

Обзор методов построения рекомендательных систем на основе сессий

М.М.Ниничук, Д.Е.Намиот

Аннотация—Рекомендательные системы в настоящее время играют значимую роль во многих сферах, связанных с обработкой больших объемов информации, например онлайн-магазины и кинотеатры, сервисы для прослушивания музыки. Как правило, построение рекомендаций основывается на анализе информации о предпочтениях пользователя, полученных в прошлом. Часто такая информация представляется в виде матрицы пользователей–объектов. Однако в некоторых ситуациях информация о пользователе может быть недоступна, например, при первом либо анонимном посещении сервиса. Подобная постановка задачи характерна для отдельного класса рекомендательных систем — рекомендательных систем на основе сессий (Session-based Recommender Systems, SBRS). В отличие от классических подходов к построению рекомендательных систем, SBRS опираются только на данные о текущей сессии пользователя, что позволяет получать информацию о его быстро изменяющихся предпочтениях. Целью рекомендаций, как правило, является предсказание следующего объекта, на который обратит внимание пользователь, либо же множество таких объектов в текущей сессии. В данной статье приводится обзор основных алгоритмов, используемых при построении SBRS. Также приводится обзор и сравнение фреймворков, предназначенных для разработки таких рекомендательных систем.

Ключевые слова—рекомендательные системы на основе сессий, рекомендательные системы

I. Введение

Экспоненциальный рост информации в последние годы привел к информационной перегрузке [1], [2]. Людям приходится оперировать объемами информации, которые значительно превышают необходимый минимум для анализа ситуации, принятия решения. Это стало одной из причин увеличения роли рекомендательных систем (Recommender Systems, RS). Они позволяют принимать более эффективные, информативные и действенные решения во многих сферах жизни [3], [4], экономить время при поиске нужной информации или товара.

Значительная часть рекомендательных систем основаны на использовании информации об объектах (content-based filtering) и пользователе, его прошлых покупках (collaborative filtering) [5]. Подразумевается, что пользователи, чьи действия или предпочтения в прошлом схожи между собой, будут схожи и в дальнейшем. Это не совсем верно, так как предпочтения пользователей меняются со временем. Кроме того, в реальном мире долгосрочная информация о пользователе не всегда доступна, например, это может быть новый пользователь

или пользователь, не вошедший в систему. При этом возникает проблема "холодного старта"— непонятно, что можно рекомендовать таким пользователям.

Одним из возможных подходов могут стать не персонализированные рекомендации (можно рекомендовать объекты с самым высоким рейтингом, с наибольшим количеством просмотров). Однако это не единственный вариант, в случае отсутствия информации о пользователе для построения рекомендаций можно использовать информацию о его действиях в рамках текущей сессии, тем самым определить его текущие потребности [6], [7]. Данный вид алгоритмов называется рекомендательными системами для сессий (Session-based Recommender Systems, SBRS) [3].

В данном обзоре рассматриваются различные подходы к построению рекомендательных систем на основе данных сессий, их преимущества и недостатки. Также приводится описание фреймворков для построения рекомендательных систем.

II. Подходы к рекомендациям с использованием данных сессий

A. Описание данных

Как было сказано ранее, во многих областях релевантность рекомендуемого элемента зависит не только от общих предпочтений пользователей, но также от их текущей ситуации, а также их краткосрочных намерений и интересов [8]. Это стало предпосылкой к использованию данных сессий для построения рекомендаций.

Сессия — последовательность действий пользователя в течение некоторого ограниченного периода времени [5]. Минимальный временной промежуток между различными сессиями не определен, он может составлять как несколько минут, так и несколько недель или месяцев.

Некоторые характеристики сессии как последовательности действий и объектов могут влиять на рекомендации [3]:

- 1) Длина сессии — количество взаимодействий пользователя в данной сессии;
- 2) Упорядоченность — наличие или отсутствие порядка взаимодействий внутри сессии;
- 3) Тип взаимодействия — действие, которое пользователь совершил над объектом (например, просмотр, добавление в корзину или "избранное");
- 4) Анонимность — сессия анонимная, если информация о пользователе (его профиль) отсутствует.

Важная с точки зрения построения рекомендаций информация может содержаться в метаданных сессии - геопозиция пользователя, текущее время суток, контекст

Марина Матвеевна Ниничук, МГУ им. М.В. Ломоносова, (email: ninichuk.marina@yandex.ru).

Дмитрий Евгеньевич Намиот, МГУ им. М.В. Ломоносова, (email: dnamiot@gmail.com).

взаимодействия (например, что делает пользователь в течение сессии — слушает музыку, смотрит видео и т.п.)[8], [9].

Вся доступная информация о сессии называется ее контекстом.

Учет контекста особенно важен для систем, в которых много новых или анонимных пользователей. Из-за отсутствия исторических данных о них важно в полной мере использовать всю доступную информацию, так как она может помочь разделить анонимных пользователей на грубые категории без какой-либо реальной персонализации[8], [10].

Помимо контекста сессии, для построения рекомендаций могут быть использованы следующие данные:

- 1) Информация о пользователе — история покупок пользователя, его предпочтения (в случае неанонимной сессии). Позволяют учитывать долгосрочные предпочтения/потребности пользователей;
- 2) Описание объектов (товаров) или непосредственно сам контент (например, музыка, книги, фильмы);
- 3) Информация о взаимодействии пользователей и объектов — как правило, представлены в виде матрицы предпочтений "пользователь-объект";
- 4) Информация о взаимодействии пользователей — данные из социальных сетей и т.п.

В. Классы рекомендательных систем, использующих данные сессий

Рекомендательные системы, использующие данные сессий подразумевают наличие упорядоченности информации во времени. Так классические подходы на основе матрицы предпочтений позволяют учитывать только однократное взаимодействие пользователя с объектом, тогда как данный подход учитывает многократные взаимодействия в течение одной либо разных сессий.

В зависимости от наличия исторических данных о пользователях (их профилей) и важности их краткосрочных либо долгосрочных предпочтений в контексте задачи, можно выделить два основных класса рекомендательных систем[8]:

- 1) Session-aware Recommender Systems (SARS) — задача подразумевает наличие исторических данных о пользователе в системе. Таким образом, речь идет о постоянных пользователях, которые раз за разом возвращаются. Типичным примером могут быть рекомендации в онлайн-магазинах или магазинах приложений.
- 2) Session-based Recommender Systems (SBRS) — отсутствует информация о профиле пользователя, можно опираться только на анонимные сессии и их метаданные. Примером может быть анонимный просмотр онлайн-магазина, выдача рекламы в интернете.

С точки зрения цели рекомендаций можно выделить следующие классы SBRS[3]:

- 1) Next Interaction Recommendation — предсказание следующего объекта взаимодействия в текущей сессии.
- 2) Next Partial-Session Recommendation — предсказание всех оставшихся объектов/взаимодействий в текущей сессии.

III. Постановка задачи для построения SBRS

Учитывая данные из предыдущих пунктов, формализовать задачу построения рекомендательной системы можно следующим образом. Пусть текущая сессия $s = o_1, o_2, \dots, o_n$ — множество взаимодействий пользователя с объектами. Каждое взаимодействие o_i состоит из объекта $v_j, v_j \in V$ и типа взаимодействия $a_k, a_k \in A$, которое пользователь совершил над данным объектом. S — множество всех возможных действий A пользователей над объектами V .

c — входные данные, контекст сессии. Все возможные контексты сессий формируют множество C .

Пусть f — функция, оценивающая качество рекомендаций l , полученных на основе контекста сессии c . Тогда цель SBRS — найти $\hat{s} \in S$ такой, что:

$$\hat{s} = \operatorname{argmax} f(c, s), c \in C, s \in S.$$

IV. Подходы к построению

Все алгоритмы, применяемые для построения SBRS можно разделить на две основные группы: традиционные подходы и подходы на основе нейронных сетей. Далее каждый из упомянутых классов будет рассмотрен более подробно.

А. Традиционные подходы

Класс традиционных подходов объединяет в себе простые с точки зрения понимания, интерпретации и реализации алгоритмы[3]. Тем не менее, часто данные алгоритмы не уступают в качестве более сложным моделям[11], [12]. Кроме того, возможность интерпретации результатов дает им преимущество по сравнению с более сложными моделями.

1) *Простые ассоциативные правила (Association Rules, AR)*: Суть метода заключается в поиске пар объектов, которые часто встречаются в сессиях вместе. Иными словами, алгоритм находит взаимосвязи вида "если было куплено А, значит, обязательно будет куплено В". Алгоритмически правила и их важность формируются в результате подсчета частоты встречаемости пар различных объектов в сессиях пользователей[11], [13]. Если рассматривать $w_{i,j}$ как "вес" ассоциативного правила для пары объектов v_i и v_j , для текущей сессии s оценка качества рекомендуемого объекта v_i будет выглядеть следующим образом:

$$\operatorname{score}_{AR}(s, v_i) = \max(w_{i,j}), v_j \in s$$

$$w_{i,j} = \sum_{s \in S} \sum_{v_i \in V} \sum_{v_j \in V} \frac{1_{v_i! = v_j} * 1_{v_i \in s} * 1_{v_j \in s}}{|S|}$$

Часто при построении рекомендаций рассматриваются ассоциативные правила только для последнего объекта в сессии.

2) *Простые последовательные правила (Sequential Rules, SR)*: Разновидность ассоциативных правил, подразумевающая упорядоченность объектов в паре. Помимо упорядоченности важное значение имеет их расположение внутри сессии — чем ближе расположены объекты, тем более высока вероятность того, что они образуют последовательное правило. При подсчете "значимости" каждой пары объектов учитывается их совместное вхождение в сессии пользователей и их близость в этих сессиях[11], [13]

$$score_{SR}(s, v_i) = \max(w_{i,j}), v_j \in s$$

$$w_{i,j} = \frac{\sum_{s \in S} \sum_{v_i \in V} \sum_{v_j \in V} \frac{1_{i < j} * 1_{v_i \in s} * 1_{v_j \in s}}{|S| * |i - j|}}$$

3) *Марковские цепи (Markov Chains, MC)*: Последовательность случайных событий с конечным множеством исходов, обладающая свойством — переход от одного события к другому зависит только от текущего состояния процесса, а не от всех предыдущих.

Марковские цепи определяются как множество кортежей $\{ST, P_t, P_0\}$, где ST — множество состояний, P_t — матрица вероятностей переходов, P_0 — начальные вероятности каждого состояния ST [3], [14], [15]. Вероятность перехода между элементами v_i, v_j определяется следующим образом:

$$P_t(i, j) = P(v_i \rightarrow v_j) = \frac{freq(v_i \rightarrow v_j)}{\sum_{v_t} (v_i \rightarrow v_t)}$$

Переход по цепочке событий $(v_i \rightarrow v_j \rightarrow v_k)$ выглядит следующим образом:

$$P(v_i \rightarrow v_j \rightarrow v_k) = P(v_i) * P(v_j | v_i) * P(v_k | v_j, v_i)$$

При построении рекомендательных систем марковские цепи применяются как в базовом виде, так и в комбинации с другими подходами. Например, в [14] представили рекомендательную на основе марковского процесса принятия решения (Markov Decision Process-based RS). Состояния в их системе соответствуют релевантной информации о пользователе, в том числе информации о его последних взаимодействиях с объектами; матрица переходов содержит вероятности перехода к объекту после того, как пользователь провзаимодействовал с некоторой последовательностью объектов (авторы учитывают только последовательности из 5 элементов). Авторы [15] предложили модель PME (Personal Markov Embedding), отображающую объекты и пользователей в евклидово пространство, и далее на основе расстояния между объектами генерируются рекомендации следующего объекта в сессии (следующая песня). Подход, описанный в [16] объединяет в себе марковские цепи и разложение матриц (Matrix Factorization, MF). Метод основан на персонализированных графах переходов по основным цепям Маркова. У каждого пользователя есть собственная матрица перехода, их объединение формирует гиперкуб переходов. Однако данное решение, как и многие другие, не учитывает фактор времени. Это привело к появлению рекомендательной системы на основе марковских цепей (Markov Chain RS, MCRS)[17]. Каждому пользователю, совершившему некоторое действие с объектом, рекомендуются объекты из сессий пользователей, интересовавшихся тем же объектом с учетом фактора времени.

Среди работ, посвященных рекомендательным системам на основе сессий с помощью алгоритмов, основанных на Марковских цепях можно выделить рекомендательные системы на основе контекстных деревьев (они же Марковские модели переменного порядка, Variable-order Markov Models, VMM)[18]. Данная рекомендательная система предназначена для адаптивных последовательных рекомендаций. В основе метода — дерево, которое строится на основе последовательностей объектов из

пользовательских сессий. Для построения рекомендаций осуществляются следующие шаги:

- 1) Сопоставление текущей сессии пользователя с контекстным деревом;
- 2) Расширение контекстного дерева при встрече элементов/последовательностей, не встречавшихся ранее;
- 3) Построение рекомендаций
- 4) *Метод ближайших соседей, K Nearest Neighbours, KNN*: Метрический алгоритм классификации. Класс каждого объекта определяется на основе агрегации (коллаборации) классов его "соседей"(наиболее похожих объектов, пользователей, сессий). Несмотря на свою простоту, данный метод может показывать высокую эффективность. Как правило данный алгоритм не учитывает порядок элементов внутри сессии, однако все зависит от выбранной меры сходства. Возможны следующие подходы к реализации алгоритма:

- 1) *Item-based*. Для контекста сессии c SBRS будет рекомендовать K объектов, наиболее подходящих к контексту c с точки зрения их появления в других сессиях.
- 2) *Session-based*. Для контекста сессии c ищутся наиболее похожие сессии. Затем для каждого кандидата \hat{v} вычисляется оценка

$$score(\hat{v}) = \sum_{s \in Neighbours(c)} similarity(c, s) * 1_s(\hat{v})$$

Для последнего session-based метода ближайших соседей возможны следующие расширения[11]:

- 1) v-sknn (Vector Multiplication Session-based KNN) — действует предположение о том, что последние элементы в сессии более показательны, поэтому каждому объекту сопоставляется вес. В результате при поиске ближайших сессий большее значение имеют последние элементы.
- 2) s-sknn (Sequential Session-based KNN) — также наделяет последние элементы сессии более значительными весами, при этом при вычислении оценки каждого варианта рекомендации учитывается его вхождение в сессии, которые наиболее похожи на текущую:

$$score_{s-sknn}(\hat{v}) = \sum_{s \in Neighbours(c)} similarity(c, s) * w_s(c) * 1_s(\hat{v})$$

- 3) sf-sknn (Sequential Filter Session-based KNN) — рассматривается последний объект, с которым пользователь взаимодействовал в текущей сессии, в качестве рекомендаций могут выступать только те объекты схожих сессий, которые шли в них после этого объекта:

$$score_{sf-sknn}(\hat{v}) = \sum_{s \in Neighbours(c)} similarity(c, s) * 1_s(v_{c-last}, \hat{v})$$

Метод ближайших соседей часто применяется в гибридных рекомендательных системах для улучшения их качества. Так в [19] используется в комбинации с рекуррентной нейронной сетью GRU4REC[20].

5) *Сравнение традиционных подходов*: В таблице 1 представлено сравнение традиционных подходов с точки зрения их достоинств и недостатков. Согласно исследованиям [3], наиболее универсальными, показывающими

высокое качество на различных датасетах является метод ближайших соседей для сессий (session-based KNN).

В. Нейронные сети

Глубокое обучение — совокупность широкого класса методов, основанных на искусственных нейронных сетях. В последние годы эти методы привели к значительному прогрессу во многих областях, например классификации объектов на изображениях, распознаванию речи, генерации текстов. Преимущество глубоких нейронных сетей — возможность улавливать малейшие зависимости в данных, благодаря чему они нашли широкое применение в задачах построения рекомендаций, в том числе на основе сессий.

1) *Рекуррентные нейронные сети (Recurrent Neural Networks, RNN)*: Рекуррентные нейронные сети были разработаны для моделирования последовательностей данных переменной длины [11], [20]. Главная особенность рекуррентных нейронных сетей заключается в наличии внутреннего скрытого состояния внутренних слоев сети. Стандартные RNN обновляют свое скрытое состояние h_t , используя следующую функцию обновления:

$$h_t = g(W * x_t + U * h_{t-1}),$$

где g — гладкая ограниченная функция, x_t — входные данные слоя t .

Проблема базовых RNN — неспособность удерживать в памяти длинных цепочек последовательностей. Это связано с проблемой затухающего/взрывающегося градиента. Решить эту проблему призваны сети долгой краткосрочной памяти (Long Short-Term Memory, LSTM)

Управляемый рекуррентный блок (Gated Recurrent Unit, GRU) [21] — механизм вентилей для рекуррентных нейронных сетей, позволяющий бороться с проблемой затухающего градиента путем анализа изменений скрытого состояния. Внутреннее состояние обновляется согласно следующим формулам [20], [21]:

$$\begin{aligned} h_t &= (1 - z_t) * h_{t-1} + z_t * \hat{h}_t, \\ z_t &= \sigma(W_z * x_t + U_z * h_{t-1}), \\ r_t &= \sigma(W_r * x_t + U_r * h_{t-1}), \\ \hat{h}_t &= \tanh(W * x_t + U * (r_t \odot h_{t-1})), \end{aligned}$$

где z_t — ворота обновления, r_t — ворота сброса.

Каждый блок GRU представляет одно скрытое состояние, а слой, состоящий из последовательности соединенных блоков GRU, образует скрытый слой RNN.

Наиболее известная архитектура RNN для рекомендаций на основе сессий — GRU4REC [20]. Вход сети — это фактическое состояние сеанса, а выход — элемент следующего события в сеансе. Состояние сеанса может быть либо элементом фактического события, либо событиями в сеансе на данный момент. Результатом является распределение вероятностей появления следующих элементов в сессии.

Существуют расширения архитектуры GRU4REC. Так в [22], [23] представлена архитектура NARM. Ее особенность — гибридный энкодер (кодировщик) с механизмом

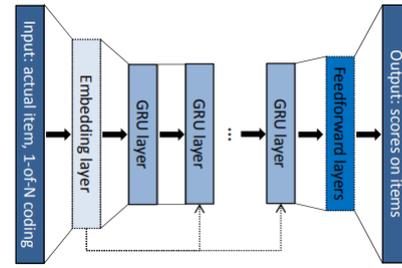


Рис. 1. Архитектура GRU4REC [20]

внимания для моделирования последовательного поведения пользователей и фиксации их основного намерения в текущей сессии.

В [24] представлен метод, расширяющий SBRS на основе RNN до SARS за счет добавления второй "межсеансовой" RNN, отвечающей за связи между сессиями пользователя. Внутрисеансовый RNN такой же, как и в модели GRU4REC. Межсеансовая RNN учится на последних сеансах пользователя и передает информацию во внутрисеансовую RNN. В начале каждого сеанса окончательный вывод межсеансовой RNN инициализирует скрытое состояние внутрисеансовый RNN.

2) *Графовые нейронные сети*: Несмотря на то, что подход к построению рекомендательных систем на основе RNN довольно популярен, он позволяет учитывать только односторонние переходы между объектами в сессии, так как каждая сессия представляется как последовательность объектов. Графовые нейронные сети (GNN) способны улавливать более сложные зависимости между объектами в сессии благодаря ее представлению в качестве ориентированного графа.

Структура графов позволяет хранить информацию о различных взаимосвязях между пользователями и объектами. GNN итеративно агрегирует информацию о соседях каждой вершины и обновляет ее [25].

В зависимости от того, какую информацию и как мы представляем в виде графа, можно выделить следующие типы графовых нейронных сетей для рекомендаций:

- 1) GNN на информации о взаимодействиях пользователей и объектов — информация представляется в виде двудольного графа, первая часть вершин представляет пользователей, вторая — объекты.
- 2) GNN, дополненные информацией о социальных взаимосвязях — помимо графа с информацией о взаимодействиях пользователей и объектов имеется граф социальных взаимодействий, характеризующих взаимоотношения пользователей между собой. Другой вариант — имеется единственный гетерогенный граф, одновременно содержащий всю описанную ранее информацию.
- 3) GNN, дополненные информацией из базы знаний — дополнительно известна некоторая информация об объектах, представлена либо в виде отдельного графа, либо как отдельный тип ребер в гетерогенном графе. Обычно используется в рекомендациях, которые связывают объекты на основе их некоторых атрибутов.

Одной из первых моделей GNN, показавших хорошее качество относительно прочих нейронных архитектур и традиционных подходов стала SR-GNN [26]. Графовые

Таблица I
Сравнение традиционных подходов для SBRS

Метод	Достоинства	Недостатки
Associated Rulers	интерпретируемость, простота метода	учитывает только простые парные зависимости, не подходит для работы со сложно структурированными, упорядоченными данными
Sequential Rulers	интерпретируемость, простота метода	учитывает только простые парные упорядоченные зависимости, не подходит для работы со сложно структурированными данными
Markov Chains	хорошо работает для предсказания простых жестко-упорядоченных последовательностей (последовательные рекомендации)	сильное предположение об упорядоченности объектов, концентрация на зависимости находящихся рядом объектов, плохо улавливает связи между не находящимися рядом объектами
K Nearest Neighbours	простота, интерпретируемость результатов	необходимость подбора количества соседей, значительное увеличение сложности алгоритма с увеличением объема выборки, не улавливает внутренних зависимостей сессий

нейронные сети также используются совместно с другими подходами для построения более точных рекомендаций. Так в [27] Graph Contextualized Self-Attention Network (GC-SAN) объединяет GNN и механизм self-attention, также как и в Full Graph Neural Network (FGNN)[28], [29] и Shortcut Graph Attention (SGAT)[30].

3) *Механизм внимания (Attention Mechanism, AM)*: Техника, применяемая в нейросетях для выявления взаимосвязей между различными частями входных и выходных данных. Важная особенность — способность выделять ключевую информацию в потоке данных, благодаря данному механизму получил широкое применение в задачах обработки текстов, в том числе задачах машинного перевода. Механизм внимания позволяет моделировать зависимости между объектами вне зависимости от расстояния между ними[31], [32], [33].

Частный случай механизма внимания — self-attention — применяется для выявления закономерностей исключительно внутри входных данных. Как правило данный механизм используется в качестве одного из слоев в нейронных сетях (например, рекуррентных), однако существует архитектура нейронных сетей, основывающаяся исключительно на механизме self-attention — Transformer. В отличие от RNN, данная архитектура не требует обязательной пошаговой обработки последовательности, благодаря чему модель проще распараллеливать, и обучение модели проходит быстрее[34].

Базовая архитектура состоит из последовательностей энкодеров и декодеров (encoder-decoder model), параметризована матрицами весов W_Q, W_K и W_V (queries, keys, values). Для вычисления значения функции внимания вектора X вычисляются значения Q, K, V путем умножения соответствующих матриц весов на вектор с входными данными, далее:

$$Attention(Q, K, V) = softmax\left(\frac{KQ^T}{\sqrt{d_k}}\right)V,$$

где d_k — размерность входного вектора.

4) *Обучение с подкреплением (Reinforcement Learning, RL)*: Обучение с подкреплением (Reinforcement Learning, RL) — метод машинного обучения, целью которого является максимизация долгосрочного накапливаемого вознаграждения путем обучения агента взаимодействию

с некоторой средой. Алгоритм не требует размеченных данных, обучается соблюдать баланс между получением новых знаний и применением имеющихся[35], [36].

В случае рекомендательных систем для сессий RL моделирует взаимодействие пользователя с рекомендательной системой, некоторый диалог между ними. Существует два подхода к построению RL алгоритмов:

- 1) Model-based — создается модель для симулирования среды, затем агент обучается взаимодействовать со средой. Качество обученного таким образом агента напрямую зависит от качества построенной среды, генерация поведения пользователей все еще считается сложной задачей.
- 2) Model-free — отсутствует модель, симулирующая среду, агент обучается на реальных данных. При этом обучение моделей с помощью взаимодействия с реальными пользователями может оказаться более эффективным с точки зрения качества модели, но несет в себе риски получения негативного пользовательского опыта. Это может привести к отказу пользователей от использования данного сервиса, снижению прибыли.

Взаимодействие с пользователем позволяет агенту корректировать стратегию со временем, пока не будет найдена оптимальная.

Типичные работы по SBRS на основе RL — включают List-wise Recommendation framework based on Deep reinforcement learning (LIRD)[37] и аналогичная работа под названием DeepPage для постраничных рекомендаций[38].

C. Сравнение нейросетевых подходов

В таблице II представлено сравнение основных нейросетевых подходов для SBRS.

V. Фреймворки для построения SBRS

A. Session-rec

Фреймворк, разработанный для проведения экспериментов по построению и обучению рекомендательных систем для сессий[39]. Был разработан для проведения экспериментов, связанных с построением рекомендательных систем на основе сессий.

Таблица II
Сравнение нейросетевых подходов к построению SBRS

Метод	Достоинства	Недостатки
RNN	позволяет моделировать как долгосрочные, так и краткосрочные (моментальные) предпочтения пользователей	предположение о строгой упорядоченности данных
GNN	позволяет учитывать сложные взаимосвязи в данных, в том числе социальные связи пользователей, подходит для сессий со сложной структурой взаимосвязей объектов и пользователей	высокая алгоритмическая сложность, моделировать сложных переходов между объектами в одной сессии, способен обрабатывать гетерогенные графовые данные
Attention	выделение важной информации, игнорирование шумовых объектов в сессии, возможность распараллеливания, ускоренное обучение модели,	не учитывает информацию о порядке элементов в сессии
RL	построение интерактивных моделей, способных максимизировать долговременное вознаграждение	трудное обучение модели, предположение о том, что модель действует в Марковском мире (каждое следующее действие зависит только от текущего)

Запуск экспериментов осуществляется с помощью файлов с настройками. Доступно несколько режимов запуска, есть возможность выбора алгоритма, метрики для оценивания качества, типа запуска (однократное или многократное разбиение выборки на обучающую и тестовую).

Реализовано более двадцати алгоритмов построения рекомендаций для сессий, для удобства они разбиты на пять групп[39]:

- 1) Базовые алгоритмы — простейшие рекомендательные системы, качество работы которых используется для сравнения при построении более сложных систем. Включают в себя ассоциативные правила, марковские цепи, последовательные правила, байесовский персонализированный рейтинг и контекстные деревья.
- 2) Метод ближайших соседей — алгоритмы, основанные на поиске решения среди схожих объектов/пользователей/сессий. Фреймворк содержит пять различных реализаций данного подхода.
- 3) Нейросети на основе анонимных сессий — методы глубокого обучения для построения рекомендаций, включают в себя RNN, GNN, нейронные сети с механизмом self-attention.
- 4) Нейросети на основе неанонимных сессий — в отличие от моделей из предыдущего пункта, способны моделировать длительные предпочтения, так как известна информация о пользователе. В основе те же алгоритмы, что и в случае анонимных сессий.
- 5) Факторные методы — включает в себя факторные персонализированные марковские цепи,

Данный фреймворк открывает доступ к большому количеству реализаций лучших наработок в построении рекомендательных систем для сессий. Несмотря на это, фреймворк имеет особенности, затрудняющие работу с ним: отсутствие полноценной документации и общей архитектуры реализованных алгоритмов.

B. Transformers4Rec

Фреймворк для построения рекомендательных систем для сессий от компании NVIDIA, в основе которого — применение техник NLP (Natural Language Processing) и архитектуры Transformer для рекомендаций[40].

Transformers4Rec был разработан на основе открытой библиотеке HF Transformers, содержащей эффективные реализации различных архитектур Transformers и позволяющей стоять как модели для исследований, так и для применения в реальных бизнес-задачах. Transformers4Rec дополняет языковые модели, позволяя разработчикам напрямую использовать последние исследования в сфере NLP и последние архитектуры Transformers для экспериментов по построению рекомендаций на основе сессий.

Фреймворк позволяет проводить все этапы построения моделей, начиная с предобработки данных и заканчивая получением результатов (предсказаний).

Предобработка данных — узкое место в процессе разработки рекомендательных систем. Библиотека NVTabular, полностью интегрирована для совместного использования с Transformers4Rec, позволяет обрабатывать терабайты данных сессий, также возможно ускорение обработки с помощью вычислений на GPU. Предварительно обработанные данные сохраняются в структурированном формате Parquet. Transformers4Rec использует загрузчик данных NVTabular, который считывает файлы Parquet непосредственно в память графического процессора, ускоряя обучение и оценку модели. Для получения информации о признаках, на которых будет обучаться модель, их типах и метаданных в качестве входных данных используется конфигурационный файл.

Transformers4Rec наследует от HF Transformers класс Trainer, отвечающий за обучение моделей, и адаптирует его под задачи рекомендательных систем. Мета-архитектура фреймворка предоставляет широкие возможности для построения вычислительных графов функций с возможностью настройки каждого этапа. Оценка качества рекомендаций осуществляется с помощью метрик ранжирования TOP-N.

C. FlowRec

FlowRec — расширение фреймворка Scikit-Multiflow, предназначенного для потокового интеллектуального анализа данных[41].

Фреймворк состоит из трех основных объектов: потока (stream), оценщика (evaluator) и модели. В каждый момент времени t поток предоставляет набор данных (X, y) , где X — вектор признаков объекта y . Далее модель

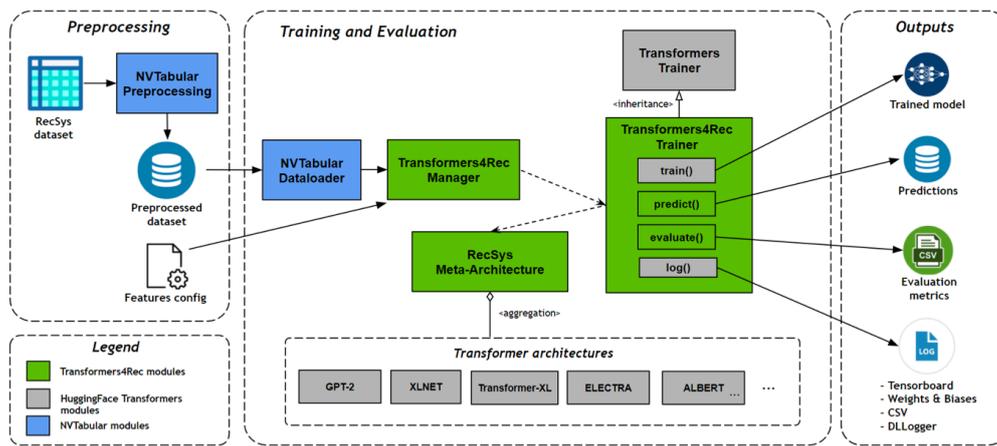


Рис. 2. Transformers4Rec [40]

пытается предсказать набор элементов, которые соответствуют признаковому описанию X . После получения предсказания происходит оценка текущего качества рекомендательной системы и ее дальнейшее обучение. Данные (X, y) добавляются к скользящему окну из n последних событий в сессии. Размер окна выбирается с учетом доступной памяти и временных ограничений.

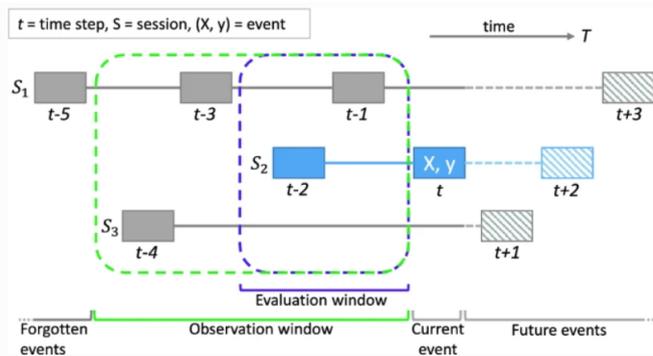


Рис. 3. FlowRec[41]

FlowRec поддерживает несколько моделей для построения рекомендательных систем на основе сессий: ассоциативные и последовательные правила, метод ближайших соседей и BEER — ансамбль бандитов, использующий сэмпирование Томпсона для выбора модели, в качестве поведенческих компонентов ансамбля используются модели, основанные на правилах. Для проверки качества рекомендаций реализован baseline — простейшая рекомендательная система, выдающая в качестве рекомендаций объекты с наибольшим рейтингом.

D. Сравнение Фреймворков

В таблице III приведено сравнение описанных выше фреймворков. Рассматриваются реализованные в фреймворках методы, а также достоинства и недостатки.

VI. Заключение

Рекомендательные системы стали неотъемлемой частью повседневной жизни людей; они помогают нам ориентироваться в огромном количестве информации.

Они также стали необычайно важны для поддержания жизнеспособности и конкурентоспособности онлайн-магазинов, сервисов. Для них возможность предсказать интересы пользователя на основе онлайн-сеанса является актуальной проблемой. При этом универсальной модели не существует, для каждого отдельного случая существует наиболее подходящий подход для построения рекомендательной системы. В работе были рассмотрены основные алгоритмы для построения рекомендательных систем на основе сессий, их достоинства и недостатки. Также были рассмотрены три фреймворка, позволяющих строить модели рекомендательных систем и проводить эксперименты без лишних затрат на написание базовых функций.

Статья подготовлена в рамках работы над магистерской диссертацией. Направление работ относится к проекту кафедры ИБ факультета ВМК МГУ имени М.В. Ломоносова, посвященному устойчивым моделям машинного обучения[42]. Обзор моделей рекомендательных систем для сессий также приводится в работе[43].

Таблица III
Сравнение фреймворков для SBRS

Фреймворк	Реализованные методы	Достоинства	Недостатки
Session-rec	метод ближайших соседей, нейросетевые методы, факторные модели, ассоциативные правила, последовательные правила, марковские цепи, байесовский персонализированный рейтинг, контекстные деревья	большое количество реализованных алгоритмов, функционал для самостоятельного проведения экспериментов по построению рекомендательных систем	отсутствие документации
Transformers4Rec	нейросетевые методы	полная совместимость с другими библиотеками NVIDIA для анализа данных, возможность проведения вычислений на GPU	
FlowRec	метод ближайших соседей, ассоциативные правила, последовательные правила, марковские цепи	построение потоковых рекомендательных систем	узкий класс реализованных алгоритмов

Список литературы

- [1] Business Dictionary for 21st century. URL: <https://businessdictionary.info/>. [Дата обращения — 21.11.2022]
- [2] Çano, Erion; Morisio, Maurizio (2017). Hybrid Recommender Systems: A Systematic Literature Review. In: INTELLIGENT DATA ANALYSIS, vol. 21 n. 6, pp. 1487-1524. - ISSN 1088-467X
- [3] Shoujin Wang, Longbing Cao, Yan Wang, Quan Z. Sheng, Mehmet A. Orgun, and Defu Lian. 2021. A Survey on Session-based Recommender Systems. ACM Comput. Surv. 9, 4, Article 39 (May 2021), 39 pages.
- [4] Shoujin Wang, Gabriella Pasi, Liang Hu, and Longbing Cao. 2020. The era of intelligent recommendation: editorial on intelligent recommendation with advanced AI and learning. IEEE Intelligent Systems 35, 5 (2020), 3–6.
- [5] Cloudera Fast Forward Labs Research. Session-based Recommender Systems. URL: <https://session-based-recommenders.fastforwardlabs.com/>. [Дата обращения — 22.04.2022]
- [6] Hidasi, Balázs and Karatzoglou, Alexandros and Baltrunas, Linas and Tikk, Domonkos. (2015). Session-based Recommendations with Recurrent Neural Networks. arXiv preprint arXiv:1511.06939, 2015.
- [7] Ludewig, Malte and Mauro, Noemi and Latifi, Sara and Jannach, Dietmar. (2021). Empirical analysis of session-based recommendation algorithms. User Modeling and User-Adapted Interaction. 31. 1-33. 10.1007/s11257-020-09277-1.
- [8] Massimo Quadrana, Paolo Cremonesi, and Dietmar Jannach. 2018 Sequence-Aware Recommender Systems. ACM Comput. Surv. 1, 1, Article 1 (February 2018), 35 pages.
- [9] G. Adomavicius and A. Tuzhilin. Context-aware recommender systems. In Recommender Systems Handbook, pages 217–253. Springer, 2011.
- [10] F. Garcin, C. Dimitrakakis, and B. Faltings. Personalized news recommendation with context trees. In RecSys '13, pages 105–112, 2013.
- [11] Malte Ludewig and Dietmar Jannach. 2018. Evaluation of session-based recommendation algorithms. UMUAI 28, 4-5 (2018), 331–390.
- [12] Malte Ludewig, Noemi Mauro, and et al. 2019. Performance comparison of neural and non-neural approaches to session-based recommendation. In RecSys. 462–466.
- [13] Kamehkhosh, Iman et al. “A Comparison of Frequent Pattern Techniques and a Deep Learning Method for Session-Based Recommendation.” RecTemp@RecSys (2017).
- [14] Guy Shani, David Heckerman, and Ronen I Brafman. 2005. An MDP-based recommender system. JMLR 6, Sep (2005), 1265–1295.
- [15] Xiang Wu, Qi Liu, Enhong Chen, Liang He, and et al. 2013. Personalized next-song recommendation in online karaokes. In RecSys. ACM, 137–140.
- [16] Rendle, Steffen and Freudenthaler, Christoph and Schmidt-Thieme, Lars. (2010). Factorizing personalized Markov chains for next-basket recommendation. Proceedings of the 19th International Conference on World Wide Web, WWW '10. 811-820. 10.1145/1772690.1772773.
- [17] Ahmed, Ahmed and Salim, Naomie. (2016). Markov Chain Recommendation System (MCRS). International Journal of Novel Research in Computer Science and Software Engineering. 3. 11-26.
- [18] Mi, Fei and Faltings, Boi. (2018). Context Tree for Adaptive Session-based Recommendation.
- [19] Dietmar Jannach and Malte Ludewig. 2017. When recurrent neural networks meet the neighborhood for session-based recommendation. In RecSys. ACM, 306–310.
- [20] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2016. Session-based recommendations with recurrent neural networks. In ICLR. 1–10.
- [21] Cho, Kyunghyun, van Merriënboer, Bart, Bahdanau, Dzmitry, and Bengio, Yoshua. On the proper-ties of neural machine translation: Encoder-decoder approaches. arXiv preprint arXiv:1409.1259, 2014.
- [22] Latifi, Sara and Mauro, Noemi and Jannach, Dietmar. (2021). Session-aware Recommendation: A Surprising Quest for the State-of-the-art. Information Sciences. 573. 10.1016/j.ins.2021.05.048.
- [23] Jing Li, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Tao Lian, and Jun Ma. Neural attentive session-based recommendation. In In Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, CIKM '17, pages 1419–1428, 2017.
- [24] Massimiliano Ruocco, Ole Steinar Lillestøl Skrede, and Helge Langseth. Inter-session modeling for session-based recommendation. In Proceedings of the 2nd Workshop on Deep Learning for Recommender Systems, DLRS 2017, page 24–31, 2017.
- [25] Yan Wen, Shitao Kang, Qingtian Zeng, Hua Duan, Xin Chen, Wenkai Li, “Session-Based Recommendation with GNN and Time-Aware Memory Network Mobile Information Systems, vol. 2022, Article ID 1879367, 12 pages, 2022.
- [26] Shu Wu, Yuyuan Tang, Yanqiao Zhu, and et al. 2019. Session-based recommendation with graph neural networks. In AAAI. 346–353.
- [27] Chengfeng Xu, Pengpeng Zhao, Yanchi Liu, and et al. 2019. Graph contextualized self-attention network for sessionbased recommendation. In IJCAI. 3940–3946
- [28] 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.
- [29] 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.
- [30] Tianwen Chen and Raymond Chi-Wing Wong. 2020. Handling information loss of graph neural networks for session-based recommendation. In SIGKDD. 1172–1180.
- [31] Sun, Shiming Dai, Zemei Tang, Yuanhe Zhou, Fu. (2019). Self-Attention Network for Session-based Recommendation. IEEE Access. PP. 1-1. 10.1109/ACCESS.2019.2931945.
- [32] Fang, Jun. (2021). Session-based Recommendation with Self-Attention Networks.
- [33] Luo, Anjing Zhao, Pengpeng Liu, Yanchi Zhuang, Fuzhen Wang, Deqing Xu, Jiajie Fang, Junhua Sheng, Victor. (2020). Collaborative Self-Attention Network for Session-based Recommendation. 2563-2569. 10.24963/ijcai.2020/355.
- [34] Vaswani, Ashish Shazeer, Noam Parmar, Niki Uszkoreit, Jakob Jones, Llion Gomez, Aidan Kaiser, Lukasz Polosukhin, Illia. (2017). Attention Is All You Need.
- [35] Garg, Diksha Gupta, Priyanka Malhotra, Pankaj Vig, Lovkesh Shroff, Gautam. (2020). Batch-Constrained Distributional Reinforcement Learning for Session-based Recommendation.
- [36] Karatzoglou, Alexandros Arapakis, Ioannis Jose, Joemon Xin, Xin. (2020). Self-Supervised Reinforcement Learning for Recommender Systems. 10.1145/3397271.3401147.

- [37] 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).
- [38] 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.
- [39] GitHub. session-rec. URL: <https://github.com/rn5l/session-rec>. [Дана обращения — 22.04.2022]
- [40] Moreira, Gabriel de Souza Pereira, Sara Rabhi, Jeong Min Lee, Ronay Ak and Even Oldridge. “Transformers4Rec: Bridging the Gap between NLP and Sequential / Session-Based Recommendation.” Fifteenth ACM Conference on Recommender Systems (2021).
- [41] Paraschakis, Dimitris and Nilsson, Bengt. (2020). FlowRec: Prototyping Session-Based Recommender Systems in Streaming Mode. 10.1007/978-3-030-47426-3_6.
- [42] 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.
- [43] Yakupov, Dmitry, and Dmitry Namiot. “Session-Based Recommender Systems-Models and Tasks.”International Journal of Open Information Technologies 10.7 (2022): 128-155.

Survey On Methods For Building Session-Based Recommender Systems

Marina Ninichuk, Dmitry Namiot

Abstract—Recommender systems currently play a significant role in many areas related to the processing of large amounts of information, such as online stores and cinemas, services for listening to music. As a rule, the construction of recommendations is based on the analysis of information about the user's preferences received in the past. Often such information is presented in the form of a matrix of users-objects. However, in some situations, information about the user may not be available, for example, when visiting the service for the first time or anonymously. A similar statement of the problem is typical for a special class of recommender systems — session-based recommender systems (SBRS). Unlike classical approaches, SBRS learn user's information from current sessions, which makes it possible to obtain information about his rapidly changing preferences. The purpose of recommendations is usually to predict the next object that the user will pay attention to, or the set of such objects in the current session. This article provides an overview of the algorithms used in session-based recommender systems. It also provides an overview and comparison of frameworks for building session-based recommender systems.

Keywords—session-based recommender systems, recommender systems

References

- [1] Business Dictionary for 21st century. URL: <https://businessdictionary.info/>. [Accessed: 21.11.2022]
- [2] Čano, Erion; Morisio, Maurizio (2017). Hybrid Recommender Systems: A Systematic Literature Review. In: INTELLIGENT DATA ANALYSIS, vol. 21 n. 6, pp. 1487-1524. - ISSN 1088-467X
- [3] Shoujin Wang, Longbing Cao, Yan Wang, Quan Z. Sheng, Mehmet A. Orgun, and Defu Lian. 2021. A Survey on Session-based Recommender Systems. ACM Comput. Surv. 9, 4, Article 39 (May 2021), 39 pages.
- [4] Shoujin Wang, Gabriella Pasi, Liang Hu, and Longbing Cao. 2020. The era of intelligent recommendation: editorial on intelligent recommendation with advanced AI and learning. IEEE Intelligent Systems 35, 5 (2020), 3–6.
- [5] Cloudera Fast Forward Labs Research. Session-based Recommender Systems. URL: <https://session-based-recommenders.fastforwardlabs.com/>. [Accessed: 22.04.2022]
- [6] Hidasi, Balázs and Karatzoglou, Alexandros and Baltrunas, Linas and Tikk, Domonkos. (2015). Session-based Recommendations with Recurrent Neural Networks. arXiv preprint arXiv:1511.06939, 2015.
- [7] Ludwig, Malte and Mauro, Noemi and Latifi, Sara and Jannach, Dietmar. (2021). Empirical analysis of session-based recommendation algorithms. User Modeling and User-Adapted Interaction. 31. 1-33. 10.1007/s11257-020-09277-1.
- [8] Massimo Quadrana, Paolo Cremonesi, and Dietmar Jannach. 2018. Sequence-Aware Recommender Systems. ACM Comput. Surv. 1, 1, Article 1 (February 2018), 35 pages.
- [9] G. Adomavicius and A. Tuzhilin. Context-aware recommender systems. In Recommender Systems Handbook, pages 217–253. Springer, 2011.
- [10] F. Garcin, C. Dimitrakakis, and B. Faltings. Personalized news recommendation with context trees. In RecSys '13, pages 105–112, 2013.
- [11] Malte Ludewig and Dietmar Jannach. 2018. Evaluation of session-based recommendation algorithms. UMUAI 28, 4-5 (2018), 331–390.
- [12] Malte Ludewig, Noemi Mauro, and et al. 2019. Performance comparison of neural and non-neural approaches to session-based recommendation. In RecSys. 462–466.
- [13] Kamehkhosh, Iman et al. “A Comparison of Frequent Pattern Techniques and a Deep Learning Method for Session-Based Recommendation.” RecTemp@RecSys (2017).
- [14] Guy Shani, David Heckerman, and Ronen I Brafman. 2005. An MDP-based recommender system. JMLR 6, Sep (2005), 1265–1295.
- [15] Xiang Wu, Qi Liu, Enhong Chen, Liang He, and et al. 2013. Personalized next-song recommendation in online karaokes. In RecSys. ACM, 137–140.
- [16] Rendle, Steffen and Freudenthaler, Christoph and Schmidt-Thieme, Lars. (2010). Factorizing personalized Markov chains for next-basket recommendation. Proceedings of the 19th International Conference on World Wide Web, WWW '10. 811-820. 10.1145/1772690.1772773.
- [17] Ahmed, Ahmed and Salim, Naomie. (2016). Markov Chain Recommendation System (MCRS). International Journal of Novel Research in Computer Science and Software Engineering. 3. 11-26.
- [18] Mi, Fei and Faltings, Boi. (2018). Context Tree for Adaptive Session-based Recommendation.
- [19] Dietmar Jannach and Malte Ludewig. 2017. When recurrent neural networks meet the neighborhood for session-based recommendation. In RecSys. ACM, 306–310.
- [20] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2016. Session-based recommendations with recurrent neural networks. In ICLR. 1–10.
- [21] Cho, Kyunghyun, van Merriënboer, Bart, Bahdanau, Dzmitry, and Bengio, Yoshua. On the proper-ties of neural machine translation: Encoder-decoder approaches. arXiv preprint arXiv:1409.1259, 2014.
- [22] Latifi, Sara and Mauro, Noemi and Jannach, Dietmar. (2021). Session-aware Recommendation: A Surprising Quest for the State-of-the-art. Information Sciences. 573. 10.1016/j.ins.2021.05.048.
- [23] Jing Li, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Tao Lian, and Jun Ma. Neural attentive session-based recommendation. In In Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, CIKM '17, pages 1419–1428, 2017.
- [24] Massimiliano Ruocco, Ole Steinar Lillestøl Skrede, and Helge Langseth. Inter-session modeling for session-based recommendation. In Proceedings of the 2nd Workshop on Deep Learning for Recommender Systems, DLRS 2017, page 24–31, 2017.
- [25] Yan Wen, Shitao Kang, Qingtian Zeng, Hua Duan, Xin Chen, Wenkai Li, “Session-Based Recommendation with GNN and Time-Aware Memory Network”, Mobile Information Systems, vol. 2022, Article ID 1879367, 12 pages, 2022.
- [26] Shu Wu, Yuyuan Tang, Yanqiao Zhu, and et al. 2019. Session-based recommendation with graph neural networks. In AAAI. 346–353.
- [27] Chengfeng Xu, Pengpeng Zhao, Yanchi Liu, and et al. 2019. Graph contextualized self-attention network for sessionbased recommendation. In IJCAI. 3940–3946
- [28] 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.
- [29] 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.
- [30] Tianwen Chen and Raymond Chi-Wing Wong. 2020. Handling information loss of graph neural networks for session-based recommendation. In SIGKDD. 1172–1180.
- [31] Sun, Shiming Dai, Zemei Tang, Yuanhe Zhou, Fu. (2019). Self-Attention Network for Session-based Recommendation. IEEE Access. PP. 1-1. 10.1109/ACCESS.2019.2931945.
- [32] Fang, Jun. (2021). Session-based Recommendation with Self-Attention Networks.
- [33] Luo, Anjing Zhao, Pengpeng Liu, Yanchi Zhuang, Fuzhen Wang, Deqing Xu, Jiajie Fang, Junhua Sheng, Victor. (2020). Collaborative Self-Attention Network for Session-based Recommendation. 2563-2569. 10.24963/ijcai.2020/355.

- [34] Vaswani, Ashish Shazeer, Noam Parmar, Niki Uszkoreit, Jakob Jones, Llion Gomez, Aidan Kaiser, Lukasz Polosukhin, Illia. (2017). Attention Is All You Need.
- [35] Garg, Diksha Gupta, Priyanka Malhotra, Pankaj Vig, Lovekesh Shroff, Gautam. (2020). Batch-Constrained Distributional Reinforcement Learning for Session-based Recommendation.
- [36] Karatzoglou, Alexandros Arapakis, Ioannis Jose, Joemon Xin, Xin. (2020). Self-Supervised Reinforcement Learning for Recommender Systems. 10.1145/3397271.3401147.
- [37] 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).
- [38] 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.
- [39] GitHub. session-rec. URL: <https://github.com/rn5l/session-rec>. [Accessed: 22.04.2022]
- [40] Moreira, Gabriel de Souza Pereira, Sara Rabhi, Jeong Min Lee, Ronay Ak and Even Oldridge. "Transformers4Rec: Bridging the Gap between NLP and Sequential / Session-Based Recommendation." Fifteenth ACM Conference on Recommender Systems (2021).
- [41] Paraschakis, Dimitris and Nilsson, Bengt. (2020). FlowRec: Prototyping Session-Based Recommender Systems in Streaming Mode. 10.1007/978-3-030-47426-3_6.
- [42] 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.
- [43] Yakupov, Dmitry, and Dmitry Namiot. "Session-Based Recommender Systems-Models and Tasks." International Journal of Open Information Technologies 10.7 (2022): 128-155.