Рекомендательные системы на основе сессий – модели и задачи

Д.Р. Якупов, Д.Е. Намиот

***Аннотация*— Рекомендательные системы были одним из первых массовых применений анализа данных в самых разных областях. Причиной является прозрачный для конечных пользователей их конечный результат (рекомендации) и понятные метрики измерения качества их работы. Конечные пользователи всегда могут оценить полезность рекомендаций, формальные измерения всегда могут оперировать конверсией, чтобы под ней не подразумевалось – покупки рекомендованных товаров, переходы по ссылкам и т.п. Чаще всего работа рекомендательных систем основана на обобщении и анализе предпочтений других пользователей (что включает рассмотрение различных аспектов их поведения), и имеющейся информации о текущем пользователе. Вместе с тем, есть класс задач, когда рекомендации должны (или только и могут) основываться на текущих действиях пользователя. Например, в системе электронной коммерции неавторизованный (анонимный) пользователь посещает различные страницы сайта. Или предпочтения пользователя в системе носят только краткосрочный характер. Все эти примеры характерны для отдельного большого класса рекомендательных систем – рекомендательных систем для сессий, где под сессией понимается последовательность действий пользователя. Рекомендательная система в таком случае решает одну из трех задач: рекомендует следующий товар (контент, активность и т.п.) в рамках текущей сессии, рекомендует следующие товары (активности и т.п.) до конца текущей сессии, рекомендует следующую возможную сессию. В статье содержится обзор описанных задач и моделей для такого рода рекомендательных систем.**

***Ключевые слова*— рекомендательные системы для сессий, краткосрочные предпочтения**

I. Введение

В современном мире в условиях перегруженности пользователей информационных систем огромным количеством различного рода информацией о товарах, услугах, событиях и т.д. все более важную роль начинают играть системы рекомендаций или рекомендательные системы.

Статья получена 20 июня 2022. Исследование выполнено при поддержке Междисциплинарной научно-образовательной школы Московского университета «Мозг, когнитивные системы, искусственный интеллект»

Д.Р. Якупов – МГУ имени М.В. Ломоносова (email: dyakupov1@gmail.com)
Д.Е. Намиот – МГУ имени М.В. Ломоносова (email: dnamiot@gmail.com)

В общем случае, системы рекомендаций представляют собой вспомогательные системы, которые помогают пользователям находить информацию, продукты или услуги (такие как книги, фильмы, музыку, цифровые продукты, веб-сайты, телепрограммы и т.д.). Чаще всего это делается посредством обобщения и анализа предпочтений других пользователей (что означает рассмотрение различных аспектов их поведения), и атрибутов (характерных особенностей) самого текущего пользователя [1].

Системы рекомендаций превратились в фундаментальный инструмент для принятия более информативных, эффективных и действенных решений практически во всех областях современной жизни, включая бизнес, финансы, здравоохранение, транспорт, образование, общение, развлечения и т.д. [2].

В настоящее время существуют различные подходы [8, 9] к генерированию рекомендаций в системах рекомендаций, такие как генерирование рекомендаций на основе контента (Content-based recommender systems) [3], совместной фильтрации (Collaborative filtering recommender systems) [4], использование гибридного подхода (Hybrid recommender systems) [5], объединяющего в себе подходы на основе контента и совместной фильтрации, и другие.

Однако, в основном, данные подходы, как правило, используют информацию обо всех исторических взаимодействиях пользователя с элементами (под элементом здесь и далее понимается продукт или услуга, предназначенная для рекомендации пользователю, например, книга, фильм, музыка и т.д.), для выявления долгосрочных и статических предпочтений каждого пользователя в отношении элементов. Такая практика часто связана с основополагающим предположением о том, что все исторические взаимодействия пользователя одинаково важны для его текущих предпочтений. В реальности это может быть не так по следующим причинам [6]:

Во-первых, выбор пользователем элементов зависит не только от его долгосрочных исторических предпочтений, но также и от его недавних краткосрочных предпочтений и контекста, учитывающего временной фактор (например, недавно просмотренные или купленные товары). Информация об этих краткосрочных предпочтениях заложена в самых последних взаимодействиях пользователя [7], которые часто составляют небольшую часть его исторических

взаимодействий.

Во-вторых, предпочтения пользователя в отношении элементов, как правило, не статичны, а динамичны, то есть развиваются с течением времени.

В-третьих, что часть является определяющим, истории действий пользователя может просто не быть. Представьте веб-пользователя зашедшего на сайт электронного магазина, аккаунта на сайте нет, доступ часто осуществляется через какой-то прокси-сервер (часто – не один такой сервер). Информации о посетителе просто нет. Все что есть (если пользователь не покинул сайт сразу со страницы входа) – это последовательность его действий на сайте. Которую и желательно как-то использовать для того, чтобы не отпустить посетителя без покупки.

В целях устранения данных недостатков, в последнее время исследователями все больше внимания уделяется системам рекомендаций на основе сессий (Session-based Recommender Systems, далее также - SBRS), которые, в отличие от вышеупомянутых подходов, анализируют краткосрочные предпочтения пользователей и динамику их изменения.

Каждая сессия состоит из нескольких взаимодействий пользователя с элементами системы, которые происходят совместно в течение непрерывного периода времени. Рассматривая каждую сессию в качестве базовой единицы входных данных, SBRS может выявлять как краткосрочные предпочтения пользователя из его последних сессий, так и динамику его предпочтений, отражающую изменение его предпочтений от сессии к сессии, и использовать данную информацию для генерирования более точных и своевременных рекомендаций.

II. Общая постановка задачи для SBRS

A. Классификация решаемых SBRS задач

Решаемые SBRS задачи по генерированию рекомендаций можно условно разделить на три категории [6], а именно:

- рекомендация следующего взаимодействия в текущей сессии;
- рекомендация оставшейся части сессии (т.е. полный список оставшихся взаимодействий для завершения текущей сессии);
- рекомендация по содержанию следующей сессии.

Сравнение данных категорий задач представлено в таб. 1.

Таблица 1. Данные сессий

№	Категория	Входные данные SBRS	Выходные данные SBRS	Примеры рекомендаций
1	Рекомендация следующего взаимодействия	- Данные, известные о текущей сессии; - Комбинация данных, известных о текущей сессии, и исторических данных	Следующее взаимодействие (элемент)	Рекомендация следующего товара, услуги, веб-страницы, новостной статьи, музыкального произведения, фильма и т.д.
2	Рекомендация оставшейся части сессии		Полный список взаимодействий для завершения текущей сессии	Рекомендация списка следующих элементов (товаров, услуг и т.д.) для завершения оформления корзины покупок в текущей сессии.
3	Рекомендация по содержанию следующей сессии	Данные, известные о прошлых сессиях (исторические данные)	Следующая сессия	Рекомендация по составу корзины покупок, пакета услуг и т.д. в следующей сессии.

B. Входные и выходные данные SBRS

Входные данные SBRS называют контекстом сессии. SBRS генерирует рекомендации в зависимости от контекста текущей сессии.

В зависимости от категории решаемой задачи в качестве входных данных может использоваться следующая информация:

1. данные, известные о текущей сессии – представляют собой список взаимодействий, произошедший в сессии к текущему моменту времени;
2. данные, известные о прошлых сессиях – исторические данные;
3. комбинация пунктов 1) и 2).

Входные данные 1) используются SBRS анализирующих зависимости внутри сессии и генерирующих рекомендации по следующему взаимодействию в текущей сессии, либо списку взаимодействий для завершения текущей сессии (оставшейся части сессии). Например, рекомендация следующего товара, либо списка всех последующих товаров для завершения заполнения корзины покупок с учетом товаров, добавленных в нее ранее.

Входные данные 2) используются для SBRS анализирующих зависимости между сессиями и генерирующих рекомендации по содержанию следующей сессии (например, корзины продуктов в интернет магазине для следующей сессии);

Входные данные 3) используются для SBRS анализирующих зависимости как внутри сессии, так и между сессиями и генерирующих рекомендации по следующему взаимодействию (списку взаимодействий для завершения текущей сессии).

Выходные данные SBRS представляют собой прогнозируемое взаимодействие (список взаимодействий), которые произойдут в текущей или в следующих сессиях.

В зависимости от категории решаемой задачи выходные данные могут представлять собой:

- рекомендации по следующему взаимодействию – представляет собой ранжированный по наилучшему соответствию список альтернативных взаимодействий (элементов);
- рекомендации по списку взаимодействий для завершения текущей сессии – например, сформированный список аксессуаров для выбранного (добавленного в корзину покупок) смартфона;
- рекомендации по следующей сессии – представляет собой список дополнительных взаимодействий (элементов) для формирования следующей сессии (например, сформированная корзина покупок продуктов питания для достижения общей цели: завтрак, обед и т.д.).

C. Основные сущности SBRS

Основные сущности, используемые при описании SBRS [6]:

пользователь (u) – это субъект, который совершает какие-либо действия (нажатие кнопки «мыши», переход между товарами, покупка товара, и т.д.) и который, в результате своих действий, получает рекомендации от SBRS;

элемент (v) – это объект (продукт или услуга), предназначенный для рекомендации пользователю (товар на сайте магазина, курс на сайте образовательной организации, статья на сайте СМИ и т.д.);

действие (a) – это манипуляция, выполняемая пользователем в рамках сессии;

взаимодействие (o) – это троичный кортеж, состоящий из пользователя (u), элемента (v) и действия (a), выполняемого пользователем u над элементом v ($o = \langle u, v, a \rangle$).

сессия (s) – это непустой ограниченный список взаимодействий, произошедших в течение непрерывного периода времени ($s = \{o_1, o_2, \dots, o_{|S|}\}$).

Рассмотрим более подробно основные свойства каждой сущности SBRS.

1) Пользователь

Каждый пользователь имеет свой уникальный идентификатор и набор атрибутов, описывающих его характеристики (пол, возраст, вес и т.д.).

Атрибуты пользователя могут влиять на совершаемые им действия в текущей или последующих сессиях. Например, мужчины могут чаще выбирать к просмотру фильмы жанра «боевик», в то время женщины предпочитают смотреть мелодрамы.

Атрибуты пользователя разделяют на явные и неявные. К явным атрибутам относят атрибуты, которые могут быть четко определены (пол, возраст, вес и т.д.). К неявным атрибутам относят атрибуты, отражающие внутреннее состояние пользователя, например, его настроение или намерения.

Наряду с явными атрибутами неявные атрибуты, также могут оказывать существенное влияние на действия пользователя.

Совокупность всех пользователей образует собой множество пользователей системы $U = \{u_1, u_2, \dots, u_{|U|}\}$..

2) Элемент

Каждый элемент имеет свой уникальный идентификатор и набор атрибутов, описывающих его характеристики (категория, цвет, цена и т.д.)

Совокупность всех элементов образует собой множество элементов системы $V = \{v_1, v_2, \dots, v_{|V|}\}$.

3) Действие

Каждое действие имеет свой уникальный идентификатор и набор атрибутов, описывающих его характеристики (например, тип действия: просмотр, выбор, покупка элемента и т.д.).

Действие может выполняться как над элементом SBRS, так и не быть связанным с каким-либо конкретным элементом (например, поиск или навигация по каталогу).

Совокупность всех возможных действий образует собой множество действий $A = \{a_1, a_2, \dots, a_{|A|}\}$.

4) Взаимодействие

В общем случае взаимодействие включает в себя информацию о пользователе (u), элементе (v) и действие (a), выполняемым пользователем u над элементом v (o=⟨u,v,a⟩).

В частном случае, когда информация о пользователе недоступна (анонимные сессии) взаимодействие упрощается до $o = \langle v, a \rangle$.

Отметим, что обычно информацию о пользователе прогнозировать не требуется, так как, в случае персонализированных сессий она доступна по умолчанию, а в случае анонимных сессий, она недоступна, и, следовательно, непредсказуема.

Кроме того, когда в SBRS определен только один тип действий, содержание прогнозируемого взаимодействия для анонимных сессий упрощается до элемента $o = \langle v \rangle$.

Совокупность всех взаимодействий образует собой множество взаимодействий $O = \{o_1, o_2, \dots, o_{|O|}\}$.

5) Сессия

Выделяют пять основных свойств [6], характеризующих сессию:

- длительность сессии – это суммарное количество взаимодействий в рамках сессии;
- наличие внутреннего порядка – определяет наличие (отсутствие) хронологического порядка во взаимодействиях в рамках сессии;
- набор доступных действий – определяет

количество различных типов действий, которые пользователь может выполнить в рамках сессии;

- доступность информации о пользователе – определяет наличие (отсутствие) информации о пользователе сессии;
- структура данных – определяет наличие иерархической структуры в данных сессии, либо в данных связанных между собой нескольких сессий.

Совокупность всех сессий образует собой множество сессий $S = \{s_1, s_2, \dots, s_{|S|}\}$.

Классификация сессий в соответствии с рассмотренными выше характеристиками приведена на рис. 1, более подробно каждое из свойств сессии рассматривается в разделе 3.

Рис. 1. Классификация сессий

D. Формализация постановки задачи для SBRS

В общем виде задача SBRS заключается в том, чтобы комплексно проанализировать зависимости между взаимодействиями внутри и (или) между сессиями, а затем на основе изученных зависимостей построить прогноз последующих взаимодействий и сгенерировать соответствующие рекомендации.

Формализуем задачу для SBRS [6].

Пусть:

- $l = \{o_1, o_2, \dots, o_n\}$ - это список из n взаимодействий, где каждое взаимодействие состоит из элемента (v) и действия (a), выполняемого пользователем над элементом v.
- L – это множество списков, которое содержит все возможные списки взаимодействий, полученные из множества элементов V и множества действий A.
- C – это множество всех возможных контекстов сессии (c). Контекст сессии c включает в себя всю информацию о сессии, используемую для формирования рекомендаций, и подается на вход SBRS.
- $f(c,l)$ – функция полезности, возвращающая оценку полезности для оцениваемого списка взаимодействий l (кандидатов в рекомендации) и текущего контекста сессии c.

Тогда задачей SBRS является выбрать для рекомендации такой список взаимодействий $\hat{l}$ $(\hat{l} \in L)$, при котором функция полезности будет достигать максимального значения, т.е.:

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

Функция полезности может задаваться в различных формах (например, вероятность, условная вероятность и т.д.).

III Анализ основных свойств сессии

A. Длительность сессии

Сессии, по длительности, условно разделяют на длительные, средние и короткие сессии [6]. Критерии классификации сессий могут варьироваться в зависимости от конкретных наборов данных. Для определенности будем считать сессии, содержащие более 10 взаимодействий, длительными, от 4 до 9 – средними, менее 4 – короткими.

Длительные сессии, по сравнению со средними и короткими, содержат больше контекстной информации, что позволяет формировать более точные рекомендации. С другой стороны, в связи с неопределенностью поведения пользователя, длительная сессия может содержать случайные взаимодействия, которые не имеют отношения к другим взаимодействиям внутри сессии, что приводит к появлению шумов и снижает точность рекомендаций.

Кроме того, длительные сессии могут содержать сложные зависимости. Например, отдаленные зависимости (long-range dependencies) [10], т.е. зависимости между двумя далеко расположенными друг от друг взаимодействиями в сессии, или зависимости высокого порядка (high-order dependencies) [11], т.е. зависимости между несколькими взаимодействиями в сессии.

Средние сессии, в отличие от длинных, с меньшей вероятностью будут содержать много случайных взаимодействий, при этом, они обычно содержат достаточное количество контекстной информации для формирования точных рекомендаций.

Короткие сессии содержит небольшое количество взаимодействий, и, соответственно, ограниченное количество контекстной информации. Например, в автономной анонимной сессии, состоящей из двух взаимодействий, единственной контекстной информацией для прогнозирования второго взаимодействия, является первое взаимодействие.

В связи с изложенным, в зависимости от длительности сессии, возникают различные проблемные вопросы и задачи требующие решения.

Общей задач для всех видов сессий является задача эффективного извлечения необходимой и точной контекстной информации. Для длительных сессий, также необходимо решать задачи по эффективному снижению информационного шума и анализу сложных зависимостей, а для коротких сессий - задачу по эффективному генерированию рекомендаций в условиях ограниченности контекстной информации.

Характерные особенности сессии и проблемные вопросы, требующие решения, в зависимости от длительности сессий приведены в таб. 2.

Таб. 2. Характерные особенности и проблемные вопросы в зависимости от длительности сессий

№	Длительность сессии	Характерные особенности	Проблемные вопросы
1	Длительные	1) Содержат много контекстной информации; 2) Могут содержать значительное количество случайных взаимодействий (информационный шум); 3) Могут содержать сложные зависимости (отдаленные и высокого порядка).	1) Снижение информационного шума, создаваемого случайными взаимодействиями; 2) Анализ сложных зависимостей; 3) Эффективное извлечение важной и точной контекстной информации.
2	Средние	1) Содержат достаточное количество контекстной информации; 2) Содержат незначительное количество случайных взаимодействий.	Эффективное извлечение важной и точной контекстной информации.
3	Короткие	Содержат мало контекстной информации.	1) Генерирование эффективных рекомендаций в условиях существенной ограниченности контекстной информации; 2) Эффективное извлечение важной и точной контекстной информации.

В Наличие внутреннего порядка

Сессии, по наличию внутреннего порядка, условно разделяют на упорядоченные, не упорядоченные и сессии с гибким порядком [6].

В неупорядоченной сессии отсутствует хронологический порядок между взаимодействиями, то есть, не имеет значения, в какой момент времени в сессии произойдет конкретное взаимодействие, раньше или позже других взаимодействий. Например, сессия покупок продуктов в интернет-магазине может быть неупорядоченной, так как пользователь может формировать корзину покупок (хлеб, яблоки, молоко и т.д.) не следуя какому-либо четкому порядку.

В неупорядоченных сессиях зависимости между взаимодействиями основаны на их совместном возникновении, а не на их последовательности.

Зависимости, основанные на совместном возникновении, по сравнению с последовательными зависимостями, обычно относительно слабые (нечеткие) и сложнее поддаются анализу.

Кроме того, большинство зависимостей, основанных на совместном возникновении, являются коллективными зависимостями, когда несколько взаимодействий совместно приводят к возникновению следующего взаимодействия, которое еще труднее предсказать.

В упорядоченной сессии взаимодействия строго упорядочены и обычно между ними существуют сильные последовательные зависимости. Например, в сессии покупок обучающих курсов на сайте образовательной организации для прохождения сложного курса может потребоваться сначала пройти базовые курсы для получения необходимых предварительных знаний.

Сильные последовательные зависимости в упорядоченных сессиях проще анализировать, чем слабые (основанные на совместном возникновении) зависимости в неупорядоченных сессиях. В тоже время, сложно эффективно анализировать каскадные долгосрочные последовательные зависимости, которые со временем постепенно затухают в длинных упорядоченных сессиях.

В сессиях с гибким порядком взаимодействия не являются полностью упорядоченными или полностью не упорядоченными, то есть некоторые части сессии являются упорядоченными, а другие нет. Например, на сайте бронирования пользователь бронирует себе авиабилеты, отель, трансфер до отеля, аренду автомобиля, экскурсии и досуг. Взаимодействия при бронировании авиабилетов, отеля, трансфера до отеля

последовательно зависимы, в тоже время, взаимодействия при бронировании аренды автомобиля, экскурсий и досуга могут производиться случайным образом без какого-либо порядка.

Особенностью сессий с гибким порядком является наличие сложных и смешанных зависимостей (mixed dependencies), то есть одновременное наличие последовательных зависимостей в упорядоченных взаимодействиях в сессии и непоследовательных зависимостей в неупорядоченных взаимодействиях.

Характерные особенности сессии и проблемные вопросы, требующие решения, в зависимости от внутреннего порядка приведены в таб. 3.

Таблица 3 Характерные особенности и проблемные вопросы в зависимости от внутреннего порядка

№	Наличие внутреннего порядка в сессии	Характерные особенности	Проблемные вопросы
1	Упорядоченные	1) Взаимодействия строго упорядочены; 2) Зависимости основаны на последовательностях взаимодействий; 3) Зависимости обычно сильные (четкие) и проще поддаются анализу. 4) Могут содержать каскадные долгосрочные последовательные зависимости постепенно затухающие в длинных сессиях.	Эффективный анализ каскадных долгосрочных последовательных зависимостей.
2	Не упорядоченные	1) Взаимодействия не упорядочены; 2) Зависимости основаны на совместном возникновении взаимодействий; 3) Зависимости обычно слабые (нечеткие) и сложнее поддаются анализу; 4) Большинство зависимостей являются коллективными.	Эффективный анализ слабых (нечетких) зависимостей (особенно коллективных).
3	С гибким порядком	1) Взаимодействия не являются полностью упорядоченными или не упорядоченными; 2) Наличие сложных и смешанных зависимостей.	Эффективный анализ сложных и смешанных зависимостей.

C. Набор доступных действий

Сессии, по набору доступных действий, разделяют на сессии, в которых доступен только один тип действий (например, выбор элемента), и сессии, в которых доступно несколько различных типов действий (например, выбор элемента, сравнение элементов, покупка элемента и т.д.) [6].

Сессии с одним типом действий относительно легко поддаются анализу, так как будут содержать только один тип взаимодействий и соответственно однотипные зависимости.

Напротив, сессии с несколькими типами действий, будут содержать различные типы взаимодействий, что приводит к возникновению сложных зависимостей. Например, в сессии покупок в интернет магазине пользователь обычно сначала выбирает несколько товаров, сравнивает их, а затем покупает один или несколько товаров. При этом, товары, выбираемые и сравниваемые пользователем, скорее всего, являются похожими товарами (товарами - конкурентами), а товары, добавляемые пользователем в корзину покупок – взаимодополняющими товарами. Таким образом, возникают зависимости не только между взаимодействиями одного типа (выбор товаров), но и между взаимодействиями разных типов (выбор, сравнение и покупка).

Характерные особенности сессии и проблемные вопросы, требующие решения в зависимости от доступного набора действий приведены в таб. 4.

Таблица 4. Характерные особенности и проблемные вопросы в зависимости от доступного набора действий

№	Доступные действия	Характерные особенности	Проблемные вопросы
1	Однотипные действия	1) Доступен только один тип действий; 2) Содержат только один тип взаимодействий; 3) Содержат однотипные зависимости (зависимости между взаимодействиями одного типа).	-
2	Разнотипные действия	1) Доступно несколько различных типов действий; 2) Содержат различные типы взаимодействий; 3) Содержат сложные зависимости, включающие как зависимости между однотипными взаимодействиями, так и между взаимодействиями разных типов.	Анализ сложных зависимостей.

D. Доступность информации о пользователе

Сессии, по доступности информации о пользователе, разделяют на персонализированные (информация доступна), и анонимные (информация не доступна) [6].

В персонализированных сессиях, взаимодействия содержат информацию о пользователе (его идентификаторе), что позволяет использовать информацию из его прошлых сессий, и узнать его долгосрочные предпочтения, а также их эволюцию от сессии к сессии. В тоже время, из-за относительно длительного периода времени и динамики предпочтений, довольно сложно точно определить долгосрочные предпочтения пользователя на основе его нескольких персонализированных сессий.

Информация о пользователе может быть не всегда доступна в связи с защитой его конфиденциальности, либо потому, что пользователь не осуществил вход в систему при взаимодействии с интернет-платформой. В таких случаях сессия становится анонимной и для подготовки рекомендаций можно использовать только контекстную информацию текущей сессии. Ограниченность данной информации усложняет определение индивидуальных предпочтений пользователя для генерирования точных рекомендаций.

Характерные особенности сессии и проблемные вопросы, требующие решения в зависимости от доступности информации о пользователе приведены в таб. 5.

E Структура данных сессии

Сессии, по структуре данных, условно разделяют на одноуровневые и многоуровневые [6].

Рассмотрим две разные сессии одного пользователя на сайте интернет-магазина (рис. 2). Пусть в рамках каждой из сессий пользователь рассматривает товары какого-либо определенного бренда. В данном случае можно выделить три уровня в иерархии данных: отношения на уровне атрибутов элементов (бренд товара), отношения на уровне взаимодействий над элементами внутри сессии (просмотр/покупка товара в рамках сессии), отношения на уровне сессий (объединяет в себе информацию о нескольких исторических сессиях текущего пользователя).

Таблица 5. Характерные особенности и проблемные вопросы в зависимости доступности информации о пользователе

№	Доступность информации о пользователе	Характерные особенности	Проблемные вопросы
1	Персонализированные сессии	1) Доступна информация о пользователе сессии; 2) Возможность использовать для анализа исторические данные из прошлых сессий пользователя.	Эффективное определение долгосрочных предпочтений пользователя на основе его нескольких персонализированных сессий
2	Анонимные сессии	1) Информация о пользователе сессии не доступна; 2) Для анализа доступна только контекстная информация текущей сессии.	Эффективное определение индивидуальных предпочтений пользователя в условиях ограниченности доступной контекстной информации.

Рис. 2. Пример иерархической структуры данных

Количество уровней в структуре данных сессии определяет объем информации, который может быть

использован для рекомендаций.

Уровень взаимодействия существует в любых сессиях, соответственно, в одноуровневых сессиях все данные относятся к уровню взаимодействия. В многоуровневых сессиях, в дополнение к уровню взаимодействия, данные могут включать уровень атрибутов и/или уровень сессий.

Отсутствие вспомогательной информации с других уровней в SBRS, построенных на одноуровневых сессиях, усугубляет ситуацию с проблемами холодного старта (cold-start) [12, 13] и/или разреженности данных (sparsity issues) [12, 13].

В многоуровневых сессиях на рекомендации влияют как зависимости внутри одного уровня иерархии, так и межуровневые зависимости. Например, категории нескольких товаров (уровень атрибутов) могут влиять на то, будут ли эти товары покупаться вместе в одной сессии (уровень взаимодействия).

Характерные особенности и проблемные вопросы, требующие решения, в зависимости от структуры данных сессии приведены в таб. 6.

Таблица 6. Характерные особенности и проблемные вопросы в зависимости структуры данных сессии

№	Структура данных сессии	Характерные особенности	Проблемные вопросы
1	Одноуровневые сессии	1) Все данные в сессии находятся на одном уровне иерархии; 2) Возможность использовать для анализа только данные уровня взаимодействия.	Эффективное решение проблем холодного старта и/или разреженности данных.
2	Многоуровневые сессии	Данные в сессии имеют иерархическую структуру, состоящую по меньшей мере из двух уровней (уровень взаимодействия и уровень атрибутов и/или уровень сессии). 2) Возможность использовать для анализа данные разных уровней.	Эффективный и всесторонний анализ как зависимостей внутри одного уровня иерархии, так и межуровневых зависимостей.

IV Обзор и сравнение подходов к реализации SBRS

Классификация подходов к реализации SBRS распространенных в настоящее время [6] приведена на рис. 3.

Рис. 3. Классификация подходов к реализации SBRS

В общем случае, традиционные подходы относительно просты, понятны и легки для понимания и реализации. При этом, несмотря на простоту, они эффективны в некоторых случаях, особенно для простых наборов данных, в которых зависимости внутри или между сессиями очевидны и легко моделируются и выявляются.

В частности, в исследовании [15] подходы, основанные на методе K ближайших соседей (K-Nearest Neighbor, KNN), например, session-KNN, достигли превосходной точности рекомендаций даже в сравнении с некоторыми подходами на основе глубоких нейронных сетей, например, GRU4Rec, за гораздо меньшее время работы с некоторыми наборами данных электронной коммерции.

Напротив, подходы, основанные на глубоких

нейронных сетях, обычно относительно сложны, включают сложную и многоуровневую сетевую архитектуру и часто требуют серьезных вычислений. Обычно считается, что они более эффективны для всестороннего моделирования и выявления сложных зависимостей, например, долгосрочных зависимостей или зависимостей высокого порядка, имеющихся в сложных наборах данных (например, несбалансированных или разреженных наборы данных) [16].

Превосходство подходов, основанных на глубоких нейронных сетях, было подтверждено множеством работ последних лет, например, [17, 18, 19].

Подходы, основанные на скрытом представлении (Latent Representation), немного сложнее традиционных подходов, но менее сложны, чем подходы, основанные на глубоких нейронных сетях. В отличие от подходов, основанных на глубоких нейронных сетях, они обычно не требуют глубокой сетевой архитектуры, что приводит к относительно низкой стоимости вычислений. В некоторых исследованиях [20, 21] подходы, основанные на скрытом представлении, смоги превзойти не только некоторые традиционные подходы, например, основанные на цепях Маркова [22], но также и некоторые подходы, основанные на глубоких нейронных сетях, например, на основе рекуррентной нейронной сети [17].

В таблице 7 приведено сравнение подходов в части типов зависимостей, которые они могут анализировать [6]:

Таблица 7. Сравнение подходов

№	Подход	Типы анализируемых зависимостей			
		Последовательные/ Непоследовательные	**Краткосрочные/ Долгосрочные**	**Первого порядка[1]/ Высокого порядка**	**Одиночные[2]/ Коллективные**
1	Анализ шаблонов/правил:	-	-	-	-
1.1	- анализ частых шаблонов *(frequent pattern mining)*	Непоследовательные	Оба типа	Оба типа	Оба типа
1.2	- анализ шаблонов последова-тельностей (*sequential pattern mining*)	Последовательные			
2	К-ближайших соседей:	-	-	-	-
2.1	- для элементов (*item-KNN*);	В основном непоследовательные	Оба типа	В основном первого порядка	Одиночные
2.2	- для сессий (*session-KNN*)				Коллективные
3	Цепь Маркова	Последовательные	Краткосрочные	Первого порядка	Одиночные
4	Генеративная вероятностная модель	Последовательные	Долгосрочные	Высокого порядка	Коллективные
5	Модель скрытого фактора	Последовательные	Краткосрочные	Первого порядка	Одиночные
6	Распределенное представление	В основном непоследовательные	Оба типа	В основном первого порядка	Коллективные
7	Рекуррентные нейронные сети	Последовательные	Долгосрочные	Высокого порядка	Одиночные
8	Многослойные сети персептронов	Непоследовательные	Оба типа	Первого порядка	Коллективные
9	Сверточные нейронные сети	В основном последовательные	Оба типа	В основном первого порядка	Коллективные
10	Графовые нейронные сети	Оба типа	Оба типа	Высокого порядка	Одиночные
11	Модели внимания	В основном непоследовательные	Оба типа	Первого порядка	В основном одиночные
12	Сети памяти	Непоследовательные	Оба типа	Первого порядка	Одиночные
13	Смешанные модели	Оба типа	Оба типа	Оба типа	Оба типа
14	Генеративные модели	Анализируемые зависимости в основном зависят от энкодера, используемого для кодирования входных данных генеративной модели.			
15	Обучение с подкреплением	Последовательные	Оба типа	Высокого порядка	Одиночные

A. Традиционные подходы

В основе традиционных подходов к реализации SBRS

для выявления зависимостей используются классические методы интеллектуального анализа данных или машинного обучения.

1) Анализ шаблонов/правил

В основном в SBRS используются два типа подходов на основе анализа шаблонов/правил:

- анализ частых шаблонов, заключающийся в выявлении внутри неупорядоченной сессии часто встречающихся шаблонов или ассоциативных правил для взаимодействий и генерировании последующих рекомендаций на основе данной информации;
- анализ шаблонов последовательностей, заключающийся в выявлении в упорядоченных сессиях шаблонных последовательностей в последовательности сессий или взаимодействий и генерировании последующих рекомендаций на основе данной информации.

Данный класс подходов может обрабатывать только сессии с одним типом доступных действий, поэтому каждое взаимодействие в сессии упрощается до элемента ($o = \langle v \rangle$).

2) Анализ частых шаблонов

Подход на основе анализа частых шаблонов включает три этапа:

Выявление шаблонов или ассоциативных правил.

Сопоставление контекста сессии с выявленными шаблонами (ассоциативными правилами).

Генерирование рекомендаций.

Пусть:

Пусть:

- $V = \{v_1, v_2, \dots, v_{|V|}\}$ - множество элементов;
- $S = \{s_1, s_2, \dots, s_{|S|}\}$ - множество сессий над V.

На первом этапе с помощью алгоритма интеллектуального поиска шаблонов выявляют множество часто встречающихся шаблонов $FP = \{p_1, p_2, \dots, p_{|FP|}\}$. В качестве алгоритма интеллектуального поиска шаблонов могут использоваться разные алгоритмы, например, FP-Tree [23].

На втором этапе контекст текущей сессии $\hat{s}$ (например, список элементов выбранных на текущий момент) сопоставляют с множеством выявленных шаблонов FP. В случае, если существует такой элемент $\hat{v} \in V \setminus \hat{s}$, что $\hat{s} \cup \{\hat{v}\} \in FP$, тогда элемент $\hat{v}$ добавляется в список кандидатов в рекомендации.

На третьем этапе элементы $\hat{v}$, для которых условная вероятность $P(\hat{v}|\hat{s})$ превышает заранее определенный доверительный порог, добавляются в список рекомендаций [24, 25].

Кроме рассмотренного базового алгоритма могут применяться различные его модификации, например, для учета значимости веб-страниц и рекомендации наиболее полезных для оценки веса каждой страницы можно использовать продолжительность ее просмотра, а затем включать такой вес в ассоциативные правила [26, 27].

SBRS построенные на основе анализа частых шаблонов применяются в традиционных сайтах электронной коммерции (например, для генерирования рекомендаций на основе корзины покупок), а также широко используются при рекомендации веб-страниц [28], музыкальных произведений [29] и т.п.

3) Анализ шаблонов последовательностей

Анализ шаблонов последовательностей включает в себя два подвида:

- анализ на уровне сессий и генерирование рекомендации по следующей сессии [30];
- анализ на уровне взаимодействий и генерирование рекомендаций по следующим взаимодействиям [31].

Далее рассмотрим алгоритм выполнения анализа шаблонов последовательностей на уровне сессий (алгоритм для уровня взаимодействий подробно рассмотрен в [31]).

Подход на основе анализа шаблонов последовательностей для уровня сессий включает три этапа:

- Выявление шаблонов последовательностей сессий.
- Сопоставление последовательности сессий пользователя с выявленными шаблонами.
- Генерирование рекомендаций.

Пусть $Q = \{q_1, q_2, \dots, q_{|Q|}\}$ – множество последовательностей, где $q = \{s_1, s_2, \dots, s_{|q|}\}$ является упорядоченной в хронологическом порядке последовательностью сессий одного пользователя.

На первом этапе с помощью алгоритма интеллектуального поиска шаблонов анализируют множества Q, результатом которого является множество шаблонов последовательностей сессий $SP = \{p_1, p_2, \dots, p_{|SP|}\}$, где p – шаблон последовательности сессий.

На втором этапе последовательность сессий $q_u = \{s_1, s_2, \dots, s_g\}$ текущего пользователя u сопоставляют с каждым шаблоном $p \in SP$. В случае, если последняя сессия пользователя s_g из q_u принадлежит p, т.е. $p = \{s_1, s_2, \dots, s_g, s_r \dots\}$, тогда p - это релевантный шаблон для генерирования рекомендации по следующей сессии пользователя u, а элементы сессии, следующей в p после s_g, то есть элементы сессии s_r, являются кандидатами в рекомендацию.

На третьем этапе для каждого кандидата в рекомендацию $\hat{v}$ вычисляется частота встречаемости, так называемая поддержка (*support*), которая равна сумме поддержек всех релевантных шаблонов:

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

В список рекомендаций пользователю u включаются элементы-кандидаты с наиболее высокими значениями поддержки.

Кроме рассмотренного базового алгоритма могут применяться различные его модификации, например, для формирования персонализированных рекомендаций каждому шаблону может присваиваться вес, основанный на его сходстве с прошлыми последовательностями целевого пользователя [32].

Другим вариантом является создание гибридной рекомендательной системы путем комбинирования анализа шаблонов последовательностей и совместной фильтрации для учета как динамических индивидуальных шаблонов, так и их общих предпочтений пользователей [33, 34].

SBRS реализованные на основе анализа шаблонов последовательностей в основном применяются для формирования рекомендаций продуктов на основе корзины покупок [30] и рекомендаций веб-страниц [31].

4) K-ближайших соседей

Подход на основе метода K-ближайших соседей (далее также - KNN) включает три этапа:

- Поиск в контексте сессии K-взаимодействий (сессий), которые наиболее похожи на текущее взаимодействие (сессию).
- Вычисление оценки сходства каждого взаимодействия - кандидата в рекомендацию с текущим взаимодействием и выбор наиболее релевантных.
- Генерирование рекомендаций.

Данный класс подходов может обрабатывать только сессии с одним типом доступных действий, поэтому каждое взаимодействие в сессии упрощается до элемента ($o = \langle v \rangle$).

В зависимости от того, для чего вычисляется оценка сходства, для элементов или сессий, различают метод K-ближайших соседей для элементов (item-KNN) и сессий (session-KNN).

a) K-ближайших соседей для элементов

На основе текущего контекста сессии в список рекомендаций включают K-элементов наиболее похожих на текущий элемент, с точки зрения совместного появления с ним в качестве следующего выбора в других сессиях.

Технически каждый элемент кодируется в двоичный вектор, в котором каждая из координат указывает, встречается ли элемент в конкретной сессии или нет (0 – не встречается; 1 – встречается).

Сходство между элементами вычисляется по их векторам с выбранной мерой сходства, в качестве которой, например, может использоваться косинусное сходство [37].

b) K-ближайших соседей для сессий

На основе контекста текущей сессии c, определяют множество $N(c)$, содержащее K-сессий наиболее похожих на текущую сессию, посредством вычисления сходства между текущей сессией и всеми другими сессиями.

Далее для каждого элемента-кандидата $\hat{v}$ вычисляют оценку сходства:

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

где:

- sim – функция оценки сходства сессий;
- $1_{s_{nb}}(\hat{v})$ – функция индикатор, возвращая значение «1», если $\hat{v}$ встречается в s_{nb}, и «0» - если нет.

В отличие от метода для элементов, в методе для сессий рассматривается весь контекст сессии, а не только текущий элемент, таким образом он учитывает больше информации и позволяет генерировать более точные рекомендации.

Кроме рассмотренного базового алгоритма могут применяться его различные модификации, например, гибридный подход, который комбинируется метод K-ближайших соседей для сессий и алгоритм GRU4Rec (на основе рекуррентной нейронной сети) [38].

5) Цепь Маркова

Данный подход использует цепи Маркова для моделирования переходов между взаимодействиями внутри или между сессиями, чтобы предсказать следующее вероятное взаимодействие(-я) или сессию с учетом контекста текущей сессии [39].

В зависимости от того, на основе чего вычисляются вероятности перехода, разделяют подходы на основе:

простой цепи Маркова, в которых вероятности перехода вычисляются на основе явных наблюдений;

скрытого представления Маркова, в которых вероятности перехода вычисляются на основе скрытого пространства.

a) Простая цепь Маркова

Подход на основе простой цепи Маркова обычно включает четыре этапа [40]:

1. Вычисление вероятностей перехода между взаимодействиями;
2. Прогнозирование путей перехода между взаимодействиями;
3. Сопоставление контекста сессии и прогнозируемых путей;
4. Генерирование рекомендаций на основе результатов сравнения.

В большинстве случаев взаимодействия упрощаются до элементов.

Определим модель цепи Маркова как набор кортежей $\{ST, \boldsymbol{P}_t, P_0\}$, где:

- ST – состояние пространства, включающее все различные взаимодействия;
- $\boldsymbol{P}_t$ – матрица (m на m) вероятностей перехода за один шаг между m различными взаимодействиями;
- P_0 – начальная вероятность каждого состояния в ST.

На первом этапе вычисляются вероятности перехода между взаимодействиями за один шаг (первого порядка), например, вероятность перехода от взаимодействия o_i к взаимодействию o_j вычисляется как:

$$P_t(i,j) = P(o_i \to o_j) = \frac{freq(o_i \to o_j)}{\sum_{o_t} freq(o_i \to o_t)}$$

На втором этапе производится прогнозирование путей перехода между взаимодействиями посредством оценки их вероятностей с использованием цепи Маркова первого порядка, например, для перехода $\{o_1 \to o_2 \to o_3\}$:

$$P(o_1 \to o_2 \to o_3) = P(o_1) * P(o_2|o_1) * P(o_3|o_2)$$

На третьем этапе последовательность произошедших взаимодействий из контекста текущей сессии сравнивается с множеством прогнозируемых путей, из которых в качестве релевантных путей выбираются пути с наиболее высокой вероятностью.

На четвертом этапе взаимодействия, расположенные в релевантном пути после взаимодействия, соответствующего последнему взаимодействию текущей сессии, включаются в список рекомендаций.

Кроме рассмотренного базового алгоритма могут применяться его различные модификации, например, в работе [41] комбинируют модели Маркова первого и второго порядка для формирования более точных рекомендаций веб-страниц, в работе [56] разработали скрытую модель Маркова на основе вероятностной модели для рекомендации следующего элемента, в работе [95] матрицу вероятностей перехода разложили на множители для оценки ненаблюдаемых переходов между взаимодействиями.

b) Скрытое представление Маркова

В отличие от простой цепи Маркова, в которой вероятности перехода вычисляются на основе явных наблюдений, в скрытом представлении Маркова сначала строится представление цепи Маркова в евклидовом пространстве, а затем вычисляются вероятности перехода между взаимодействиями на основе евклидова расстояния между ними [42].

Таким образом, можно получить ненаблюдаемые переходы и решить проблему разреженности данных в условиях ограниченности наблюдаемых данных.

В данном подходе каждое взаимодействие o представляется в виде вектора $\boldsymbol{o}$ в d-мерном Евклидовом пространстве, и предполагается, что вероятность перехода $P(o_i \to o_j)$ убывает с ростом евклидового расстояния $||\boldsymbol{o}_i - \boldsymbol{o}_j||_2$ между взаимодействиями o_i и o_j.

Таким образом, вероятность перехода по пути:

$$pa = \{o_1 \to o_2 \to, \dots, \to o_{|pa|}\}$$

может быть определена на основе модели Маркова:

$$P(\{o_1 \to o_2 \to, \dots, \to o_{|pa|}\}) = \prod_{i=2}^{|pa|} P(o_{i-1} \to o_i) = \prod_{i=2}^{|pa|} \frac{e^{-||o_i - o_{i-1}||_2^2}}{\sum_{o_t} e^{-||o_t - o_{i-1}||_2^2}}$$

Дальнейшие шаги по генерированию рекомендаций аналогичны описанным подходе на основе простой цепи Маркова.

Кроме рассмотренного базового алгоритма могут применяться его различные модификации, например, в работе [43] для генерирования персонализированных рекомендаций предложена модель *Personalized Markov Embedding (PME)*, в которой в евклидовом пространстве представляются и элементы, и пользователи, и расстояния «пользователь-элемент» и «элемент-элемент» отражают соответствующие попарные отношения.

6) Генеративная вероятностная модель

Подход на основе генеративной вероятностной модели обычно включают четыре этапа:

Определение скрытой таксономии элементов в сессиях (например, темы или жанры песен);

Анализ переходов между выявленными таксонами внутри или между сессиями;

Прогноз следующего таксона на основе контекста текущей сессии;

Генерирование рекомендаций по следующим элементам на основе прогноза по следующему таксону.

Более подробно с исследованиями данной модели можно ознакомиться в работе [44], в которой генерирование рекомендаций производится на основе скрытых шаблонов последовательностей жанров музыки, и в работе [45] в которой список воспроизведения музыки генерируется на основе статистической модели сессий прослушивания музыки.

7) Сравнение традиционных подходов

Сравнение традиционных подходов, их основные характеристики, плюсы и минусы представлены в таб. 8.

Таблица 8. Сравнение традиционных подходов

№	Наименование	Для каких сессий применяются	Плюсы	Минусы
1	Анализ шаблонов/ правил	- простые, сбалансированные и плотные по данным сессии; - упорядоченные или неупорядоченные сессии.	- интуитивно понятный; - простой и эффективный для сессий с простыми зависимостями.	- потеря информации; - не может применяться для обработки сложных (например, несбалансированных или разреженных) данных.
2	*K*-ближайших соседей	- простые сессии; - упорядоченные или неупорядоченные сессии.	- интуитивно понятный; - простой и эффективный; - высокая	- потеря информации; - сложно выбрать значение *K*; - ограниченные возможности для

			производительность*.	сложных сессий (например, сессии с шумом).
3	Цепи Маркова	- короткие и упорядоченные сессии с краткосрочными и невысокого порядка зависимостями.	- хорошо моделирует краткосрочные и не высокого порядка последовательные зависимости.	- обычно игнорируются долгосрочные и высокого порядка зависимости; - слишком сильное предположение о жестком порядке зависимостей.
4	Генеративная вероятностная модель	- упорядоченные сессии с зависимостями высокого порядка.	- хорошо моделирует зависимости высокого порядка и коллективные зависимости.	- относительно высокая стоимость вычислений.

*В работе [46] было проведено эмпирическое сравнение точности прогнозирования для первых трех классов рассмотренных традиционных подходов (анализ шаблонов/правил, K-ближайших соседей, цепи Маркова), результат которого показал, что подходы, основанные на K-ближайших соседей (особенно session-KNN) достигают превосходной производительности.

В Подходы на основе скрытого представления

В основе данных подходов лежит создание для каждого взаимодействия в сессии скрытого представления низкой размерности с использованием неглубоких моделей. Полученные представления кодируют зависимости между этими взаимодействиями и в дальнейшем используются для генерирования рекомендаций.

В зависимости от используемых методов, подходы на основе скрытого представления условно разделяют на:

- модели скрытого фактора;
- распределенного представления.

1) Модель скрытого фактора

Подход на основе модели скрытого фактора обычно включает три этапа:

- Факторизации матрицы наблюдаемых переходов между взаимодействиями (элементами) на их скрытые представления;
- Оценка ненаблюдаемых переходов на основе полученных скрытых представлений взаимодействий (элементов);
- Генерирование рекомендаций.

Например, на основе наблюдаемых данных может быть построен переходный тензор:

$$\mathcal{B}^{|U|\times|O|\times|O|}$$

где каждая запись $b_{k,i,j}$ представляет собой вероятность перехода пользователя u_k от взаимодействия o_i к o_j. Затем с помощью общей модели линейной факторизации, например, декомпозиции Такера, производится факторизации матрицы $\mathcal{B}$:

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

где:

– Co – основной тензор;

– $\boldsymbol{U}$ – матрица скрытых представлений пользователей;

– $\boldsymbol{O}_i$ – матрица скрытых представлений предшествующих взаимодействий;

– $\boldsymbol{O}_j$ - матрица скрытых представлений следующих взаимодействий.

Для снижения отрицательного эффекта разреженных переходов, наблюдаемых в $\mathcal{B}$, используют частный случай канонического разложения [47] для преобразования $\hat{\mathcal{B}} = Co \times \boldsymbol{U} \times \boldsymbol{O}_i \times \boldsymbol{O}_j$ в модель попарных взаимодействий [22]:

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

где:

– $\boldsymbol{u}_k$ – вектор скрытого представления пользователя u_k;

– $\boldsymbol{o}_i$ – вектор скрытого представления предшествующего взаимодействия o_i;

– $\boldsymbol{o}_j$ – вектор скрытого представления следующего взаимодействия o_j;

В данном случае взаимодействия обычно упрощаются до элементов.

Кроме рассмотренного базового алгоритма модели скрытого фактора (также известной, как *Factorized Personalized Markov Chain* (FPMC) модель), могут применяться различные его модификации, например, в работе [49] были добавлены ограничения на количество поездок пользователей в конкретный регион, чтобы сделать модель более похожей на реальный туризм и генерировать рекомендации по достопримечательностям. В работе [50] предложена модель совместной факторизации CoFactor для совместной декомпозиции матрицы взаимодействий пользователей с элементами «пользователи-элементы» и матрицы совместного появления элементов «элементы/элементы» с общими скрытыми факторами элементов для выявления, как индивидуальных предпочтений пользователей, так и шаблонов переходов между элементами.

2) Распределенное представление

В основе данного подхода лежит использование неглубокой нейронной сети (shallow neural network) для создания отображения каждого взаимодействия в скрытое пространство невысокой размерности.

Взаимодействия обычно упрощают до элементов, иногда с включением идентификатора пользователя. В

большинстве случаев используемая структура сети подобна модели Skip-gram [51] или CBOW [52] из класса моделей, используемых для обработки естественного языка.

В результате внутри- или межсессионные зависимости кодируются в распределенные представления, которые в дальнейшем используются для генерирования рекомендаций.

Нейронная сеть строит представление пользователя u_k и элемента v_i в скрытый вектор распределения используя логистическую функцию $\delta(\cdot)$ для нелинейного преобразования [53]:

$$\boldsymbol{u}_k = \delta\left(\boldsymbol{W}^u_{:,k}\right),$$
$$\boldsymbol{v}_i = \delta\left(\boldsymbol{W}^v_{:,i}\right),$$

где:

– $\boldsymbol{W}^u \in \mathbb{R}^{d\times|U|}$ – матрица весов для формирования представления пользователей;

– $\boldsymbol{W}^v \in \mathbb{R}^{d\times|V|}$ – матрица весов для формирования представления элементов;

– k – номер столбца матрицы $\boldsymbol{W}^u$ соответствующий пользователю u_k.

Кроме рассмотренного базового алгоритма могут применяться различные его модификации, например, в работе [54] сконструирована неглубокая нейронная сеть для встраивания одновременно идентификатора и признаков каждого элемента в комбинированное представление элемента для решения проблем «холодного старта» для элементов.

3) Сравнение подходов на основе скрытого представления

Сравнение подходов на основе скрытого представления, их основные характеристики, плюсы и минусы представлены в таб. 9.

Таблица 9. Сравнение подходов на основе скрытого представления.

№	Наименование	Для каких сессий применяются	Плюсы	Минусы
1	Модель скрытого фактора	- плотные по данным сессии; - упорядоченные сессии.	- относительно простой и эффективный.	- неэффективный на разреженных данных; - не может выявлять зависимости высокого порядка; - не может выявлять долгосрочные зависимости.
2	Распределенное представление	- неупорядоченные сессии.	- простой и эффективный; - мощные возможности кодирования.	- сложно моделировать упорядоченные или разнородные сессии (например, шумные сессии).

С Подходы на основе глубоких нейронных сетей

Подходы на основе глубоких нейронных сетей используются широкие возможности глубоких нейронных сетей для моделирования сложных внутри- и межсессионных зависимостей.

В зависимости от используемой архитектуры нейронной сети подходы на основе глубоких нейронных сетей условно разделяют на использующие:

- базовый тип архитектуры;
- продвинутые модели и алгоритмы.

1) Базовые типы архитектуры нейронных сетей

Используемые в SBRS базовые типы архитектуры нейронной сети включают:

- рекуррентную нейронную сеть (Recurrent Neural Networks, RNN);
- многослойную сеть персептронов (Multi-Layer Perceptron networks, MLP);
- сверточную нейронную сеть (Convolutional Neural Networks, CNN);
- графовую нейронную сеть (Graph Neural Networks, GNN).

2) Рекуррентная нейронная сеть

В SBRS на основе рекуррентной нейронной сети контекст упорядоченной сессии моделируется в виде последовательности взаимодействий внутри контекста.

Таким образом, в качестве представления контекста используется его модель в виде последнего скрытого состояния (hidden state) сети.

Далее данная модель используется в качестве входных данных SBRS для прогнозирования следующего взаимодействия и генерирования рекомендаций.

Для SBRS, анализирующих межсессионные зависимости, последовательность исторических сеансов моделируется аналогичным образом.

Рассмотрим более подробно SBRS на основе рекуррентной нейронной сети на примере алгоритма GRU4Rec, построенного на основе управляемых рекуррентных блоков (Gated Recurrent Units, GRU) [17].

В алгоритме GRU4Rec рекуррентная нейронная сеть используется для моделирования контекста сессии, состоящего из последовательности взаимодействий. Сначала представление $\boldsymbol{o}_t$ (соответствующего *t*-му взаимодействию o_t) принимается в качестве входных данных для *t*-го шага рекуррентной нейронной сети. Затем *GRU* используется для обновления скрытого состояния $\boldsymbol{h}_t$ на шаге t посредством совместной обработки информации, как из последнего скрытого состояния $\boldsymbol{h}_{t-1}$, так и из текущего состояния-кандидата $\widetilde{\boldsymbol{h}_t}$ с помощью вентиля обновления z_t:

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

где z_t и $\widetilde{\boldsymbol{h}_t}$ вычисляются с помощью уравнений:

$$z_t = \sigma(\boldsymbol{W}_z \boldsymbol{o}_t + \boldsymbol{X}_z \boldsymbol{h}_{t-1})$$
$$\widetilde{h_t} = tanh(\boldsymbol{W}_h \boldsymbol{o}_t + \boldsymbol{X}_h(\boldsymbol{r}_t \odot \boldsymbol{h}_{t-1}))$$

где $\odot$ обозначает произведение Адамара (покомпонентное произведение матриц), а $\boldsymbol{r}_t$ - вентиль сброса, который рассчитывается по формуле:

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

где σ - функция активации, в качестве которой может быть использована сигмовидная функция, а $\boldsymbol{W}$ и $\boldsymbol{X}$ - соответствующие матрицы весов.

Таким образом, контекст сеанса c, состоящий из $|c|$ взаимодействий, может быть смоделирован с помощью рекуррентной нейронной сети с $|c|$ блоками.

Наконец, скрытое состояние $\boldsymbol{h}_{|c|}$ с последнего временного шага используется как представление контекста $\boldsymbol{e}_{|c|}$ для прогнозирования следующего взаимодействия [17].

Кроме рассмотренного базового алгоритма GRU4Rec, существуют различные его модификации, например, в работе [56] авторы с помощью предварительной обработки последовательности увеличили количество исходных данных для улучшения процесса обучения сети и применили метод «droupout» для решения проблемы переобучения.

3) Многослойная сеть персептронов

В SBRS на основе многослойной сети персептронов нейронная сеть применяется для изучения оптимальной соединения различных представлений для формирования составного представления контекста сессии.

В отличие от рекуррентной нейронной сети многослойная сеть персептронов в основном подходит для неупорядоченных сессий в связи с отсутствием возможности моделировать последовательность данных.

В частности, в работе [57] многослойная сеть персептронов используется для соединения представлений различных частей контекста сессии c для получения унифицированного составного представления $\boldsymbol{e}_c$:

$$\boldsymbol{e}_c = \sigma\left(\boldsymbol{W}_c \boldsymbol{e}_{c_c} + \boldsymbol{W}_v \boldsymbol{e}_{c_v}\right)$$

где:

– $\boldsymbol{e}_{c_c}$ - представление части контекста сессии, содержащего действия «click»;

– $\boldsymbol{e}_{c_v}$ – представление части контекста сессии, содержащего действия «view»;

– W_c и W_v – матрицы весов для полносвязного соединения каждого из представлений со скрытым слоем многослойной сети персептронов.

В работе [58] авторы применили многослойную сеть персептронов для изучения оптимальной комбинации различных факторов, таких как популярность товара, скидка и другие, для получения составного признака.

В работе [59] авторы использовали слой многослойную сеть персептронов для объединения как статических долгосрочных, так и временных краткосрочных предпочтений пользователя для формирования более точных рекомендаций по следующему элементу.

4) Сверточная нейронная сеть

В SBRS на основе сверточной нейронной сети сначала применяют операции фильтрации (filtering) и объединения (pooling) для получения представления контекста сессии, которое в дальнейшем используют для генерирования рекомендаций [60].

Например, контекст сессии c содержит $|c|$ взаимодействий, матрица представления (*embedding matrix*) $E \in \mathbb{R}^{d \times |c|}$ контекста c может быть построена путем сопоставления каждому взаимодействию в c соответствующего d-мерного скрытого вектора, а затем объединением всех векторов в матрицу.

Далее, в горизонтальном сверточном слое m-е значение свертки a_m^x получается путем перемещения x-го фильтра $\boldsymbol{F}^x$ сверху вниз по $\boldsymbol{E}$ для взаимодействия с его горизонтальными размерами:

$$a_m^x = \phi_a(\boldsymbol{E}_{m:m+h-1} \odot \boldsymbol{F}^x)$$

где ϕ_a - функция активации сверточного слоя.

Затем конечный результат $\boldsymbol{e}_c \in \mathbb{R}^z$ из фильтров z получается путем выполнения операции выбора максимального значения (*max pooling*) на результатах свертки $\boldsymbol{a}^x = [a_1^x, a_2^x, \dots, a_{|c|-h+1}^x]$ для выявления наиболее существенных признаков в контексте сессии:

$$\boldsymbol{e}_c = \max\{\max(\boldsymbol{a}^1), \max(\boldsymbol{a}^2), \dots, \max(\boldsymbol{a}^z)\}$$

Наконец, $\boldsymbol{e}_c$ рассматривается как представление контекста сессии c и используется для генерирования последующих рекомендаций [61].

Сверточные нейронные сети являются хорошим выбором для SBRS по следующим причинам:

1) они ослабляют предположение о жестком порядке взаимодействия внутри сессий, что делает модель более надежной;
2) они обладают высокими возможностями в изучении локальных признаков определенной области и взаимосвязей между различными областями в сессии, что позволяет эффективно учитывать коллективные зависимости в данных сессии.

В работе [62] авторы используют для SBRS 3D-модель сверточной нейронной сети, которая совместно моделирует последовательность шаблонов в данных *click*-сессии и характеристик элементов из их признаков. В работе [63] модель сверточной нейронной сети применяется для аккумулирования долгосрочных предпочтений пользователей для формирования персонализированных рекомендаций.

5) Графовые нейронные сети

В SBRS на основе графой нейронной сети сначала набор данных, содержащий несколько сессий, переносится на граф G путем представления каждой сессии в виде соответствующей цепи на графе.

Каждое взаимодействие (o) в сессии представляется узлом (n) в соответствующей цепи, а каждая пары смежных взаимодействий в сессии соединяется ребром (e).

Далее построенный граф импортируется в графовую нейронную сеть для получения представления для каждого узла (взаимодействия) путем энкодинга

сложных переходов по графу.

Затем полученные представление передаются в модуль прогнозирования для генерирования рекомендаций.

В зависимости от используемой архитектуры графовой нейронной сети различают:

- нейронную сеть с замкнутым графом (Gated Graph Neural Networks, GGNN);
- графовую сверточную сеть (Graph Convolutional Networks, GCN);
- графовую сеть внимания (Graph Attention networks, GAT).

a) Нейронная сеть с замкнутым графом

В SBRS на основе нейронной сети с замкнутым графом сначала на основе всех исторических упорядоченных сессий строится ориентированный граф, в котором направление каждого ребра указывает порядок между соседними взаимодействиями внутри сессии.

Далее подграф каждой сессии последовательно обрабатывается сетью для получения представления $\boldsymbol{n_i}$ узла n_i, а именно представления соответствующего взаимодействия o_i.

Наконец, после обработки всех графов сессий получаются представления всех взаимодействий, которые затем используются для построения представления контекста сессии.

В частности, в нейронной сети с замкнутым графом для получения представления каждого узла графа сессии используются управляемые рекуррентные блоки (*GRU*) посредством рекуррентного обновления представления.

Представление, также называемое скрытым состоянием, $\boldsymbol{h_i^t}$ узла n_i на шаге t обновляется посредством использования его предыдущего скрытого состояния и представлений его соседних узлов $\boldsymbol{h_i^{(t-1)}}$ и $\boldsymbol{h_j^{(t-1)}}$:

$$\boldsymbol{h_i^t} = GRU(\boldsymbol{h_i^{(t-1)}}, \sum_{n_j \in N(n_i)} h_j^{(t-1)}, \boldsymbol{A})$$

где $N(n_i)$ - множество соседних узлов узла n_i в графе сессии, а $\boldsymbol{A}$ - матрица смежности, построенная на основе графа сессии.

После нескольких итераций до достижения устойчивого равновесия скрытое состояние на последнем шаге для узла n_i принимается за его представление $\boldsymbol{n_i}$.

Кроме рассмотренного существуют и другие подходы к генерированию рекомендаций на основе сессии с использованием графовой нейронной сети, такие как *Graph Contextualized Self-Attention Network* (*GC-SAN*) [64], в котором используется как графовая нейронная сеть, так и механизм внимания для изучения локальных и отдаленных зависимостей, а также *Target Attentive Graph Neural Network (TAGNN)* [65], в котором сначала изучается представления элементов с помощью графовой нейронной сети, а затем внимательно активируются различные интересы пользователей в отношении различных целевых элементов.

b) Графовые сверточные сети

В SBRS на основе графовой сверточной сети используется операция $pooling$ для объединения информации от узлов n_j, являющихся в графе сессии соседними с узлом n_i, для обновления скрытого состояния n_i:

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

где $N(n_i)$ - множество узлов, являющихся соседними с узлом узла n_i.

В зависимости от конкретного сценария могут использоваться различные $pooling$-операции, такие как $mean\ pooling$ или $max\ pooling$.

Далее результат объединения информации от соседних узлов используется для итеративного обновления скрытого состояния узла n_i [66]:

$$\boldsymbol{h}_i^t = \boldsymbol{h}_i^{(t-1)} + \widehat{\boldsymbol{h}}_i^t$$

После нескольких итераций до достижения устойчивого равновесия скрытое состояние на последнем шаге для узла n_i принимается за его представление $\boldsymbol{n_i}$.

c) Графовая сеть внимания

В SBRS на основе графовой сети внимания используется механизм внимания ($attention$) для объединения информации от узлов n_j, являющихся в графе сессии соседними с узлом n_i, для обновления скрытого состояния n_i на каждом слое внимания [67]:

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

где $\boldsymbol{h}_i^t$ - скрытое состояние узла n_i на t-м слое внимания.

В приведенной формуле $attention$ представляет собой обобщенный механизм внимания, который может быть задан различными механизмами внимания, такими как $self\ attention$, $multi\text{-}head\ attention$ и т.д.

Механизм $attention$ можно разделить на два этапа:

1. Вычисление весов важности для каждого соседнего узла;
2. Агрегирование скрытых состояний соседних узлов в соответствии с их весами важности.

Когда прямое распространение множества слоев внимания завершено, скрытое состояние каждого узла n_i на последнем слое в графе сессии принимается за его представление $\boldsymbol{n_i}$.

Кроме рассмотренного существуют и другие подходы к генерированию рекомендаций с использованием графовой сети внимания, такие как *Full Graph Neural Network (FGNN)*, в котором изучается внутренний порядок шаблонов перехода между элементами в сессиях с помощью сети с внимая с несколькими взвешенными графами [67], или другая *Full Graph Neural Network*, основанная на широко связанном графе сессий для внимательного использования информации как внутри, так и между сессиями [69].

D Продвинутые модели и алгоритмы

Продвинутые модели и алгоритмы, используемые в SBRS, включают:

- модели внимания (attention models);
- сети памяти (memory networks);
- модели смеси (mixture models);
- генеративные модели (generative models);
- обучение с подкреплением (reinforcement learning).

Для создание более совершенных SBRS данные модели или алгоритмы обычно сочетаются с некоторыми базовыми подходами, такими как распределенное представление (distributed representation) или рекуррентная нейронная сеть (recurrent neural network).

6) Модели внимания

В SBRS на основе модели внимании используется механизм внимания [70] для дискриминационного учета взаимодействий и/или сессий при создании информативного представления контекста сессии.

Механизм внимания позволяет SBRS усилить влияние более существенных и уменьшить влияние несущественных взаимодействий (сессий).

Модель внимания обычно включает два этапа:

- Вычисление весов важности для каждого соседнего взаимодействия;
- Агрегирование представлений всех соседних взаимодействий в соответствии с их весами важности.

Рассмотрим модель внимания на примере, когда контекст сессии включает только известную часть текущей сессии. Представления контекстов, включающих исторические сессии, строятся аналогичным образом.

На первом этапе на основе представления $\boldsymbol{o}_i$ взаимодействия o_i в контексте сессии c для каждого соседнего взаимодействия o_{tg} модель внимания рассчитывает вес $\beta_{tg,i}$ для взаимодействия o_i для индикации его важности (релевантности с взаимодействием o_{tg}). Вес $\beta_{tg,i}$ обычно рассчитывается с помощью функции softmax [21]:

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

где $e(\boldsymbol{o}_i)$ - функция полезности, которая может быть задана как внутреннее произведение между обучаемым вектором весов $\boldsymbol{w}$ и представлением $\boldsymbol{o}_i$.

Иногда $\boldsymbol{o}_{tg}$ также используется в составе входных данных функции полезности для того, чтобы сделать полученный вес более чувствительным к целевому взаимодействию o_{tg}.

На втором этапе представления всех взаимодействий в контексте сеанса c объединяются с изученными весами для построения представления контекста $\boldsymbol{e}_c$:

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

где $aggregate$ - функция агрегирования, в качестве которой часто используется взвешенная сумма.

Затем представление контекста передается в модуль прогнозирования для генерирования рекомендаций.

Кроме рассмотренного существуют и другие подходы к генерированию рекомендаций на основе сессии с использованием модели внимания, например, в работах [72, 73] предложена иерархическая модель внимания для интеграции как исторических сессий, так и текущей сессии пользователя, для выявления его долгосрочных и краткосрочных предпочтений, и генерации более точных рекомендаций.

b) Сети памяти

В SBRS на основе сети памяти используется сеть памяти для выявления зависимости между любым взаимодействием в контексте сессии и следующим взаимодействием посредством введения внешней матрицы памяти.

Такая матрица хранит и обновляет информацию о каждом взаимодействии в контексте сессии для сохранения наиболее актуальной и важной информации для генерации рекомендаций.

SBRS на основе сети памяти в основном состоит из двух основных компонентов [74]:

- матрицы памяти, которая хранит представление взаимодействий в контексте сессии c;
- контроллера, который выполняет операции (включая чтение и запись) на матрице памяти.

Предположим, что M^c матрица памяти для хранения представлений последних взаимодействий в контексте сессии c, в которой каждый столбец соответствует представлению для одного взаимодействия.

После того как взаимодействие o_i происходит в сессии и добавляется в контекст сессии c, в матрице M^c обновляется информация о последних взаимодействиях путем записи в нее представления o_i:

M^c←write(M^c,o_i)

где $write$ - операция записи, в качестве которой может использоваться один из различных процессов записи, например, Least Recently Used Access (LRUA) [75].

Во время прогнозирования информация считывается из матрицы памяти для построения представления контекста сессии e_c:

e_c=read(M^c,o_tg)

где:

o_tg – представление кандидата-взаимодействия o_tg (в процессе чтения учитывается возможность считывания информации, более релевантной к o_tg);

read - операция чтения, которая может быть задана в различных формах, в том числе в форме рассмотренного в разделе 4.3.2.1 механизма внимания.

Кроме рассмотренного существуют и другие подходы к генерированию рекомендаций на основе сессии с использованием сети памяти, например, в работе [76] предложены два параллельных модуля памяти: энкодер внутренней памяти (Inner Memory Encoder) и энкодер внешней памяти (Outer Memory Encoder) для

моделирования текущей и соседних сессий соответственно для создания более информативного представления контекста сессии.

c) Смешанная модель

SBRS на основе смешанной модели использует составную модель, включающую несколько подмоделей, что позволяет использовать преимущества каждой из подмоделей при комплексном моделировании различных сложных зависимостей.

Обычно каждая из подмоделей имеет преимущества перед другими подмоделями при моделировании определенного типа зависимостей (например, зависимостей низкого или высокого порядка).

Работа SBRS на основе смешанной модели включает в себя два основных этапа:

- Изучение различных типов зависимостей с использованием разных подмоделей;
- Аккуратная интеграция изученных зависимостей для получения точных рекомендаций.

Типовые работы по SBRS на основе смешанной модели включают Neural Multi-Temporal range Mixture Model (M3) [77], в которой комбинируются различные типы энкодеров для выявления краткосрочных и долгосрочных зависимостей, и Mixture-channel Purpose Routing Networks (MCPRN) [71], в которой используются несколько рекуррентных сетей для моделирования зависимостей внутри сессии, в соответствии с различными целями покупок пользователя.

d) Генеративные модели

SBRS на основе генеративной модели формируют рекомендации путем генерации следующего взаимодействия(-й) или следующей сессии посредством скрупулезно разработанной стратегии генерации.

Генеративная модель хорошо подходит для моделирования поведения пользователей при совершении онлайн покупок в интернете, где товары при формировании корзины покупок часто подбираются шаг за шагом [36].

В данной модели на основе контекста сессии c генерируется рекомендуемый список взаимодействий (элементов) l:

l=*generate(c)*

где $generate$ - процесс генерации, который может быть задан в различных формах, например, в виде вероятностной генеративной модели [68].

Типовые работы по SBRS на основе генеративной модели включают NextItNet [48], в которой разработана вероятностная генеративная модель для генерации распределения вероятности элементов-кандидатов; Intention2Basket [36], в которой разработан генератор на основе функции полезности для генерации сессии-кандидата с максимальной полезностью.

e) Обучение с подкреплением

SBRS на основе обучения с подкреплением моделируют взаимодействие между пользователем и системой рекомендаций в сессии подобно Марковскому процессу принятия решений (Markov Decision Process).

Например, сначала система рекомендует пользователю товар, который предоставляет системе свою обратную связь (отзыв) о товаре, а затем система рекомендует пользователю следующий товар в соответствии с его отзывом, чтобы лучше соответствовать его предпочтениям.

SBRS на основе обучения с подкреплением нацелена на изучение оптимальных стратегий рекомендаций посредством метода проб и ошибок, и получения обратной связи от пользователей [35].

Таким образом, SBRS может постоянно обновлять свои стратегии во время взаимодействия с пользователями до достижения оптимальной стратегии, которая наилучшим образом соответствует их динамическим предпочтениям.

Кроме того, при оптимизации стратегий учитывается ожидаемые долгосрочные совокупные предпочтения пользователей.

В соответствии с работой [35] для формализации задачи SBRS на основе обучения с подкреплением определим следующие пять ключевых понятий:

1) пространство состояний Sa, где состояние $sa_t = \{sa_t^1, \dots, sa_t^{m'}\} \in Sa$ определяется как предыдущие m' элементов с которыми пользователь взаимодействовал до момента времени t;

2) пространство действий Ac, где действие $ac_t = \{ac_t^1, \dots, ac_t^{n'}\} \in Ac$ представляет собой сформированный на основе текущего состояния sa_t список элементов рекомендуемых пользователю в момент времени t, а n' количество элементов рекомендуемых пользователю в каждый момент времени;

3) вознаграждение Re - после того, как система выполнит действие ac_t в состоянии sa_t, она сразу получает вознаграждение Re_t в соответствии с обратной связью от пользователя;

4) вероятность перехода $Tp(sa_{t+1}|sa_t, ac_t)$ - определяет вероятность перехода из состояния sa_t в состояние sa_{t+1} при совершении действия ac_t;

5) коэффициент дисконтирования $df \in [0,1]$ - определяет коэффициент дисконтирования при вычислении размера будущего вознаграждения в текущий момент времени.

Таким образом, задача SBRS состоит в том, чтобы найти политику рекомендаций $\pi: Sa \rightarrow Ac$ которая максимизирует совокупное вознаграждение системы с учетом Sa, Ac, Re, Tp, df.

В работе SBRS на основе обучения с подкреплением выделяют три основных этапа:

1) вычисление весовых параметров для конкретного состояния путем сопоставления состоянию sa_t матрицы весов $\boldsymbol{W}_t$:

$$f_t: sa_t \rightarrow \boldsymbol{W}_t$$

2) вычисление оценки для каждого элемента-кандидата v_i с помощью функции оценки f_s и последующий выбор для рекомендаций элементов с наивысшим значением оценки:

$$score(v_i) = f_s(v_i, \boldsymbol{W}_t)$$

3) вычисление функции значения-действия $E(sa_t, ac_t)$ для потенциального действия ac_t, которая представляет собой суждение о том, будет ли ac_t соответствовать текущему состоянию sa_t в случае рекомендации выбранного элемента или нет [35]. Затем, в соответствии полученным значением функции $E(sa_t, ac_t)$ система обновляет свои параметры в направлении повышения производительности для генерации правильных действий на следующих итерациях.

Обычно в качестве функции значения-действия $E^*(sa_t, ac_t)$ используется максимальная ожидаемая доходность, достижимая при оптимальной политике [55]:

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

Впоследствии стратегии рекомендаций оптимизируются за счет минимизации ошибки между значением рассчитанных действий и значение опробованных действий.

Типовые работы по SBRS на основе обучения с подкреплением включают *List-wise recommendation framework based on deep reinforcement learning (LIRD)* [35], в которой изучаются стратегии рекомендаций для списка элементов, а также аналогичную работу под названием *Page-wise recommendation framework based on deep reinforcement learning (DeepPage)* для рекомендаций двумерных страниц элементов [55].

D. Сравнение подходов на основе глубоких нейронных сетей

Сравнение подходов на основе глубоких нейронных сетей, их основные характеристики, плюсы и минусы представлены в таб. 10 [6].

Таблица 10. Сравнение подходов на основе глубоких нейронных сетей

№	Наименование	Для каких сессий применяются	Плюсы	Минусы
Базовые типы архитектур				
1	Рекуррентная нейронная сеть	- длительные и строго упорядоченные сессии.	- моделирование долгосрочных последовательных зависимостей; - моделирование последовательных зависимостей высоких порядков.	- необходимость строго порядка в данных сессии.
2	Многослойная сеть персептронов	- неупорядоченные сессии; - сессии с множеством учитываемых факторов (например, статические и динамические признаки), которые комбинируются в единый составной признак.	- простая структура; - возможность изучения комбинаций различных факторов.	- невозможно моделировать сложные сессии (например, упорядоченные, разнородные сессии).
3	Сверточная нейронная сеть	- сессии с гибким порядком; - неоднородные сессии; - зашумленные сессии.	- нет необходимости жесткого порядка; - выявление коллективных зависимостей.	- относительно высокая сложность.
4	Графовая нейронная сеть	- сложные сессии со сложными переходами (например, повторяющиеся взаимодействия).	- возможность моделирования сложных переходов между взаимодействиями.	- сложный и затратный подход.
Продвинутые модели и алгоритмы				
5	Модель внимания	- разнородные сессии; - зашумленные сессии; - длительные сессии.	- возможность определять важную информацию.	- невозможно выявлять последовательную информацию.
6	Сеть памяти	- длительные сессии; - инкрементные сессии; - зашумленные сессии.	- хранение и динамическое обновление последней информации.	- невозможно выявлять последовательную информацию.
7	Смешанная модель	- разнородные сессии; - зашумленные сессии.	- возможность моделирования различных типов зависимостей (например, долгосрочных и краткосрочных).	- относительной сложный и затратный подход.
8	Генеративная модель	- динамические сессии; - инкрементные сессии.	- хорошо отражает реальное поведение пользователей при совершении онлайн покупок.	- сложный подход.
9	Обучение с подкреплением	- динамические сессии; - инкрементные сессии.	- интерактивный процесс, учитывающий будущие последствия действий.	- сложно имитировать интерактивное окружение.

V Программные реализации SBRS

A. Программные реализации алгоритмов

Исходный код большинства алгоритмов для SBRS находится в открытом доступе. В таб. 11 представлена консолидированная информация об алгоритмах для SBRS с открытым исходным кодом [6]:

Таблица 11. Алгоритмы для SBRS с открытым исходным кодом

№	Наименование алгоритма	Решаемая задача по рекомендации	Используемая модель	Место презентации алгоритма	Ссылка на исходный код
1	TBP [78]	Следующая корзина покупок	Анализ шаблонов	ICDM 2017	https://github.com/GiulioRossetti/tbp-next-basket
2	UP-CF [79]	Следующая корзина покупок	К-ближайших соседей	UMAP 2020	https://github.com/MayloIFERR/RACF
3	FPMC [22]	Следующая корзина покупок	Цепь Маркова	WWW 2010	https://github.com/khesui/FPMC
4	HRM [80]	Следующая корзина покупок	Распределенное представление	SIGIR 2015	https://github.com/chenghu17/Sequential_Recommendation
5	DREAM [19]	Следующая корзина покупок	Рекуррентная нейронная сеть	SIGIR 2016	https://github.com/yihong-chen/DREAM
6	Beacon [81]	Следующая корзина покупок	Рекуррентная нейронная сеть	IJCAI 2019	https://github.com/PreferredAI/beacon
7	TIFUKNN [82]	Следующая корзина покупок	К-ближайших соседей	SIGIR 2020	https://github.com/HaojiHu/TIFUKNN
8	AR [37]	Следующий элемент	Ассоциативные правила	UMUAI 2018	https://github.com/rn5l/session-rec
9	BPR-MF [37, 83]	Следующий элемент	Скрытого фактора	UAI 2009	https://github.com/rn5l/session-rec
10	IKNN [38]	Следующий элемент	К-ближайших соседей	RecSys 2017	https://github.com/rn5l/session-rec
11	SKNN [38]	Следующий элемент	К-ближайших соседей	RecSys 2017	https://github.com/rn5l/session-rec
12	FOSSIL [84]	Следующий элемент	Скрытого фактора	ICDM 2016	https://github.com/rn5l/session-rec
13	SMF [37]	Следующий элемент	Скрытого фактора	UMUAI 2018	https://github.com/rn5l/session-rec
14	GRU4Rec [85, 17]	Следующий элемент	Рекуррентная нейронная сеть	ICLR 2016	https://github.com/rn5l/session-rec
15	STAMP [20]	Следующий элемент	Модель внимания	KDD 2018	https://github.com/rn5l/session-rec
16	NARM [86]	Следующий элемент	Модель внимания, Рекуррентная нейронная сеть	CIKM 2017	https://github.com/rn5l/session-rec
17	SR-GNN [87]	Следующий элемент	Графовая нейронная сеть	AAAI 2019	https://github.com/CRIPAC-DIG/SR-GNN
18	CSRM [76]	Следующий элемент	Сеть памяти	SIGIR 2019	https://github.com/wmeirui/CSRM_SIGIR2019
19	RepeatNet [88]	Следующий элемент	Рекуррентная нейронная сеть, Модель внимания	AAAI 2019	https://github.com/PengjieRen/RepeatNet
20	DGRec [89]	Следующий элемент	Графовая нейронная сеть	WSDM 2019	https://github.com/DeepGraphLearning/RecommenderSystems/tree/master/socialRec
21	FGNN [67]	Следующий элемент	Графовая нейронная сеть	CIKM 2019	https://github.com/RuihongQiu/FGNN
22	TAGNN [65]	Следующий элемент	Графовая	SIGIR 2020	https://github.com/CRIP

			нейронная сеть		AC-DIG/TAGNN
23	LESSR [90]	Следующий элемент	Графовая нейронная сеть	KDD 2020	https://github.com/twchen/lessr
24	MKM-SR [12]	Следующий элемент	Рекуррентная нейронная сеть, Графовая нейронная сеть	SIGIR 2020	https://github.com/ciecus/MKM-SR

Для испытания и оценки алгоритмов необходимы соответствующие наборы данных (датасеты). В таб. 12 представлена консолидированная информация об общедоступных реальных наборах данных обычно используемых для оценки качества алгоритмов SBRS [6]:

Таблица 1 Наборы данных для оценки качества алгоритмов SBRS

№	Группа	Название набора данных	Количество			Средняя длитель-ность сессии*	Ссылка
			сессий	взаимо-действий	элементов		
1	Электронная коммерция	RSC 2015	1 375 128	5 426 961	28 582	3,95	https://www.kaggle.com/chadgostopp/recsys-challenge-2015
2		Tmall	1 774 729	13 418 695	425 348	7,56	https://tianchi.aliyun.com/dataset/dataDetail?dataId=42
3		Tafeng	19 538	144 777	5 263	7,41	https://www.kaggle.com/chiranjivdas09/ta-feng-grocery-dataset
4		Diginetica	573 935	2 451 565	134 319 529	4,27	https://competitions.codalab.org/competitions/11161
5		RetailRocket	59 962	212 182	31 968	3,54	https://www.kaggle.com/retailrocket/ecommerce-dataset
6	Новости	CLEF 2017	1 644 442	5 540 486	742	3,37	https://www.newsreelchallenge.org/dataset/
7		Globo	1 031 167	2 930 849	13 092	2,84	https://www.kaggle.com/gspmoreira/news-portal-user-interactions-by-globocom
8		Adressa 16G	2 215	62 908	6 765	28,40	http://reclab.idi.ntnu.no/dataset/
9	Музыка	Last.FM	169 576	2 887 349	449 037	17,03	http://millionsongdataset.com/lastfm/
10		30Music	2 764 474	31 351 954	210 633	11,34	http://recsys.deib.polimi.it/datasets/
11		NowPlaying	27 005	271 177	75 169	10,04	https://www.kaggle.com/chelseapower/nowplayingrs
12	Достопримечательности	Gowalla	**	245 157	6 871	-	http://snap.stanford.edu/data/loc-gowalla.html
13		Foursquare	**	155 365	2 675	-	https://www.kaggle.com/chetanism/foursquare-nyc-and-tokyo-checkin-dataset

*Средняя длительность сессии – среднее количество взаимодействий, приходящихся на одну сессию.

**Необработанные данные в группе «Достопримечательности» не имеют сессионной структуры, при анализе исследователи часто создают сессии вручную, рассматривая в качестве сессии взаимодействия пользователя в течение дня.

B.Библиотека Transformers4Rec

В настоящее время на рынке существует множество различных систем рекомендаций [91], но лишь единицы из них позиционируются, как системы рекомендаций на основе сессий или с поддержкой данного функционала.

В этом разделе рассмотрено одно из современных промышленных решений для разработки систем рекомендаций на основе сессий и на основе последовательностей – библиотека Transformers4Rec.

Transformers4Rec – это библиотека с открытым исходным кодом, разработанная командой NVIDIA Merlin и представленная сообществу разработчиков в 2021 году [92].

Библиотека Transformers4Rec тесно интегрирована с библиотекой обработки естественного языка HuggingFace Transformers и доступна для использования в PyTorch и TensorFlow. За счет интеграции с HuggingFace Transformers библиотека Transformers4Rec выступает в качестве связующего звена между

процессами обработки естественного языка (natural language processing, NLP) и генерацией рекомендаций. Использование библиотеки HuggingFace Transformers предоставляет возможность использовать более 60-ти эффективных и стандартизированных реализаций современных архитектур Transformers [93].

Библиотека Transformers4Rec имеет модульную структуру, позволяющую создавать разработчику собственные архитектуры [93].

Сценарий использования Transformers4Rec представлен на рис. 4, он включает в себя три основных этапа [93]:

1. Предварительная обработка данных;
2. Обучение и оценка модели;
3. Выдача результатов.

Рассмотрим подробнее первые два этапа.

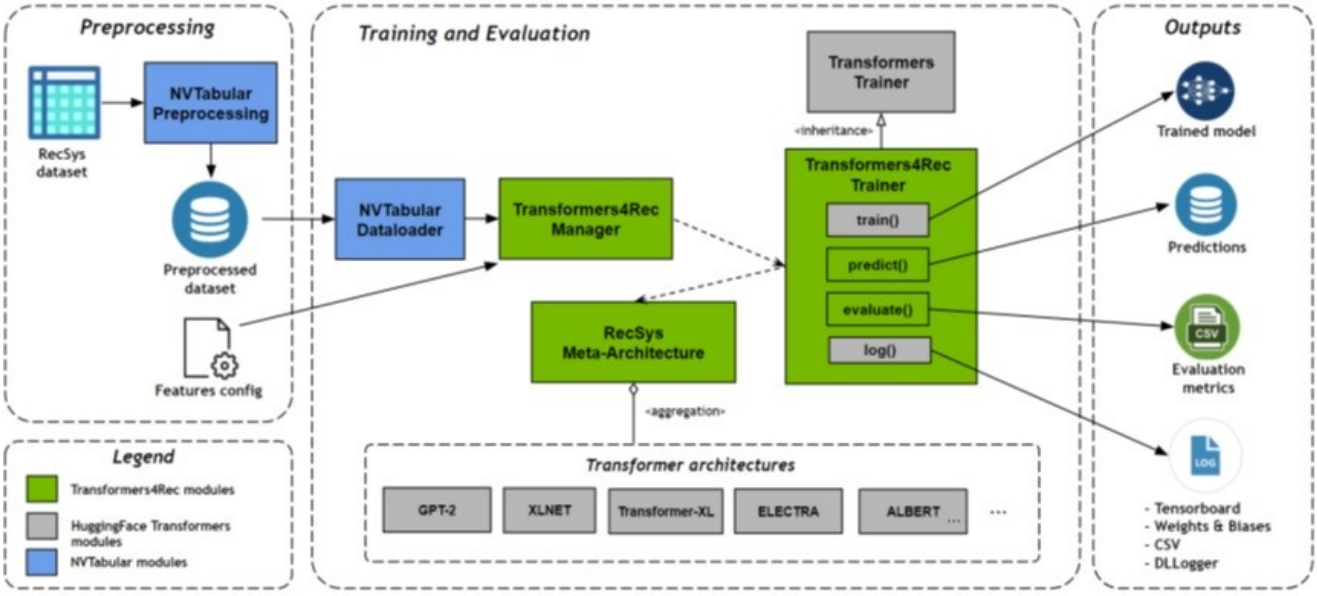

Рис. 4. Сценарий использования Transformers4Rec [93]

Результаты предварительной обработки данных библиотека NVTabular сохраняет в формате Apache Parquet, который ориентирован на эффективное хранение и извлечение данных [93].

В Transformers4Rec на этапе предварительной обработки также предусмотрено создание конфигурационного файла для следующего этапа обучения модели, позволяющего определить перечень функций, которые должны использоваться моделью, их тип (например, непрерывные или категориальные) и метаданные (например, мощность) [93].

2)Обучение модели и оценка рекомендаций

Transformers4Rec использует для загрузки сформированных на предыдущем этапе данных загрузчик библиотеки NVTabular. Загрузчик считывает файлы Parquet непосредственно в память графического процессора, что позволяет ускорить процессы обучения и оценки [93].

1) Предварительная обработка данных

В промышленном применении систем рекомендаций предварительная обработка данных часто является узким местом. Посредством интеграции с библиотекой NVIDIA NVTabular библиотека Transformers4Rec позволяет проводить ускоренную обработку больших объемов данных терабайтного масштаба за счет использования графических процессоров (graphics processing unit, GPU) [93].

Библиотека NVTabular также разработана с учетом специфики систем рекомендаций на основе сессий и на основе последовательностей. Например, в ней реализованы специальные операции, такие как группировка по пользователю/сессии отсортированных по времени взаимодействий, или усечение последовательностей в объеме N-первых или N-последних взаимодействий [93].

Библиотека HuggingFace Transformers имеет свой собственный оптимизированный процесс обучения и оценки для задач обработки естественного языка, за управление которым отвечает класс Trainer. Библиотека Transformers4Rec наследует и переопределяет методы predict() и evaluate() данного класса, адаптируя их к задаче генерации рекомендаций. Метод train() не меняется поскольку он идентичен как для задач обработки естественного языка, так и для задач генерации рекомендаций [93].

Оценка выполняется с использованием традиционных «Top-N» показателей ранжирования, которые могут выводить данные в различных форматах (см. рис. 4, блок «Outputs») [93].

Библиотека Transformers4Rec поддерживает протокол инкрементного обучения и оценки [94, 95, 96]. Данный протокол имитирует реалистичный производственный сценарий работы системы рекомендаций. В соответствии с данным сценарием, модель системы рекомендаций с заданной частотой (раз в день/раз в час

и т.д.) переобучается на основе потока поступающих данных. После переобучения модель вводиться в эксплуатацию и используется для генерации рекомендаций в последующих сессиях. Далее с заданной частотой процесс повторяется.

Практические эксперименты, проведенные разработчиками библиотеки Transformers4Rec, показали высокую производительность и эффективность рекомендаций на основе сессий в решениях на базе данной библиотеки [93].

Кроме того, решения команды NVIDIA, реализованные на базе библиотеки Transformers4Rec, победили в недавних конкурсах систем рекомендаций на основе сессий: WSDM Web Tour Workshop Challenge 2021, организованный Booking.com, и SIGIR eCommerce Workshop Data Challenge 2021, организованный Coveo [93].

VI Заключение

Системы рекомендаций на основе сессий (SBRS) являются перспективным направлением в разработке систем рекомендаций, так как они позволяют выявлять и учитывать краткосрочные предпочтения пользователей, заложенные в самых последних взаимодействиях пользователей, а также динамику изменения предпочтений. Данные возможности позволяют SBRS генерировать для пользователя более точные индивидуальные рекомендации.

В настоящей работе представлен обзор современного состояния дел в области исследований SBRS, в том числе: проведена классификация решаемых SBRS задач, рассмотрены входные/выходные данные для SBRS, определены основные сущности SBRS и в общем виде сформулирована задача SBRS.

Так как принцип работы SBRS основан на детальном анализе сессий, в работе подробно описаны основные свойства сессий, представлена классификация сессий в зависимости от их свойств, рассмотрены характерные особенности сессий и проблемные вопросы SBRS в зависимости от свойств сессии.

В работе также представлен детальный обзор, классификация и сравнение основных подходов к реализации SBRS, включающих в себя, как традиционные подходы, так и современные подходы на основе глубоких нейронных сетей и продвинутых моделей и алгоритмов.

В заключении работы приводится консолидированные сведения о программных реализациях современных алгоритмов SBRS и наборов данных (датасетов), используемых для тестирования алгоритмов SBRS, а также рассмотрено одно из современных промышленных решений для разработки систем рекомендаций на основе сессий – библиотека с открытым исходным кодом Transformers4Rec.

Это первая известная нам работа на русском языке, посвященная рекомендательным системам для сессий. Это обзорная статья, выполненная в рамках магистерской программы обучения на факультете ВМК МГУ имени М.В. Ломоносова. Наши собственные результаты последуют уже в форме магистерских диссертаций. Рекомендательные системы часто становятся объектами атак, и изучение таких процессов входит в магистерскую программу "Искусственный интеллект в кибербезопасности" [97]. Например, генерация фальшивых отзывов в системе электронной коммерции в случае периодического дообучения рекомендательной системы есть ни что иное, как атака отравлением данных [98]. И в этом смысле рассматриваемые в статье подходы, связанные с машинным обучением также нуждаются в устойчивых моделях [99].

Благодарности

Мы благодарны сотрудникам кафедры Информационной безопасности факультета Вычислительной математики и кибернетики МГУ имени М.В. Ломоносова за ценные обсуждения данной работы.

Исследование выполнено при поддержке Междисциплинарной научно-образовательной школы Московского университета «Мозг, когнитивные системы, искусственный интеллект».

Библиография

[1] Deuk Hee Park, Hyea Kyeong Kim, Il Young Choi, Jae Kyeong Kim. A Literature Review and Classification of Recommender Systems on Academic Journals. // Journal of Intelligence and Information Systems. 2011.

[2] Shoujin Wang, Gabriella Pasi, Liang Hu, and Longbing Cao. The Era of Intelligent Recommendation: Editorial on Intelligent Recommendation with Advanced AI and Learning. // IEEE Intelligent Systems. 2020.

[3] Charu C Aggarwal. Content-based recommender systems. // Recommender Systems. Springer. 2016.

[4] J Ben Schafer, Dan Frankowski, Jon Herlocker, and Shilad Sen. Collaborative filtering recommender systems. // Adaptive Web. Springer. 2007.

[5] Robin Burke. Hybrid recommender systems: survey and experiments. // User Modeling and User-Adapted Interaction 12(4). 2002.

[6] Shoujin Wang, Longbing Cao, Yan Wang, Quan Z. Sheng, Mehmet A. Orgun, and Defu Lian. A Survey on Session-based Recommender Systems. // ACM Comput. Surv. 9, 4, Article 39. 2021.

[7] Dietmar Jannach, Malte Ludewig, and et al. Session-based item recommendation in e-commerce: on short-term intents, reminders, trends and discounts. // User Modeling and User-Adapted Interaction 27(6). 2017.

[8] Shahab Saquib Sohail, Jamshed Siddiqui, Rashid Ali. Classifications of Recommender Systems: A review. // Engineering Science and Technology Review. 2017.

[9] J. Bobadilla, F. Ortega, A. Hernando, A. Gutierrez. Recommender systems survey. // Recommender systems survey. Knowledge-Based Systems 46, с. 109–132. 2013.

[10] Fajie Yuan, Alexandros Karatzoglou и др. A simple convolutional generative network for next item recommendation. // WSDM, с. 582–590. 2019.

[11] Shoujin Wang, Liang Hu, Yan Wang и др. Sequential recommender systems: challenges, progress and prospects. // IJCAI. AAAI Press, с. 6332–6338. 2019.

[12] Wenjing Meng, Deqing Yang, Yanghua Xiao. Incorporating user micro-behaviors and item knowledge into multi-task learning for session-based recommendation. // SIGIR. 2020.

[13] Ivica Obadic, Gjorgji Madjarov, Ivica Dimitrovski, and Dejan Gjorgjevikj. Addressing Item-Cold Start Problem in Recommendation Systems using Model Based Approach and Deep Learning. // ICT Innovations. 2017.

[14] Yibo Chen, Chanle Wu, Ming Xie, Xiaojun Guo. Solving the Sparsity Problem in Recommender Systems Using Association Retrieval. // Journal of Computers 6(9), с. 1896-1902. 2011.
[15] Malte Ludewig, Noemi Mauro и др. Performance comparison of neural and non-neural approaches to session-based recommendation. // RecSys. с. 462–466. 2019.
[16] Shuai Zhang, Lina Yao, Aixin Sun, Yi Tay. Deep learning based recommender system: a survey and new perspectives. // CSUR (52, 1), с. 1–38. 2019.
[17] Balzs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, Domonkos Tikk. Session-based recommendations with recurrent neural networks. // ICLR, с. 1–10. 2016.
[18] Jiaxuan You, Yichen Wang, Aditya Pal, Pong Eksombatchai и др. Hierarchical temporal convolutional networks for dynamic recommender systems. // WWW. с. 2236–2246. 2019.
[19] Feng Yu, Qiang Liu и др. A dynamic recurrent model for next basket recommendation. // SIGIR. ACM, с. 729–732. 2016.
[20] Qiao Liu, Yifu Zeng, Refuoe Mokhosi, Haibin Zhang. STAMP: short-Term attention/memory priority model for session-based recommendation. // SIGKDD. ACM, с. 1831–1839. 2018.
[21] Shoujin Wang, Liang Hu, Longbing Cao и др. Attention-based transactional context embedding for next-item recommendation. // AAAI, с. 2532–2539. 2018.
[22] Steffen Rendle, Christoph Freudenthaler, Lars Schmidt-Thieme. Factorizing personalized markov chains for next-basket recommendation. // WWW. ACM, с. 811–820. 2010.
[23] Jiawei Han, Jian Pei, Yiwen Yin. Mining frequent patterns without candidate generation. // ACM Sigmod Record, Vol. 29. ACM, 1–12. 2000.
[24] Bamshad Mobasher and et al. 2001. Effective personalization based on association rule discovery from web usage data. In WIDM. ACM, 9–15.
[25] Shoujin Wang and Longbing Cao. 2017. Inferring implicit rules by learning explicit and hidden item dependency. IEEE Transactions on Systems, Man, and Cybernetics: Systems 50, 3 (2017), 935–946.
[26] R Forsati, MR Meybodi, and A Ghari Neiat. 2009. Web page personalization based on weighted association rules. In ICECT. IEEE, 130–135.
[27] Liang Yan and Chunping Li. 2006. Incorporating pageview weight into an association-rule-based web recommendation system. In AI. Springer, 577–586.
[28] Marıa N Moreno, Francisco J Garcıa, and et al. 2004. Using association analysis of web data in recommender systems. In EC-Web. Springer, 11–20.
[29] 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.
[30] Ghim-Eng Yap, Xiao-Li Li, and S Yu Philip. 2012. Effective next-items recommendation via personalized sequential pattern mining. In DASFAA. Springer, 48–64.
[31] Utpala Niranjan, RBV Subramanyam, and V Khanaa. 2010. Developing a web recommendation system based on closed sequential patterns. In ICT. Springer, 171–179.
[32] Wei Song and Kai Yang. 2014. Personalized recommendation based on weighted sequence similarity. In Practical Applications of Intelligent Systems. Springer, 657–666.
[33] 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.
[34] 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.
[35] Xiangyu Zhao, Liang Zhang, Long Xia, Zhuoye Ding, Dawei Yin, and Jiliang Tang. 2017. Deep reinforcement learning for list-wise recommendations. arXiv preprint arXiv:1801.00209 (2017).
[36] ShoujinWang, Liang Hu, YanWang, and et al. 2020. Intention2Basket: a neural intention-driven approach for dynamic next-basket planning. In IJCAI. 2333–2339.
[37] Malte Ludewig and Dietmar Jannach. 2018. Evaluation of session-based recommendation algorithms. UMUAI 28, 4-5 (2018), 331–390.
[38] Dietmar Jannach and Malte Ludewig. 2017. When recurrent neural networks meet the neighborhood for session-based recommendation. In RecSys. ACM, 306–310.
[39] Guy Shani, David Heckerman, and Ronen I Brafman. 2005. An MDP-based recommender system. JMLR 6, Sep (2005), 1265–1295.
[40] Magdalini Eirinaki, Michalis Vazirgiannis, and et al. 2005. Web path recommendations based on page ranking and markov models. In WIDM. ACM, 2–9.
[41] Zhiyong Zhang and Olfa Nasraoui. 2007. Efficient hybrid Web recommendations based on Markov click stream models and implicit search. In WI. 621–627.
[42] Shuo Chen, Josh L Moore, and et al. 2012. Playlist prediction via metric embedding. In SIGKDD. ACM, 714–722.
[43] Xiang Wu, Qi Liu, Enhong Chen, Liang He, and et al. 2013. Personalized next-song recommendation in online karaokes. In RecSys. ACM, 137–140.
[44] Negar Hariri, Bamshad Mobasher, and Robin Burke. 2012. Context-aware music recommendation based on latent topic sequential patterns. In RecSys. 131–138.
[45] Elena Zheleva, John Guiver, Eduarda Mendes Rodrigues, and et al. 2010. Statistical models of music-listening sessions in social media. In WWW. 1019–1028.
[46] Priit Järv. 2019. Predictability limits in session-based next item recommendation. In RecSys. 146–150.
[47] 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.
[48] Fajie Yuan, Alexandros Karatzoglou, and et al. 2019. A simple convolutional generative network for next item recommendation. In WSDM. 582–590.
[49] Chen Cheng, Haiqin Yang, Michael R Lyu, and Irwin King. 2013. Where you like to go next: successive point-of-interest recommendation. In IJCAI. 2605–2611.
[50] Dawen Liang, Jaan Altosaar, and et al. 2016. Factorization meets the item embedding: regularizing matrix factorization with item co-occurrence. In RecSys. ACM, 59–66.
[51] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In EMNLP. 1532–1543.
[52] Tomas Mikolov, Quoc V Le, and Ilya Sutskever. 2013. Exploiting similarities among languages for machine translation. arXiv preprint arXiv:1309.4168 (2013).
[53] Liang Hu, Longbing Cao, ShoujinWang, and et al. 2017. Diversifying personalized recommendation with user-session context. In IJCAI. 1858–1864.
[54] Shoujin Wang, Liang Hu, and Longbing Cao. 2017. Perceiving the next choice with comprehensive transaction embeddings for online recommendation. In ECML-PKDD. Springer, 285–302.
[55] Xiangyu Zhao, Long Xia, Liang Zhang, Zhuoye Ding, Dawei Yin, and Jiliang Tang. 2018. Deep reinforcement learning for page-wise recommendations. In RecSys. 95–103.
[56] Yong Kiam Tan, Xinxing Xu, and Yong Liu. 2016. Improved recurrent neural networks for session-based recommendations. In DLRS. ACM, 17–22.
[57] Chen Wu and Ming Yan. 2017. Session-aware information embedding for e-commerce product recommendation. In CIKM. ACM, 2379–2382.
[58] Dietmar Jannach, Malte Ludewig, and et al. 2017. Session-based item recommendation in e-commerce: on short-term intents, reminders, trends and discounts. UMUAI 27, 3-5 (2017), 351–392.
[59] Yang Song and et al. 2016. Multi-rate deep learning for temporal recommendation. In SIGIR. ACM, 909–912.
[60] Fajie Yuan, Xiangnan He, Haochuan Jiang, Guibing Guo, and et al. 2020. Future data helps training: modeling future contexts for session-based recommendation. In The Web Conference. 303–313.
[61] Jiaxi Tang and KeWang. 2018. Personalized top-n sequential recommendation via convolutional sequence embedding. In WSDM. 565–573.
[62] Trinh Xuan Tuan and Tu Minh Phuong. 2017. 3D convolutional networks for session-based recommendation with content features. In RecSys. ACM, 138–146.
[63] Keunchan Park, Jisoo Lee, and Jaeho Choi. 2017. Deep neural networks for news recommendations. In CIKM. ACM, 2255–2258.
[64] Chengfeng Xu, Pengpeng Zhao, Yanchi Liu, and et al. 2019. Graph contextualized self-attention network for session based recommendation. In IJCAI. 3940–3946.
[65] Feng Yu and et al. 2020. TAGNN: target attentive graph neural networks for session-based recommendation. In SIGIR.1–5.
[66] Wen Wang, Wei Zhang, Shukai Liu, and et al. 2020. Beyond clicks: modeling multi-relational item graph for sessionbased target behavior prediction. In The Web Conference. 3056–3062.
[67] Ruihong Qiu, Jingjing Li, Zi Huang, and Hongzhi Yin. 2019. Rethinking the item order in session-based recommendation with graph neural networks. In CIKM. 579–588.
[68] Mao Ye, Xingjie Liu, and Wang-Chien Lee. 2012. Exploring social influence for recommendation: a generative model approach. In SIGIR. 671–680.
[69] Ruihong Qiu, Zi Huang, Jingjing Li, and Hongzhi Yin. 2020. Exploiting

cross-session information for session-based recommendation with graph neural networks. TOIS 38 (2020), 1–23. Issue 3
[70] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In NIPS. 5998–6008.
[71] Shoujin Wang, Liang Hu, Yan Wang, Quan Z. Sheng, Mehmet Orgun, and Longbing Cao. 2019. Modeling multipurposesessions for next-item recommendations via mixture-channel purpose routing networks. In IJCAI. AAAI Press, 3771–3777.
[72] Shoujin Wang, Longbing Cao, Liang Hu, Shlomo Berkovsky, Xiaoshui Huang, Lin Xiao, and Wenpeng Lu. 2020. Jointly modeling intra- and inter-transaction dependencies with hierarchical attentive transaction embeddings for next-item recommendation. IEEE Intelligent Systems (2020), 1–7.
[73] Haochao Ying, Fuzhen Zhuang, Fuzheng Zhang, and et al. 2018. Sequential recommender system based on hierarchical attention network. In IJCAI. 3926–3932.
[74] Xu Chen, Hongteng Xu, Yongfeng Zhang, and et al. 2018. Sequential recommendation with user memory networks. In WSDM. 108–116.
[75] Adam Santoro, Sergey Bartunov, Matthew Botvinick, and et al. 2016. Meta-learning with memory-augmented neural networks. In ICML. 1842–1850.
[76] Meirui Wang, Pengjie Ren, Lei Mei, Zhumin Chen, Jun Ma, and Maarten de Rijke. 2019. A collaborative session-based recommendation approach with parallel memory modules. In SIGIR. 345–354.
[77] Jiaxi Tang, Francois Belletti, Sagar Jain, Minmin Chen, and et al. 2019. Towards neural mixture recommender for long range dependent user sequences. In WWW. 1782–1793.
[78] Riccardo Guidotti, Giulio Rossetti, Luca Pappalardo, Fosca Giannotti, and Dino Pedreschi. 2017. Market basket prediction using user-centric temporal annotated recurring sequences. In ICDM. IEEE, 895–900.
[79] Guglielmo Faggioli, Mirko Polato, and Fabio Aiolli. 2020. Recency aware collaborative filtering for next basket recommendation. In UMAP. 80–87.
[80] Pengfei Wang, Jiafeng Guo, Yanyan Lan, Jun Xu, Shengxian Wan, and Xueqi Cheng. 2015. Learning hierarchical representation model for next basket recommendation. In SIGIR. ACM, 403–412.
[81] Duc-Trong Le, Hady W Lauw, and Yuan Fang. 2019. Correlation-sensitive next-basket recommendation. In IJCAI. AAAI Press, 2808–2814.
[82] Haoji Hu, Xiangnan He, Jinyang Gao, and Zhi-Li Zhang. 2020. Modeling personalized item frequency information for next-basket recommendation. In SIGIR. ACM.
[83] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2012. BPR: Bayesian personalized ranking from implicit feedback. arXiv preprint arXiv:1205.2618 (2012).
[84] Ruining He and Julian McAuley. 2016. Fusing similarity models with markov chains for sparse sequential recommendation. In ICDM. IEEE, 191–200.
[85] Balázs Hidasi and Alexandros Karatzoglou. 2018. Recurrent neural networks with top-k gains for session-based recommendations. In CIKM. 843–852.
[86] Jing Li, Pengjie Ren, Zhumin Chen, Zhaochun Ren, and et al. 2017. Neural attentive session-based recommendation. In CIKM. ACM, 1419–1428.
[87] Shu Wu, Yuyuan Tang, Yanqiao Zhu, and et al. 2019. Session-based recommendation with graph neural networks. In AAAI. 346–353.
[88] Pengjie Ren, Zhumin Chen, Jing Li, and et al. 2019. RepeatNet: a repeat aware neural recommendation machine for session-based recommendation. In AAAI, Vol. 33. 4806–4813.
[89] Bo Song, Yi Cao, and et al. 2019. Session-based recommendation with hierarchical memory networks. In CIKM. 2181–2184.
[90] Tianwen Chen and Raymond Chi-Wing Wong. 2020. Handling information loss of graph neural networks for session-based recommendation. In SIGKDD. 1172–1180.
[91] List of Recommender Systems. https://github.com/grahamjen-son/list_of_recommender_systems. Retrieved: May, 2022.
[92] NVIDIA-Merlin/Transformers4Rec. https://github.com/NVIDIA-Merlin/Transformers4Rec. Retrieved: May, 2022.
[93] Gabriel de Souza Pereira Moreira, Sara Rabhi, Jeong Min Lee, Ronay Ak, Even Oldridge. Transformers4Rec: Bridging the Gap between NLP and Sequential / Session-Based Recommendation. RecSys '21: Fifteenth ACM Conference on Recommender Systems. September 2021, pages 143–153.
[94] Gabriel de Souza Pereira Moreira, Felipe Ferreira, and Adilson Marques da Cunha. 2018. News session-based recommendations using deep neural networks. In Proceedings of the 3rd Workshop on Deep Learning for Recommender Systems. 15–23.
[95] Gabriel De Souza Pereira Moreira, Dietmar Jannach, and Adilson Marques Da Cunha. 2019. Contextual hybrid session-based news recommendation with recurrent neural networks. IEEE Access 7 (2019), 169185–169203.
[96] Shiming Sun, Yuanhe Tang, Zemei Dai, and Fu Zhou. 2019. Self-attention network for session-based recommendation with streaming data input. IEEE Access 7 (2019), 110499–110509.2.
[97] Artificial Intelligence in Cybersecurity. http://master.cmc.msu.ru/?q=ru/node/3496 (in Russian) Retrieved: May, 2022.
[98] Ilyushin, Eugene, Dmitry Namiot, and Ivan Chizhov. "Attacks on machine learning systems-common problems and methods." International Journal of Open Information Technologies 10.3 (2022): 17-22. (in Russian)
[99] Namiot, Dmitry, Eugene Ilyushin, and Ivan Chizhov. "The rationale for working on robust machine learning." International Journal of Open Information Technologies 9.11 (2021): 68-74.

Session-Based Recommender Systems - Models and Tasks

Dmitry Yakupov, Dmitry Namiot

Abstract— **Recommender systems were one of the first mass applications of data analysis in various fields. The reason is their final result (recommendations) that is transparent to end users and clear metrics for measuring the quality of their work. End-users can always evaluate the usefulness of recommendations, formal measurements can always operate on conversion, whatever it means - purchases of recommended products, clicks on links, etc. Most often, the work of recommender systems is based on the generalization and analysis of the preferences of other users (which includes consideration of various aspects of their behavior), and the available information about the current user. At the same time, there is a class of tasks when recommendations should (or only can) be based on the current actions of the user. For example, in an e-commerce system, an unauthorized (anonymous) user visits various pages of a site. Or the user's preferences in the system are only short-term. All these examples are typical for a separate large class of recommender systems - recommender systems for sessions, where a session is understood as a sequence of user actions. The recommender system in this case solves one of three tasks: recommends the next product (content, activity, etc.) within the current session, recommends the following products (activities, etc.) until the end of the current session, recommends the next possible session. The article contains an overview of the described tasks and models for such recommender systems.**

Keywords— **session-based recommender systems, short-term preferences**

References

[1] Deuk Hee Park, Hyea Kyeong Kim, Il Young Choi, Jae Kyeong Kim. A Literature Review and Classification of Recommender Systems on Academic Journals. // Journal of Intelligence and Information Systems. 2011.

[2] Shoujin Wang, Gabriella Pasi, Liang Hu, and Longbing Cao. The Era of Intelligent Recommendation: Editorial on Intelligent Recommendation with Advanced AI and Learning. // IEEE Intelligent Systems. 2020.

[3] Charu C Aggarwal. Content-based recommender systems. // Recommender Systems. Springer. 2016.

[4] J Ben Schafer, Dan Frankowski, Jon Herlocker, and Shilad Sen. Collaborative filtering recommender systems. // Adaptive Web. Springer. 2007.

[5] Robin Burke. Hybrid recommender systems: survey and experiments. // User Modeling and User-Adapted Interaction 12(4). 2002.

[6] Shoujin Wang, Longbing Cao, Yan Wang, Quan Z. Sheng, Mehmet A. Orgun, and Defu Lian. A Survey on Session-based Recommender Systems. // ACM Comput. Surv. 9, 4, Article 39. 2021.

[7] Dietmar Jannach, Malte Ludewig, and et al. Session-based item recommendation in e-commerce: on short-term intents, reminders, trends and discounts. // User Modeling and User-Adapted Interaction 27(6). 2017.

[8] Shahab Saquib Sohail, Jamshed Siddiqui, Rashid Ali. Classifications of Recommender Systems: A review. // Engineering Science and Technology Review. 2017.

[9] J. Bobadilla, F. Ortega, A. Hernando, A. Gutierrez. Recommender systems survey. // Recommender systems survey. Knowledge-Based Systems 46, с. 109–132. 2013.

[10] Fajie Yuan, Alexandros Karatzoglou и др. A simple convolutional generative network for next item recommendation. // WSDM, с. 582–590. 2019.

[11] Shoujin Wang, Liang Hu, Yan Wang и др. Sequential recommender systems: challenges, progress and prospects. // IJCAI. AAAI Press, с. 6332–6338. 2019.

[12] Wenjing Meng, Deqing Yang, Yanghua Xiao. Incorporating user micro-behaviors and item knowledge into multi-task learning for session-based recommendation. // SIGIR. 2020.

[13] Ivica Obadic, Gjorgji Madjarov, Ivica Dimitrovski, and Dejan Gjorgjevikj. Addressing Item-Cold Start Problem in Recommendation Systems using Model Based Approach and Deep Learning. // ICT Innovations. 2017.

[14] Yibo Chen, Chanle Wu, Ming Xie, Xiaojun Guo. Solving the Sparsity Problem in Recommender Systems Using Association Retrieval. // Journal of Computers 6(9), с. 1896-1902. 2011.

[15] Malte Ludewig, Noemi Mauro и др. Performance comparison of neural and non-neural approaches to session-based recommendation. // RecSys. с. 462–466. 2019.

[16] Shuai Zhang, Lina Yao, Aixin Sun, Yi Tay. Deep learning based recommender system: a survey and new perspectives. // CSUR (52, 1), с. 1–38. 2019.

[17] Balzs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, Domonkos Tikk. Session-based recommendations with recurrent neural networks. // ICLR, с. 1–10. 2016.

[18] Jiaxuan You, Yichen Wang, Aditya Pal, Pong Eksombatchai и др. Hierarchical temporal convolutional networks for dynamic recommender systems. // WWW. с. 2236–2246. 2019.

[19] Feng Yu, Qiang Liu и др. A dynamic recurrent model for next basket recommendation. // SIGIR. ACM, с. 729–732. 2016.

[20] Qiao Liu, Yifu Zeng, Refuoe Mokhosi, Haibin Zhang. STAMP: short-Term attention/memory priority model for session-based recommendation. // SIGKDD. ACM, с. 1831–1839. 2018.

[21] Shoujin Wang, Liang Hu, Longbing Cao и др. Attention-based transactional context embedding for next-item recommendation. // AAAI, с. 2532–2539. 2018.

[22] Steffen Rendle, Christoph Freudenthaler, Lars Schmidt-Thieme. Factorizing personalized markov chains for next-basket recommendation. // WWW. ACM, с. 811–820. 2010.

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

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

[25] Shoujin Wang and Longbing Cao. 2017. Inferring implicit rules by learning explicit and hidden item dependency. IEEE Transactions on Systems, Man, and Cybernetics: Systems 50, 3 (2017), 935–946.

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

[27] Liang Yan and Chunping Li. 2006. Incorporating pageview weight into an association-rule-based web recommendation system. In AI. Springer, 577–586.

[28] Marна N Moreno, Francisco J Garcна, and et al. 2004. Using association analysis of web data in recommender systems. In EC-Web. Springer, 11–20.

[29] 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.

[30] Ghim-Eng Yap, Xiao-Li Li, and S Yu Philip. 2012. Effective next-items recommendation via personalized sequential pattern mining. In DASFAA. Springer, 48–64.

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

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

[33] 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.
[34] 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.
[35] Xiangyu Zhao, Liang Zhang, Long Xia, Zhuoye Ding, Dawei Yin, and Jiliang Tang. 2017. Deep reinforcement learning for list-wise recommendations. arXiv preprint arXiv:1801.00209 (2017).
[36] ShoujinWang, Liang Hu, YanWang, and et al. 2020. Intention2Basket: a neural intention-driven approach for dynamic next-basket planning. In IJCAI. 2333–2339.
[37] Malte Ludewig and Dietmar Jannach. 2018. Evaluation of session-based recommendation algorithms. UMUAI 28, 4-5 (2018), 331–390.
[38] Dietmar Jannach and Malte Ludewig. 2017. When recurrent neural networks meet the neighborhood for session-based recommendation. In RecSys. ACM, 306–310.
[39] Guy Shani, David Heckerman, and Ronen I Brafman. 2005. An MDP-based recommender system. JMLR 6, Sep (2005), 1265–1295.
[40] Magdalini Eirinaki, Michalis Vazirgiannis, and et al. 2005. Web path recommendations based on page ranking and markov models. In WIDM. ACM, 2–9.
[41] Zhiyong Zhang and Olfa Nasraoui. 2007. Efficient hybrid Web recommendations based on Markov click stream models and implicit search. In WI. 621–627.
[42] Shuo Chen, Josh L Moore, and et al. 2012. Playlist prediction via metric embedding. In SIGKDD. ACM, 714–722.
[43] Xiang Wu, Qi Liu, Enhong Chen, Liang He, and et al. 2013. Personalized next-song recommendation in online karaokes. In RecSys. ACM, 137–140.
[44] Negar Hariri, Bamshad Mobasher, and Robin Burke. 2012. Context-aware music recommendation based on latent topic sequential patterns. In RecSys. 131–138.
[45] Elena Zheleva, John Guiver, Eduarda Mendes Rodrigues, and et al. 2010. Statistical models of music-listening sessions in social media. In WWW. 1019–1028.
[46] Priit Järv. 2019. Predictability limits in session-based next item recommendation. In RecSys. 146–150.
[47] 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.
[48] Fajie Yuan, Alexandros Karatzoglou, and et al. 2019. A simple convolutional generative network for next item recommendation. In WSDM. 582–590.
[49] Chen Cheng, Haiqin Yang, Michael R Lyu, and Irwin King. 2013. Where you like to go next: successive point-of-interest recommendation. In IJCAI. 2605–2611.
[50] Dawen Liang, Jaan Altosaar, and et al. 2016. Factorization meets the item embedding: regularizing matrix factorization with item co-occurrence. In RecSys. ACM, 59–66.
[51] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In EMNLP. 1532–1543.
[52] Tomas Mikolov, Quoc V Le, and Ilya Sutskever. 2013. Exploiting similarities among languages for machine translation. arXiv preprint arXiv:1309.4168 (2013).
[53] Liang Hu, Longbing Cao, ShoujinWang, and et al. 2017. Diversifying personalized recommendation with user-session context. In IJCAI. 1858–1864.
[54] Shoujin Wang, Liang Hu, and Longbing Cao. 2017. Perceiving the next choice with comprehensive transaction embeddings for online recommendation. In ECML-PKDD. Springer, 285–302.
[55] Xiangyu Zhao, Long Xia, Liang Zhang, Zhuoye Ding, Dawei Yin, and Jiliang Tang. 2018. Deep reinforcement learning for page-wise recommendations. In RecSys. 95–103.
[56] Yong Kiam Tan, Xinxing Xu, and Yong Liu. 2016. Improved recurrent neural networks for session-based recommendations. In DLRS. ACM, 17–22.
[57] Chen Wu and Ming Yan. 2017. Session-aware information embedding for e-commerce product recommendation. In CIKM. ACM, 2379–2382.
[58] Dietmar Jannach, Malte Ludewig, and et al. 2017. Session-based item recommendation in e-commerce: on short-term intents, reminders, trends and discounts. UMUAI 27, 3-5 (2017), 351–392.
[59] Yang Song and et al. 2016. Multi-rate deep learning for temporal recommendation. In SIGIR. ACM, 909–912.
[60] Fajie Yuan, Xiangnan He, Haochuan Jiang, Guibing Guo, and et al. 2020. Future data helps training: modeling future contexts for session-based recommendation. In The Web Conference. 303–313.
[61] Jiaxi Tang and KeWang. 2018. Personalized top-n sequential recommendation via convolutional sequence embedding. In WSDM. 565–573.
[62] Trinh Xuan Tuan and Tu Minh Phuong. 2017. 3D convolutional networks for session-based recommendation with content features. In RecSys. ACM, 138–146.
[63] Keunchan Park, Jisoo Lee, and Jaeho Choi. 2017. Deep neural networks for news recommendations. In CIKM. ACM, 2255–2258.
[64] Chengfeng Xu, Pengpeng Zhao, Yanchi Liu, and et al. 2019. Graph contextualized self-attention network for session based recommendation. In IJCAI. 3940–3946.
[65] Feng Yu and et al. 2020. TAGNN: target attentive graph neural networks for session-based recommendation. In SIGIR.1–5.
[66] Wen Wang, Wei Zhang, Shukai Liu, and et al. 2020. Beyond clicks: modeling multi-relational item graph for sessionbased target behavior prediction. In The Web Conference. 3056–3062.
[67] Ruihong Qiu, Jingjing Li, Zi Huang, and Hongzhi Yin. 2019. Rethinking the item order in session-based recommendation with graph neural networks. In CIKM. 579–588.
[68] Mao Ye, Xingjie Liu, and Wang-Chien Lee. 2012. Exploring social influence for recommendation: a generative model approach. In SIGIR. 671–680.
[69] Ruihong Qiu, Zi Huang, Jingjing Li, and Hongzhi Yin. 2020. Exploiting cross-session information for session-based recommendation with graph neural networks. TOIS 38 (2020), 1–23. Issue 3
[70] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In NIPS. 5998–6008.
[71] Shoujin Wang, Liang Hu, Yan Wang, Quan Z. Sheng, Mehmet Orgun, and Longbing Cao. 2019. Modeling multipurposesessions for next-item recommendations via mixture-channel purpose routing networks. In IJCAI. AAAI Press, 3771–3777.
[72] Shoujin Wang, Longbing Cao, Liang Hu, Shlomo Berkovsky, Xiaoshui Huang, Lin Xiao, and Wenpeng Lu. 2020. Jointly modeling intra- and inter-transaction dependencies with hierarchical attentive transaction embeddings for next-item recommendation. IEEE Intelligent Systems (2020), 1–7.
[73] Haochao Ying, Fuzhen Zhuang, Fuzheng Zhang, and et al. 2018. Sequential recommender system based on hierarchical attention network. In IJCAI. 3926–3932.
[74] Xu Chen, Hongteng Xu, Yongfeng Zhang, and et al. 2018. Sequential recommendation with user memory networks. In WSDM. 108–116.
[75] Adam Santoro, Sergey Bartunov, Matthew Botvinick, and et al. 2016. Meta-learning with memory-augmented neural networks. In ICML. 1842–1850.
[76] Meirui Wang, Pengjie Ren, Lei Mei, Zhumin Chen, Jun Ma, and Maarten de Rijke. 2019. A collaborative session-based recommendation approach with parallel memory modules. In SIGIR. 345–354.
[77] Jiaxi Tang, Francois Belletti, Sagar Jain, Minmin Chen, and et al. 2019. Towards neural mixture recommender for long range dependent user sequences. In WWW. 1782–1793.
[78] Riccardo Guidotti, Giulio Rossetti, Luca Pappalardo, Fosca Giannotti, and Dino Pedreschi. 2017. Market basket prediction using user-centric temporal annotated recurring sequences. In ICDM. IEEE, 895–900.
[79] Guglielmo Faggioli, Mirko Polato, and Fabio Aiolli. 2020. Recency aware collaborative filtering for next basket recommendation. In UMAP. 80–87.
[80] Pengfei Wang, Jiafeng Guo, Yanyan Lan, Jun Xu, Shengxian Wan, and Xueqi Cheng. 2015. Learning hierarchical representation model for next basket recommendation. In SIGIR. ACM, 403–412.
[81] Duc-Trong Le, Hady W Lauw, and Yuan Fang. 2019. Correlation-sensitive next-basket recommendation. In IJCAI. AAAI Press, 2808–2814.
[82] Haoji Hu, Xiangnan He, Jinyang Gao, and Zhi-Li Zhang. 2020. Modeling personalized item frequency information for next-basket recommendation. In SIGIR. ACM.
[83] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2012. BPR: Bayesian personalized ranking from implicit feedback. arXiv preprint arXiv:1205.2618 (2012).
[84] Ruining He and Julian McAuley. 2016. Fusing similarity models with markov chains for sparse sequential recommendation. In ICDM. IEEE, 191–200.
[85] Balázs Hidasi and Alexandros Karatzoglou. 2018. Recurrent neural networks with top-k gains for session-based recommendations. In CIKM. 843–852.
[86] Jing Li, Pengjie Ren, Zhumin Chen, Zhaochun Ren, and et al. 2017. Neural attentive session-based recommendation. In CIKM. ACM, 1419–1428.
[87] Shu Wu, Yuyuan Tang, Yanqiao Zhu, and et al. 2019. Session-based recommendation with graph neural networks. In AAAI. 346–353.
[88] Pengjie Ren, Zhumin Chen, Jing Li, and et al. 2019. RepeatNet: a repeat aware neural recommendation machine for session-based recommendation. In AAAI, Vol. 33. 4806–4813.
[89] Bo Song, Yi Cao, and et al. 2019. Session-based recommendation with hierarchical memory networks. In CIKM. 2181–2184.
[90] Tianwen Chen and Raymond Chi-Wing Wong. 2020. Handling information loss of graph neural networks for session-based recommendation. In SIGKDD. 1172–1180.

[91] List of Recommender Systems. https://github.com/grahamjenson/list_of_recommender_systems. Retrieved: May, 2022.
[92] NVIDIA-Merlin/Transformers4Rec. https://github.com/NVIDIA-Merlin/Transformers4Rec. Retrieved: May, 2022.
[93] Gabriel de Souza Pereira Moreira, Sara Rabhi, Jeong Min Lee, Ronay Ak, Even Oldridge. Transformers4Rec: Bridging the Gap between NLP and Sequential / Session-Based Recommendation. RecSys '21: Fifteenth ACM Conference on Recommender Systems. September 2021, pages 143–153.
[94] Gabriel de Souza Pereira Moreira, Felipe Ferreira, and Adilson Marques da Cunha. 2018. News session-based recommendations using deep neural networks. In Proceedings of the 3rd Workshop on Deep Learning for Recommender Systems. 15–23.
[95] Gabriel De Souza Pereira Moreira, Dietmar Jannach, and Adilson Marques Da Cunha. 2019. Contextual hybrid session-based news recommendation with recurrent neural networks. IEEE Access 7 (2019), 169185–169203.
[96] Shiming Sun, Yuanhe Tang, Zemei Dai, and Fu Zhou. 2019. Self-attention network for session-based recommendation with streaming data input. IEEE Access 7 (2019), 110499–110509.2.
[97] Artificial Intelligence in Cybersecurity. http://master.cmc.msu.ru/?q=ru/node/3496 (in Russian) Retrieved: May, 2022.
[98] Ilyushin, Eugene, Dmitry Namiot, and Ivan Chizhov. "Attacks on machine learning systems-common problems and methods." International Journal of Open Information Technologies 10.3 (2022): 17-22. (in Russian)
[99] Namiot, Dmitry, Eugene Ilyushin, and Ivan Chizhov. "The rationale for working on robust machine learning." International Journal of Open Information Technologies 9.11 (2021): 68-74.

.