts should be made to understand the system complexities and explore general and specific data complexities and characteristics [21] associated with recommendation problems in general and with SBRS in particular. Third, in addition to the current specific effort made on the item and session-based relation learning, more effort should be on the way to address many other aspects and layers, including users/items, their attributes, user/item groupings, and domains, and to address the hierarchical aspects from attributes to groupings. Fourth, much practical need may be on next-event, next-action, next-choice etc. related recommendations, to shift the recommendations to actionable recommendations, i.e., to enable decision-making actions based on recommendations. Lastly, many specific issues in SBRS will be further explored, including exploring deeper data-driven modeling and more advanced model-driven recommendation in terms of inventing diverse powerful modeling techniques for specific issues and purposes.

ACKNOWLEDGMENTS

The authors would like to thank Dr. Liang Hu from University of Technology Sydney for his great support and constructive suggestions on this work.

REFERENCES

[1] Fabian Abel, Ig Ibert Bittencourt, Nicola Henze, Daniel Krause, and Julita Vassileva. 2008. A rule-based recommender system for online discussion forums. In *International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*. Springer, 12–21.

[2] Gediminas Adomavicius and Alexander Tuzhilin. 2001. Expert-driven validation of rule-based user models in personalization applications. *Data Mining and Knowledge Discovery* 5, 1-2 (2001), 33–58.

[3] Gediminas Adomavicius and Alexander Tuzhilin. 2005. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE transactions on knowledge and data engineering* 17, 6 (2005), 734–749.

[4] Gediminas Adomavicius and Alexander Tuzhilin. 2011. Context-aware recommender systems. In *Recommender systems handbook*. Springer, 217–253.

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

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

[7] Charu C Aggarwal and Jiawei Han. 2014. *Frequent pattern mining*. Springer.

[8] Ahmad M Ahmad Wasfi. 1998. Collecting user access patterns for building user profiles and collaborative filtering. In *Proceedings of the 4th international conference on Intelligent user interfaces*. ACM, 57–64.

[9] Susan C Anyosa, João Vinagre, and Alípio M Jorge. 2018. Incremental Matrix Co-factorization for Recommender Systems with Implicit Feedback. In *Companion of the The Web Conference 2018 on The Web Conference 2018*. International World Wide Web Conferences Steering Committee, 1413–1418.

[10] Hans-Jürgen Bandelt and Andreas WM Dress. 1992. A canonical decomposition theory for metrics on a finite set. *Advances in mathematics* 92, 1 (1992), 47–105.

[11] Oren Barkan, Yael Brumer, and Noam Koenigstein. 2016. Modelling Session Activity with Neural Embedding.. In *RecSys Posters*.

[12] Alex Beutel, Paul Covington, Sagar Jain, Can Xu, Jia Li, Vince Gatto, and Ed H Chi. 2018. Latent Cross: Making Use of Context in Recurrent Recommender Systems. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*. ACM, 46–54.

[13] Mattia Bianchi, Federico Cesaro, Filippo Ciceri, Mattia Dagrada, Alberto Gasparin, Daniele Grattarola, Ilyas Inajjar, Alberto Maria Metelli, and Leonardo Cella. 2017. Content-Based approaches for Cold-Start Job Recommendations. In *Proceedings of the Recommender Systems Challenge 2017*. ACM, 6.

[14] Veronika Bogina and Tsvi Kuflik. 2017. Incorporating dwell time in session-based recommendations with recurrent Neural networks. In *CEUR Workshop Proceedings*, Vol. 1922. 57–59.

[15] Francesco Bonchi, Raffaele Perego, Fabrizio Silvestri, Hossein Vahabi, and Rossano Venturini. 2011. Recommendations for the long tail by term-query graph. In *Proceedings of the 20th international conference companion on World wide web*. ACM, 15–16.

[16] Longbing Cao. 2014. Non-IIDness Learning in Behavioral and Social Data. *Comput. J.* 57, 9 (2014), 1358–1370.

[17] Longbing Cao. 2015. Coupling learning of complex interactions. *Information Processing & Management* 51, 2 (2015), 167–186.

[18] Longbing Cao. 2015. *Metasynthetic Computing and Engineering of Complex Systems*. Springer.

[19] Longbing Cao. 2016. Non-IID Recommender Systems: A Review and Framework of Recommendation Paradigm Shifting. *Engineering* 2, 2 (2016), 212–224.

[20] Longbing Cao. 2017. Data science: challenges and directions. *Commun. ACM* 60, 8 (2017), 59–68.

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

[22] Longbing Cao, Yuming Ou, and Philip S. Yu. 2012. Coupled Behavior Analysis with Applications. *IEEE Trans. Knowl. Data Eng.* 24, 8 (2012), 1378–1392.

[23] Longbing Cao, Guansong Pang, Liang Hu, Chengzhang Zhu, Trong Dinh Thac Do, and Songlei Jian. [n. d.]. Non-IID Learning. ([n. d.]).

[24] Buru Chang, Yonggyu Park, Donghyeon Park, Seongsoon Kim, and Jaewoo Kang. 2018. Content-Aware Hierarchical Point-of-Interest Embedding Model for Successive POI Recommendation.. In *IJCAI*. 3301–3307.

[25] Sotirios Chatzis, Panayiotis Christodoulou, and Andreas S Andreou. 2017. Recurrent Latent Variable Networks for Session-Based Recommendation. *arXiv preprint arXiv:1706.04026* (2017).

[26] Shuo Chen, Josh L Moore, Douglas Turnbull, and Thorsten Joachims. 2012. Playlist prediction via metric embedding. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 714–722.

[27] Chen Cheng, Haiqin Yang, Michael R Lyu, and Irwin King. 2013. Where You Like to Go Next: Successive Point-of-Interest Recommendation.. In *IJCAI*, Vol. 13. 2605–2611.

[28] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al. 2016. Wide & deep learning for recommender systems. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*. ACM, 7–10.

[29] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259* (2014).

[30] 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. *Electronic Commerce Research and Applications* 11, 4 (2012), 309–317.

[31] Szu-Yu Chou, Yi-Hsuan Yang, Jyh-Shing Roger Jang, and Yu-Ching Lin. 2016. Addressing cold start for next-song recommendation. In *Proceedings of the 10th ACM Conference on Recommender Systems*. ACM, 115–118.

[32] Panayiotis Christodoulou, Sotirios P Chatzis, and Andreas S Andreou. 2017. A Variational Recurrent Neural Network for Session-Based Recommendations using Bayesian Personalized Ranking. (2017).

[33] Robin Devooght and Hugues Bersini. 2017. Long and Short-Term Recommendations with Recurrent Neural Networks. In *Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization*. ACM, 13–21.

[34] Oxford Dictionaries. 2016. English oxford living dictionaries. *URI: https://en. oxforddictionaries. com/definition/lichen. Cited* 10, 11 (2016).

[35] Tim Donkers, Benedikt Loepp, and Jürgen Ziegler. 2017. Sequential User-based Recurrent Neural Network Recommendations. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*. ACM, 152–160.

[36] Magdalini Eirinaki, Michalis Vazirgiannis, and Dimitris Kapogiannis. 2005. Web path recommendations based on page ranking and markov models. In *Proceedings of the 7th annual ACM international workshop on Web information and data management*. ACM, 2–9.

[37] Michael D Ekstrand, John T Riedl, Joseph A Konstan, et al. 2011. Collaborative filtering recommender systems. *Foundations and Trends® in Human–Computer Interaction* 4, 2 (2011), 81–173.

[38] Ali Mamdouh Elkahky, Yang Song, and Xiaodong He. 2015. A multi-view deep learning approach for cross domain user modeling in recommendation systems. In *Proceedings of the 24th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 278–288.

[39] Shanshan Feng, Xutao Li, Yifeng Zeng, Gao Cong, Yeow Meng Chee, and Quan Yuan. 2015. Personalized Ranking Metric Embedding for Next New POI Recommendation.. In *IJCAI*. 2069–2075.

[40] R Forsati, MR Meybodi, and A Ghari Neiat. 2009. Web page personalization based on weighted association rules. In *Electronic Computer Technology, 2009 International Conference on*. IEEE, 130–135.

[41] Guojun Gan, Chaoqun Ma, and Jianhong Wu. 2007. *Data clustering: theory, algorithms, and applications*. Vol. 20. Siam.

[42] Zeno Gantner, Lucas Drumond, Christoph Freudenthaler, Steffen Rendle, and Lars Schmidt-Thieme. 2010. Learning attribute-to-feature mappings for cold-start recommendations. In *Data Mining (ICDM), 2010 IEEE 10th International Conference on*. IEEE, 176–185.

[43] Yoav Goldberg and Omer Levy. 2014. word2vec explained: Deriving mikolov et al.'s negative-sampling word-embedding method. *arXiv preprint arXiv:1402.3722* (2014).

[44] Gregory Goth. 2016. Deep or shallow, NLP is breaking out. *Commun. ACM* 59, 3 (2016), 13–16.

[45] Asnat Greenstein-Messica, Lior Rokach, and Michael Friedman. 2017. Session-based recommendations using item embedding. In *Proceedings of the 22nd International Conference on Intelligent User Interfaces*. ACM, 629–633.

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

[47] Ruining He and Julian McAuley. 2016. Fusing similarity models with markov chains for sparse sequential recommendation. In *Data Mining (ICDM), 2016 IEEE 16th International Conference on*. IEEE, 191–200.

[48] Ruining He and Julian McAuley. 2016. VBPR: Visual Bayesian Personalized Ranking from Implicit Feedback.. In *AAAI*. 144–150.

[49] Xiangnan He, Hanwang Zhang, Min-Yen Kan, and Tat-Seng Chua. 2016. Fast matrix factorization for online recommendation with implicit feedback. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. ACM, 549–558.

[50] Balázs Hidasi and Alexandros Karatzoglou. 2017. Recurrent Neural Networks with Top-k Gains for Session-based Recommendations. *arXiv preprint arXiv:1706.03847* (2017).

[51] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2015. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939* (2015).

[52] Balázs Hidasi, Alexandros Karatzoglou, Oren Sar-Shalom, Sander Dieleman, Bracha Shapira, and Domonkos Tikk. 2017. DLRS 2017: Second Workshop on Deep Learning for Recommender Systems. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*. ACM, 370–371.

[53] Balázs Hidasi, Massimo Quadrana, Alexandros Karatzoglou, and Domonkos Tikk. 2016. Parallel recurrent neural network architectures for feature-rich session-based recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems*. ACM, 241–248.

[54] Balázs Hidasi and Domonkos Tikk. 2016. General factorization framework for context-aware recommendations. *Data Mining and Knowledge Discovery* 30, 2 (2016), 342–371.

[55] Liang Hu, Jian Cao, Guandong Xu, Longbing Cao, Zhiping Gu, and Can Zhu. 2013. Personalized recommendation via cross-domain triadic factorization. In *Proceedings of the 22nd international conference on World Wide Web*. ACM, 595–606.

[56] Liang Hu, Jian Cao, Guandong Xu, Jie Wang, Zhiping Gu, and Longbing Cao. 2013. Cross-domain collaborative filtering via bilinear multilevel analysis. In *International Joint Conference on Artificial Intelligence*. IJCAI/AAAI.

[57] Liang Hu, Longbing Cao, Shoujin Wang, Guandong Xu, Jian Cao, and Zhiping Gu. 2017. Diversifying personalized recommendation with user-session context. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, Vol. 1943.

[58] Cheng-Lung Huang and Wei-Liang Huang. 2009. Handling sequential pattern decay: Developing a two-stage collaborative recommender system. *Electronic Commerce Research and Applications* 8, 3 (2009), 117–129.

[59] Zan Huang, Hsinchun Chen, and Daniel Zeng. 2004. Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering. *ACM Transactions on Information Systems (TOIS)* 22, 1 (2004), 116–142.

[60] Dietmar Jannach. [n. d.]. Challenges of Session-Aware Recommendation in E-Commerce (Keynote). ([n. d.]).

[61] Dietmar Jannach. 2018. Keynote: Session-Based Recommendation–Challenges and Recent Advances. In *Joint German/Austrian Conference on Artificial Intelligence (Künstliche Intelligenz)*. Springer, 3–7.

[62] Dietmar Jannach and Malte Ludewig. 2017. Determining characteristics of successful recommendations from log data: a case study. In *Proceedings of the Symposium on Applied Computing*. ACM, 1643–1648.

[63] Dietmar Jannach and Malte Ludewig. 2017. When recurrent neural networks meet the neighborhood for session-based recommendation. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*. ACM, 306–310.

[64] Dietmar Jannach, Malte Ludewig, and Lukas Lerche. 2017. Session-based item recommendation in e-commerce: on short-term intents, reminders, trends and discounts. *User Modeling and User-Adapted Interaction* 27, 3-5 (2017), 351–392.

[65] Songlei Jian, Longbing Cao, Kai Lu, and Hang Gao. 2018. Unsupervised Coupled Metric Similarity for Non-IID Categorical Data. *IEEE Trans. Knowl. Data Eng.* 30, 9 (2018), 1810–1823.

[66] Songlei Jian, Longbing Cao, Guansong Pang, Kai Lu, and Hang Gao. 2017. Embedding-based Representation of Categorical Data by Hierarchical Value Coupling Learning. In *IJCAI'2017*. 1937–1943.

[67] How Jing and Alexander J Smola. 2017. Neural survival recommender. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*. ACM, 515–524.

[68] Saikishore Kalloori and Francesco Ricci. 2017. Improving Cold Start Recommendation by Mapping Feature-Based Preferences to Item Comparisons. In *Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization*. ACM, 289–293.

[69] Alexandros Karatzoglou and Balázs Hidasi. 2017. Deep Learning for Recommender Systems. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*. ACM, 396–397.

[70] Alexandros Karatzoglou, Balázs Hidasi, Domonkos Tikk, Oren Sar-Shalom, Haggai Roitman, Bracha Shapira, and Lior Rokach. 2016. RecSys' 16 Workshop on Deep Learning for Recommender Systems (DLRS). In *Proceedings of the 10th ACM Conference on Recommender Systems*. ACM, 415–416.

[71] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 42, 8 (2009).

[72] Balaji Krishnamurthy, Nikaash Puri, and Raghavender Goel. 2016. Learning vector-space representations of items for recommendations using word embedding models. *Procedia Computer Science* 80 (2016), 2205–2210.

[73] Duc-Trong Le, Yuan Fang, and Hady W Lauw. 2016. Modeling sequential preferences with dynamic user and context factors. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 145–161.

[74] C-H Lee, Y-H Kim, and P-K Rhee. 2001. Web personalization expert with combining collaborative filtering and association rule mining technique. *Expert Systems with Applications* 21, 3 (2001), 131–137.

[75] Lukas Lerche, Dietmar Jannach, and Malte Ludewig. 2016. On the value of reminders within e-commerce recommendations. In *Proceedings of the 2016 Conference on User Modeling Adaptation and Personalization*. ACM, 27–35.

[76] Bin Li, Qiang Yang, and Xiangyang Xue. 2009. Transfer learning for collaborative filtering via a rating-matrix generative model. In *Proceedings of the 26th annual international conference on machine learning*. ACM, 617–624.

[77] Fangfang Li, Guandong Xu, and Longbing Cao. 2015. Coupled Matrix Factorization Within Non-IID Context. In *PAKDD'2015, Part II*. 707–719.

[78] Jing Li, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Tao Lian, and Jun Ma. 2017. Neural attentive session-based recommendation. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. ACM, 1419–1428.

[79] Sheng Li, Jaya Kawale, and Yun Fu. 2015. Deep collaborative filtering via marginalized denoising auto-encoder. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*. ACM, 811–820.

[80] Yuqi Li, Weizheng Chen, and Hongfei Yan. 2017. Learning Graph-based Embedding For Time-Aware Product Recommendation. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. ACM, 2163–2166.

[81] Defu Lian, Vincent W Zheng, and Xing Xie. 2013. Collaborative filtering meets next check-in location prediction. In *Proceedings of the 22nd International Conference on World Wide Web*. ACM, 231–232.
[82] Dawen Liang, Jaan Altosaar, Laurent Charlin, and David M Blei. 2016. Factorization meets the item embedding: Regularizing matrix factorization with item co-occurrence. In *Proceedings of the 10th ACM conference on recommender systems*. ACM, 59–66.
[83] Weiyang Lin, Sergio A Alvarez, and Carolina Ruiz. 2002. Efficient adaptive-support association rule mining for recommender systems. *Data mining and knowledge discovery* 6, 1 (2002), 83–105.
[84] Xiang Lin, Shuzi Niu, Yiqiao Wang, and Yucheng Li. 2018. K-plet Recurrent Neural Networks for Sequential Recommendation. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. ACM, 1057–1060.
[85] Greg Linden, Brent Smith, and Jeremy York. 2003. Amazon. com recommendations: Item-to-item collaborative filtering. *IEEE Internet computing* 7, 1 (2003), 76–80.
[86] Duen-Ren Liu, Chin-Hui Lai, and Wang-Jung Lee. 2009. A hybrid of sequential rules and collaborative filtering for product recommendation. *Information Sciences* 179, 20 (2009), 3505–3519.
[87] Qiao Liu, Yifu Zeng, Refuoe Mokhosi, and Haibin Zhang. 2018. STAMP: Short-Term Attention/Memory Priority Model for Session-based Recommendation. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 1831–1839.
[88] Xin Liu, Yong Liu, Karl Aberer, and Chunyan Miao. 2013. Personalized point-of-interest recommendation by mining users' preference transition. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*. ACM, 733–738.
[89] Babak Loni, Yue Shi, Martha Larson, and Alan Hanjalic. 2014. Cross-domain collaborative filtering with factorization machines. In *European conference on information retrieval*. Springer, 656–661.
[90] Pablo Loyola, Chen Liu, and Yu Hirate. 2017. Modeling User Session and Intent with an Attention-based Encoder-Decoder Architecture. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*. ACM, 147–151.
[91] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013).
[92] Tomas Mikolov, Quoc V Le, and Ilya Sutskever. 2013. Exploiting similarities among languages for machine translation. *arXiv preprint arXiv:1309.4168* (2013).
[93] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. 3111–3119.
[94] Bamshad Mobasher, Honghua Dai, Tao Luo, and Miki Nakagawa. 2001. Effective personalization based on association rule discovery from web usage data. In *Proceedings of the 3rd international workshop on Web information and data management*. ACM, 9–15.
[95] C Romero Morales, AR Porras Pérez, S Ventura Soto, C Hervás Martınez, and A Zafra. 2006. Using sequential pattern mining for links recommendation in adaptive hypermedia educational systems. *Current Developments in Technology-Assisted Education* 2 (2006), 1016–1020.
[96] María N Moreno, Francisco J García, M José Polo, and Vivian F López. 2004. Using association analysis of web data in recommender systems. In *International Conference on Electronic Commerce and Web Technologies*. Springer, 11–20.
[97] Utpala Niranjan, RBV Subramanyam, and V Khanaa. 2010. Developing a web recommendation system based on closed sequential patterns. In *International Conference on Advances in Information and Communication Technologies*. Springer, 171–179.
[98] Makbule Gulcin Ozsoy. 2016. From word embeddings to item recommendation. *arXiv preprint arXiv:1601.01356* (2016).
[99] Roberto Pagano, Paolo Cremonesi, Martha Larson, Balázs Hidasi, Domonkos Tikk, Alexandros Karatzoglou, and Massimo Quadrana. 2016. The contextual turn: From context-aware to context-driven recommender systems. In *Proceedings of the 10th ACM conference on recommender systems*. ACM, 249–252.
[100] Sinno Jialin Pan and Qiang Yang. 2010. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering* 22, 10 (2010), 1345–1359.
[101] Weike Pan, Evan Wei Xiang, Nathan Nan Liu, and Qiang Yang. 2010. Transfer Learning in Collaborative Filtering for Sparsity Reduction.. In *AAAI*, Vol. 10. 230–235.
[102] Weike Pan and Qiang Yang. 2013. Transfer learning in heterogeneous collaborative filtering domains. *Artificial intelligence* 197 (2013), 39–55.
[103] Guansong Pang, Longbing Cao, and Ling Chen. 2016. Outlier Detection in Complex Categorical Data by Modeling the Feature Value Couplings.. In *IJCAI*. 1902–1908.
[104] Guansong Pang, Longbing Cao, Ling Chen, and Huan Liu. 2017. Learning Homophily Couplings from Non-IID Data for Joint Feature Selection and Noise-Resilient Outlier Detection. In *IJCAI'2017*. 2585–2591.
[105] Guansong Pang, Hongzuo Xu, Longbing Cao, and Wentao Zhao. 2017. Selective Value Coupling Learning for Detecting Outliers in High-Dimensional Categorical Data. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. ACM, 807–816.
[106] Keunchan Park, Jisoo Lee, and Jaeho Choi. 2017. Deep Neural Networks for News Recommendations. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. ACM, 2255–2258.

[107] Michael J Pazzani and Daniel Billsus. 2007. Content-based recommendation systems. In *The adaptive web*. Springer, 325–341.

[108] Wenjie Pei, Jie Yang, Zhu Sun, Jie Zhang, Alessandro Bozzon, and David MJ Tax. 2017. Interacting Attention-gated Recurrent Networks for Recommendation. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. ACM, 1459–1468.

[109] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 1532–1543.

[110] Ladislav Peska and Peter Vojtas. 2017. Using implicit preference relations to improve recommender systems. *Journal on Data Semantics* 6, 1 (2017), 15–30.

[111] Massimo Quadrana, Alexandros Karatzoglou, Balázs Hidasi, and Paolo Cremonesi. 2017. Personalizing Session-based Recommendations with Hierarchical Recurrent Neural Networks. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*. ACM, 130–137.

[112] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2010. Factorizing personalized markov chains for next-basket recommendation. In *Proceedings of the 19th international conference on World wide web*. ACM, 811–820.

[113] Massimiliano Ruocco, Ole Steinar Lillestøl Skrede, and Helge Langseth. 2017. Inter-Session Modeling for Session-Based Recommendation. In *Proceedings of the 2nd Workshop on Deep Learning for Recommender Systems*. ACM, 24–31.

[114] Jeff J Sandvig, Bamshad Mobasher, and Robin Burke. 2007. Robustness of collaborative recommendation based on association rule mining. In *Proceedings of the 2007 ACM conference on Recommender systems*. ACM, 105–112.

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

[116] 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 *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*. ACM, 513–521.

[117] Guy Shani, David Heckerman, and Ronen I Brafman. 2005. An MDP-based recommender system. *Journal of Machine Learning Research* 6, Sep (2005), 1265–1295.

[118] Bo Shao, Dingding Wang, Tao Li, and Mitsunori Ogihara. 2009. Music recommendation based on acoustic features and user access patterns. *IEEE Transactions on Audio, Speech, and Language Processing* 17, 8 (2009), 1602–1611.

[119] 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. *ACM Computing Surveys (CSUR)* 47, 1 (2014), 3.

[120] Elena Smirnova and Flavian Vasile. 2017. Contextual Sequence Modeling for Recommendation with Recurrent Neural Networks. *arXiv preprint arXiv:1706.07684* (2017).

[121] Wei Song and Kai Yang. 2014. Personalized Recommendation Based on Weighted Sequence Similarity. In *Practical Applications of Intelligent Systems*. Springer, 657–666.

[122] Yang Song, Ali Mamdouh Elkahky, and Xiaodong He. 2016. Multi-rate deep learning for temporal recommendation. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. ACM, 909–912.

[123] Xiaoyuan Su and Taghi M Khoshgoftaar. 2009. A survey of collaborative filtering techniques. *Advances in artificial intelligence* 2009 (2009), 4.

[124] Yong Kiam Tan, Xinxing Xu, and Yong Liu. 2016. Improved recurrent neural networks for session-based recommendations. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*. ACM, 17–22.

[125] Jiaxi Tang and Ke Wang. 2018. Personalized Top-N Sequential Recommendation via Convolutional Sequence Embedding. (2018).

[126] Jie Tang, Sen Wu, Jimeng Sun, and Hang Su. 2012. Cross-domain collaboration recommendation. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 1285–1293.

[127] Trinh Xuan Tuan and Tu Minh Phuong. 2017. 3D Convolutional Networks for Session-based Recommendation with Content Features. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*. ACM, 138–146.

[128] Bartłomiej Twardowski. 2016. Modelling contextual information in session-aware recommender systems with neural networks. In *Proceedings of the 10th ACM Conference on Recommender Systems*. ACM, 273–276.

[129] Moshe Unger. 2015. Latent context-aware recommender systems. In *Proceedings of the 9th ACM Conference on Recommender Systems*. ACM, 383–386.

[130] Flavian Vasile, Elena Smirnova, and Alexis Conneau. 2016. Meta-Prod2Vec: Product embeddings using side-information for recommendation. In *Proceedings of the 10th ACM Conference on Recommender Systems*. ACM, 225–232.

[131] Maksims Volkovs, Guang Wei Yu, and Tomi Poutanen. 2017. Content-based Neighbor Models for Cold Start in Recommender Systems. In *Proceedings of the Recommender Systems Challenge 2017*. ACM, 7.

[132] Shengxian Wan, Yanyan Lan, Pengfei Wang, Jiafeng Guo, Jun Xu, and Xueqi Cheng. 2015. Next Basket Recommendation with Neural Networks.. In *RecSys Posters*.

[133] Can Wang, Longbing Cao, Mingchun Wang, Jinjiu Li, Wei Wei, and Yuming Ou. 2011. Coupled nominal similarity in unsupervised learning. In *Proceedings of the 20th ACM international conference on Information and knowledge management*. ACM, 973–978.

[134] Can Wang, Chi-Hung Chi, Wei Zhou, and Raymond K Wong. 2015. Coupled Interdependent Attribute Analysis on Mixed Data.. In *AAAI*. 1861–1867.

[135] Can Wang, Xiangjun Dong, Fei Zhou, Longbing Cao, and Chi-Hung Chi. 2015. Coupled attribute similarity learning on categorical data. *IEEE transactions on neural networks and learning systems* 26, 4 (2015), 781–797.
[136] Dongjing Wang, Shuiguang Deng, and Guandong Xu. 2017. Sequence-based context-aware music recommendation. *Information Retrieval Journal* (2017), 1–23.
[137] Dongjing Wang, Shuiguang Deng, Xin Zhang, and Guandong Xu. 2016. Learning music embedding with metadata for context aware recommendation. In *Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval*. ACM, 249–253.
[138] Pengfei Wang, Jiafeng Guo, Yanyan Lan, Jun Xu, Shengxian Wan, and Xueqi Cheng. 2015. Learning hierarchical representation model for nextbasket recommendation. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*. ACM, 403–412.
[139] Shoujin Wang and Longbing Cao. 2017. Inferring implicit rules by learning explicit and hidden item dependency. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* (2017).
[140] Shoujin Wang, Liang Hu, and Longbing Cao. 2017. Perceiving the Next Choice with Comprehensive Transaction Embeddings for Online Recommendation. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 285–302.
[141] Shoujin Wang, Liang Hu, Longbing Cao, Xiaoshui Huang, Defu Lian, and Wei Liu. 2018. Attention-based Transactional Context Embedding for Next-Item Recommendation. (2018).
[142] Shoujin Wang, Wei Liu, Jia Wu, Longbing Cao, Qinxue Meng, and Paul J Kennedy. 2016. Training deep neural networks on imbalanced data sets. In *Neural Networks (IJCNN), 2016 International Joint Conference on*. IEEE, 4368–4374.
[143] Wei Wei, Xuhui Fan, Jinyan Li, and Longbing Cao. 2012. Model the complex dependence structures of financial variables by using canonical vine. In *CIKM'12*. 1382–1391.
[144] Chen Wu and Ming Yan. 2017. Session-aware Information Embedding for E-commerce Product Recommendation. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. ACM, 2379–2382.
[145] Chao-Yuan Wu, Amr Ahmed, Alex Beutel, Alexander J Smola, and How Jing. 2017. Recurrent recommender networks. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*. ACM, 495–503.
[146] Xiang Wu, Qi Liu, Enhong Chen, Liang He, Jingsong Lv, Can Cao, and Guoping Hu. 2013. Personalized next-song recommendation in online karaokes. In *Proceedings of the 7th ACM conference on Recommender systems*. ACM, 137–140.
[147] Jia Xu and Longbing Cao. 2018. Vine Copula-Based Asymmetry and Tail Dependence Modeling. In *PAKDD'2018, Part I*. 285–297.
[148] Liang Yan and Chunping Li. 2006. Incorporating pageview weight into an association-rule-based web recommendation system. In *Australasian Joint Conference on Artificial Intelligence*. Springer, 577–586.
[149] Ghim-Eng Yap, Xiao-Li Li, and S Yu Philip. 2012. Effective next-items recommendation via personalized sequential pattern mining. In *International Conference on Database Systems for Advanced Applications*. Springer, 48–64.
[150] Haochao Ying, Fuzhen Zhuang, Fuzheng Zhang, Yanchi Liu, Guandong Xu, Xing Xie, Hui Xiong, and Jian Wu. 2018. Sequential Recommender System based on Hierarchical Attention Networks. In *the 27th International Joint Conference on Artificial Intelligence*.
[151] Feng Yu, Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan. 2016. A dynamic recurrent model for next basket recommendation. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. ACM, 729–732.
[152] Xi-Zheng Zhang. 2007. Building personalized recommendation system in e-commerce using association rule-based mining and classification. In *Machine Learning and Cybernetics, 2007 International Conference on*, Vol. 7. IEEE, 4113–4118.
[153] Yuqi Zhang and Jian Cao. 2013. Personalized Recommendation Based on Behavior Sequence Similarity Measures. In *Behavior and Social Computing*. Springer, 165–177.
[154] Zhiyong Zhang and Olfa Nasraoui. 2007. Efficient hybrid Web recommendations based on Markov clickstream models and implicit search. In *Web Intelligence, IEEE/WIC/ACM International Conference on*. IEEE, 621–627.
[155] Wei Zhao, Wenyou Wang, Jianbo Ye, Yongqiang Gao, Min Yang, Zhou Zhao, and Xiaojun Chen. 2017. Leveraging Long and Short-term Information in Content-aware Movie Recommendation. *arXiv preprint arXiv:1712.09059* (2017).
[156] Wayne Xin Zhao, Jin Huang, and Ji-Rong Wen. 2016. Learning distributed representations for recommender systems with a network embedding approach. In *Asia Information Retrieval Symposium*. Springer, 224–236.
[157] Baoyao Zhou, Siu Cheung Hui, and Alvis Cheuk Ming Fong. 2006. Efficient sequential access pattern mining for web recommendations. *International Journal of Knowledge-based and Intelligent Engineering Systems* 10, 2 (2006), 155–168.
[158] Ke Zhou, Shuang-Hong Yang, and Hongyuan Zha. 2011. Functional matrix factorizations for cold-start recommendation. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*. ACM, 315–324.
[159] Chengzhang Zhu, Longbing Cao, Qiang Liu, Jianping Yin, and Vipin Kumar. 2018. Heterogeneous Metric Learning of Categorical Data with Hierarchical Couplings. *IEEE Trans. Knowl. Data Eng.* 30, 7 (2018), 1254–1267.