



# Implicit Feedback Awareness for Session Based Recommendation in E-Commerce

Ramazan Esmeli<sup>1</sup> · Mohamed Bader-El-Den<sup>2</sup> · Hassana Abdullahi<sup>1,4</sup> · David Henderson<sup>3</sup>

Received: 4 March 2022 / Accepted: 25 February 2023 / Published online: 8 April 2023  
© The Author(s) 2023

## Abstract

Information overload is a challenge in e-commerce platforms. E-shoppers may have difficulty selecting the best product from the available options. Recommender systems (RS) can filter relevant products according to user's preferences, interest or observed user behaviours while they browse products on e-commerce platforms. However, collecting users' explicit preferences for the products on these platforms is a difficult process since buyers prefer to rate the products after they use them rather than while they are looking for products. Therefore, to generate next product recommendations in the e-commerce domain, mostly shoppers' click behaviour is taken into consideration. Shoppers could indicate their interest in the products in different ways. Spending more time on a product could imply a different level of user interest than skipping quickly the product or adding basket behaviour could show more intense interest than just browsing. In this study, we investigate the effect of applying the generated explicit ratings on RS by implementing a framework that maps users' implicit feedback into explicit ratings in the e-commerce domain. We conduct computational experiments on well-known RS algorithms using two datasets containing mapped explicit ratings. The results of the experimental analysis indicate that incorporating calculated explicit ratings from users' implicit feedback can help RS models perform better. The results suggest that there is more performance gap between using implicit and explicit ratings when factorisation machine RS model is used.

**Keywords** Implicit feedback · Context awareness · Explicit rating · Recommendation · E-commerce

This article is part of the topical collection "Knowledge Discovery, Knowledge Engineering and Knowledge Management 2021" guest edited by Joaquim Filipe, Ana Fred, Jorge Bernardino and Elio Masciari.

✉ Ramazan Esmeli  
ramazan.esmeli@port.ac.uk

Mohamed Bader-El-Den  
mohamed.bader@port.ac.uk

Hassana Abdullahi  
hassana.abdullahi@port.ac.uk

David Henderson  
David.Henderson@freshrelevance.com

<sup>1</sup> School of Mathematics and Physics, University of Portsmouth, Lion Terrace, Portsmouth PO1 3HF, UK

<sup>2</sup> School of Computing, University of Portsmouth, Lion Terrace, Portsmouth PO1 3HE, UK

<sup>3</sup> Fresh Relevance Ltd., Southampton Science Park, Southampton SO16 7NS, UK

<sup>4</sup> Cardiff School of Management, Cardiff Metropolitan University, Cardiff, UK

## Introduction

Recommender systems (RS) plays an important role in digital marketing and has been widely used in several sectors such as retail, movie, news, music, book and shopping. Effective RS methods improve the user experience [11]. Also, many businesses depend on RS as powerful personalised marketing tools [29] to achieve business goals and boost sales. Thus, several methods have been developed to recommend most relevant items to users [8] including content-based filtering (CBF) [17], collaborative filtering [22] (CF) and hybrid RS [2].

Session-based recommendation systems (SBRS) [9] are one significant sort of RS, with the main goal of predicting what the next item a certain user is likely to see. Most SBRS merely use the current browsing history, i.e. the items the user has visited so far during the session. This is due to the fact that previous browsing/purchase information is not always available and may be irrelevant to the user's present purpose (the user may look for different items in different sessions). Most classical RS methods

are based on user rating. User rating is normally limited to the items the user has purchased/tried, for example if a user gave a high rating for a science fiction books, this rating could be used to recommend another book for this user. However, In online SBRS item rating is not available as users will not explicitly rate the items while they are browsing, but rating takes place after using the purchased item. Therefore, current SBRS [15, 16, 26, 47] focus on the item features and simple implicit ratings which give equal rating for all viewed items. However, in practice, a viewed item is not an accurate indication of interest, and normally users will have a different level of interest in the items they have been viewing so far.

This work extends the previously published conference paper [10] by experimenting on two new datasets that range different time periods with the previously published one to demonstrate the robustness of the suggested interest level mapping strategy. Moreover, this study further expands on the previously released conference paper by incorporating a dataset analysis section as well as a comparison with current relevant publications. The idea of the paper is that the user interaction and behaviour during a session (e.g. duration of item view, basket items, number of repeated item visits) may indicate the expected user-item rating. Therefore, in this paper, we answer the following research questions: can integration of user browsing behaviour into SBRS methods improve the RS performance, i.e. increase the prediction accuracy of the next item the user is likely to view? And how?

To answer these questions, in this paper, we propose a method to estimate a personalised item rating based on the users' behaviour in a given session. As research on the relationship between implicit feedback and explicit feedback shows that there is a meaningful correlation between these two types of feedback [21, 34]. Also, the estimated rating represents the level of the user's interest in the item based on their behavioural and contextual data since several works [20, 32, 40] showed that using estimated ratings can help to improve the RS performance. Moreover, this paper presents a test method to measure the performance of the proposed framework on different sequence length of user-item interactions on different RS algorithms for both datasets (Fresh relevance and Yoochoose Recsys datasets). The major contributions of this study are as follows:

1. We introduce a user behaviour aware model that: (a) analyses users' interest in the sessions (b) utilises derived numerical implicit ratings in RS models for next-click prediction in SBRS.
2. The proposed model is evaluated on two real-world datasets, the Yoochoose dataset from RecSys 2015, and the Fresh relevance dataset from a personalising company in the UK. Experimental results show that UIA SBRS

achieves a good level of performance, and the proposed UIA mechanism plays an important role.

The rest of this paper consists of the following sections: the next section reviews similar works to this work. The following section addresses the overall proposed framework. Computational experiments and results are presented in the next section. Finally, the last section concluding remarks and future direction are reported.

## Background and Related Works

The purpose of this work is to explore the relationship between one class and estimated numerical implicit rating as a context factor in SBRS. This is implemented using FR and Item–Item similarity CF models. Also, the RS models are analysed in terms of the effect of a user's past interaction length by running on two e-commerce dataset. Thus in this section, we will give a brief description of general RS models, FR and Item–Item similarity CF models, sequence aware RS and feedback types.

## RS Types and Methods

In [23], RS are generally classified into three categories, namely; CF, CBF and hybrid RS. Each of these methods has advantages and drawbacks. For example, CBF RS suffers from serendipity [49], CF RS is hindered from adding a new item or new user [23, 45] in other mean cold start problems and sparsity problems [1, 17] and hybrid RS [5] tries to alleviate the drawbacks of these models. However, Hybrid RS can have disadvantages in terms of resource consumption since these systems combine both models [23].

### Factorisation Recommender (FR) Model

The FR model [7, 37] tries to learn the latent factors for the users, items and side features. The latent factors are used to rank the items for each user in terms of the likelihood that the user may interact with these items. In explicit feedback dataset, latent factors are learned by the user-item interactions and their rating on each interacted item. However, in implicit feedback, latent factors are learned solely based on the interactions. This method works well in CF-based RS for both implicit and explicit feedback datasets. Regardless of feedback type, if the aim is to rank candidate items as a result of prediction at the end of model training, the items are ranked in terms of interaction likelihood by the users. When the dataset only consist of user id, item id and rating, the score for user  $i$  on item  $j$  is calculated as in Eq. 1.

$$\text{score}(i, j) = \mu + w_i + w_j + a^T x_i + b^T y_j + u_i^T v_j \quad (1)$$

In this equation,  $i$  is a user,  $j$  is an item,  $\mu$  is bias term,  $w_i, w_j$  is weight indicator for user  $i$  and item  $j$  respectively,  $x_i, y_j$  are vectors for side data of user  $i$  and item  $j$  respectively,  $a$  and  $b$  are respectively their side features weights,  $u_i, v_j$  is latent factor, which are vectors of length number of latent factors, for user  $i$  and item  $j$  respectively. When side data is not given in the FR model, loss calculation and rating calculation is similar to matrix factorization (MF) [24].

To update the weights vector in the Eq. 2, stochastic gradient descent (SGD) [3] optimization algorithm is commonly used. SGD updates each vector in each iteration by learning rate step.

$$\min_{w, a, b, V, U} \frac{1}{|D|} \sum_{(i, j, r_{ij}) \in D} L(\text{score}(i, j), r_{ij}) + \lambda_1 (\|w\|_2^2 + \|a\|_2^2 + \|b\|_2^2) + \lambda_2 (\|U\|_2^2 + \|V\|_2^2) \quad (2)$$

The Eq. 2 shows the optimisation method of the FR model. In this equation,  $D$  is dataset containing the user id, item id and side data features.  $w$  is the weight term for users and items,  $a, b$  are weight vectors for item side data and user side data respectively.  $U = (u_1, u_2, \dots), V = (v_1, v_2, \dots)$  are latent factors for users and items respectively.  $\lambda_1, \lambda_2$  are regularisation parameters for weight vectors and user-item latent factors respectively.  $\text{score}(i, j)$  is calculated score in each iteration denoted as  $\hat{r}_{ij}$ . Loss function ( $L$ ) in Eq. 3 calculates the difference between the predicted score  $\hat{r}_{ij}$  and actual score  $r_{ij}$ .

$$L(\hat{r}_{ij}, r_{ij}) = (\hat{r}_{ij} - r_{ij})^2 \quad (3)$$

### Item–Item Similarity Collaborative Filtering (Item–Item Similarity CF) Recommendation Model

In Item–Item similarity CF model [25, 43], the similarity between items is calculated by looking at the interacted items of users who have common interacted items. Jaccard and Cosine metrics can be used for the similarity measurement between items [41]. In Jaccard similarity [41], user ratings on items are not taken into account. The idea is to take the average of common users who interacted with both item  $i$  and item  $j$ .

$$J(i, j) = \frac{|U_i \cap U_j|}{|U_i \cup U_j|} \quad (4)$$

In Cosine similarity [41], the ratings on items are considered. We used this similarity measurement for the derived implicit rating dataset.  $CS(i, j)$  is similarity calculation for items  $i$  and  $j$  using Eq. 5.  $U_{ij}$  is users both rated item  $i$  and  $j$ ,  $U_i$  is users rated item  $i$ ,  $U_j$  is users rated item  $j$ ,  $r_{ui}$  is the rating of user  $u$  on item  $i$  and  $r_{uj}$  is the rating of user  $u$  on item  $j$ .

$$CS(i, j) = \frac{\sum_{u \in U_{ij}} r_{ui} r_{uj}}{\sqrt{\sum_{u \in U_i} r_{ui}^2} \sqrt{\sum_{u \in U_j} r_{uj}^2}} \quad (5)$$

Predictions,  $y_{uj}$  are calculated two ways depending on whether rating is specified or not. For example, Eq. 6 is used for prediction when ratings are not presented.

$$y_{uj} = \frac{\sum_{i \in I_u} \text{sim}(i, j)}{|I_u|} \quad (6)$$

---

#### Algorithm 1 SGD Optimisation Algorithm

---

Training data  $D$ , regularization parameters  $\lambda_1, \lambda_2$ , learning rate  $\eta$ , initialization  $\sigma$

Model parameters  $\Theta = (a, b, \mathbf{w}, \mathbf{V}, \mathbf{U})$   $w_0 \leftarrow 0$ ;  $\mathbf{w} \leftarrow (0, \dots, 0)$ ;  $\mathbf{V} \sim \mathcal{N}(0, \sigma)$

**repeat** stopping criterion is not met

$w_i \leftarrow w_i - \eta(\frac{\partial}{\partial w_i} \text{equation } 2)$

$w_j \leftarrow w_j - \eta(\frac{\partial}{\partial w_j} \text{equation } 2)$   $a \leftarrow a - \eta(\frac{\partial}{\partial a} \text{equation } 2)$

$b \leftarrow b - \eta(\frac{\partial}{\partial b} \text{equation } 2)$   $V \leftarrow V - \eta(\frac{\partial}{\partial V} \text{equation } 2)$

$U \leftarrow U - \eta(\frac{\partial}{\partial U} \text{equation } 2)$

---

The difference between FR and MF [24] is that in addition to conventional MF models, FR models learn latent vectors of side data of users and items, however, in MF, only latent factors for users and items are learnt.

$y_{uj}$  is prediction of user  $u$  on item  $j$ ,  $i \in I_u$  is items interacted by user  $u$ ,  $\text{sim}(i, j)$  is similarity between item  $i$  and item  $j$  calculated using Eq. 4.

$$y_{uj} = \frac{\sum_{i \in I_u} \text{sim}(i, j) r_{ui}}{\sum_{i \in I_u} \text{sim}(i, j)} \quad (7)$$

When the ratings are specified in the dataset for example if the given dataset has explicit feedback, Eq. 7 is used to calculate predictions,  $r_{ui}$  is rating given by user  $u$  on item  $i$ ,  $\text{sim}(i, j)$  is calculated using Eqs. 4 or 5. The RS used in Amazon website [25] is a well-known example of Item–Item similarity CF.

## Context Awareness in Session-Based Recommender Systems

The recommendation list can be influenced by the context the user is in. The works [4, 19, 39] investigated the context factor on the performance of the recommendation model. [4] examined position and context awareness of SBRS using deep-learning basing method, the experiments showed 3% of improvement on recall and precision. The experiments showed better recall and precision scores after applying context-awareness. Moreover, [19] investigated the role of discounts, the effects of adopting users' short term intention and popularity trends of the products on RS performance.

## Feedback Types

In the RS domain, there are two types of feedback. These are implicit and explicit feedback.

1. **Implicit feedback:** Implicit feedback is observed behaviours without interrupting users' usage of the system. Namely, a user is not aware while performing implicit feedback obtaining process. This data can be used in RS by interpreting the user interest level [31]. Implicit feedback can be purchase history, read time, number of clicks or session duration.
2. **Explicit feedback:** In this type of feedback, the user explicitly indicates his interest score on a service such as listened to music, watched a movie or any object that user interacted [20].

## Explicit and Implicit Feedback Correlation

In [20], they proposed a work to show the correlation between implicit feedback and explicit feedback. To compare both feedback types, they analysed a music-listening platform *last.fm* dataset.<sup>1</sup> In *last.fm* dataset, like or unlike of a song is explicit preference indicator and number of times

a track played by a user is implicit preference indicator. The authors derived numerical implicit ratings from implicit preference indicators, and they created another dataset to store numerical implicit ratings. They built CF RS models on the two types of feedback. Their results show that different types of feedback complement each other. Also, the models trained on different feedback types showed similar performance to each other.

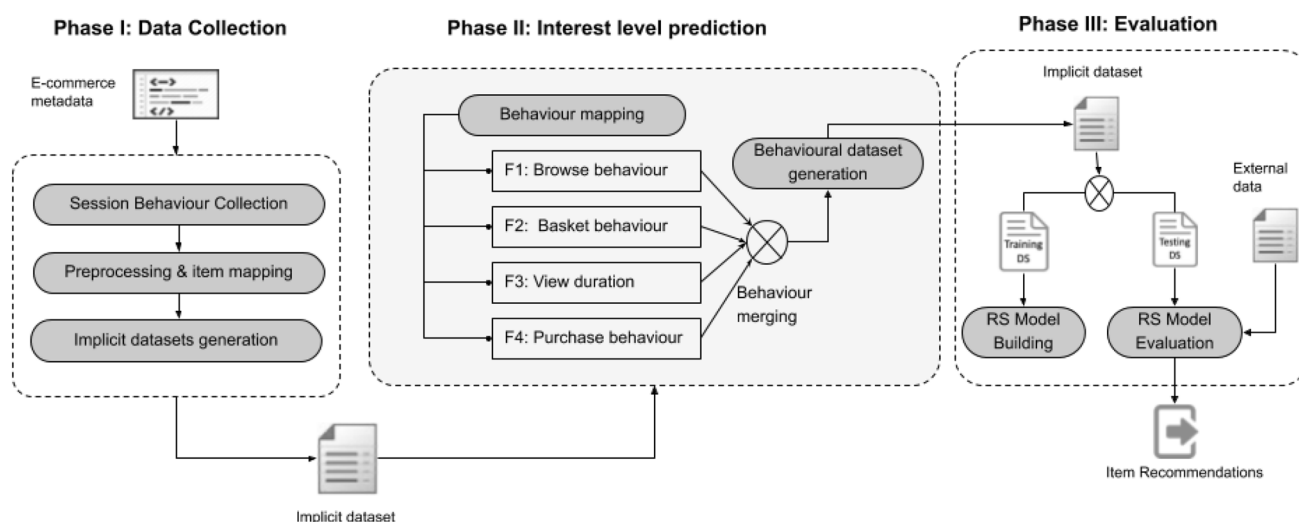
In [40], a method to compare performance measurement of two types of feedback applied to the job domain is designed. They presented a user-user similarity CF approach using only implicit feedback. Before applying their method, they analysed which factors could be strong indicators of user interest to a job. They aim to find which resources better-represent users' interest level and how to represent users' implicit feedback level to the explicit level. Similarly, in [31], user's behaviours on an electronic book domain were captured. The authors converted observed user behaviours into explicit ratings. Their results indicate that user behaviour modelling showed a significant improvement on RS model's performance.

## Limitations of the Previous Approaches

Previous works about converting user behaviours to numerical ratings mainly focused on User-based CF that they investigated already observed user behaviours in the past for only registered users. The limitation of this approach is that User-based CF models cannot produce recommendations when users' rating history is absent [19, 24] since recently, e-commerce websites have become popular. In these e-commerce websites, shoppers can browse items without registering even they can purchase items as a guest, and there is no any user-product rating history. Accordingly, to solve the drawback of User-based CF models for anonymous users in e-commerce platforms, session-based recommender (SBRS) models have been developed where only click behaviours are considered for the next item recommendations [15, 19]. On the other hand, users leave valuable data about their intentions and preferences while browsing the items in the sessions such as duration spent on an item and the number of clicks for an item. One of the limitations of current SBRS models is that user's valuable behavioural indications are ignored, and these models provide next item recommendations solely based on user's click behaviour in the session.

Moreover, as mentioned above, context factors such as price and category of the browsed products in the session are already used for filtering purpose in SBRS. The limitation of this approach is that restricting recommendation models to only filtering based on context factors can cause to losing valuable user preference indicators since not only item and time-based features also user behaviours are strong signals for showing users' interest level on the browsed items in the

<sup>1</sup> <https://archive.ics.uci.edu/ml/datasets/Repeat+Consumption+Matrices>.



**Fig. 1** User interest aware framework (UIA) [10]

sessions. For instance, the minutes user spent on an item while exploring the item, the number of clicks of an item and basket actions (added to cart, browsed only) of the user in the session could be considered as users' interest level indicator on the item. In this paper, we combine all the user activities in the session and create an implicit numerical rating that estimates users' interest level on an item in the on-going session.

## User Interest Aware Framework

As mentioned before, current SBRS is mainly based on implicit item rating, where session viewed items are equally rated in terms of user interest. The proposed algorithm in this paper is motivated by the idea that the user-item interaction in a given session can indicate the level of the user interest in the items viewed so far.

This section presents the user interest aware (UIA) SBRS framework. In this framework, we propose a novel method to predict the user interest (rating) for an item in the given session and use the predicted rating in the RS algorithms (Item-Item CF and FR).

In the proposed UIA framework (Fig. 1), we have created a method to predict users' interest levels on the item by taking into account their implicit feedback and recommended products based on their behaviours in the on-going sessions. The framework consists of three main phases. The first phase is the data collection, data pre-processing and feature selection. The second phase is interest level prediction, which could be seen as a way for converting implicit to explicit rating, and the last phase is utilising the derived ratings on SBRS models.

**Table 1** Dataset statistics used in this work

Dataset	Sessions	Items	Interactions
Fresh relevance	65,417	32,604	1,825,609
Yoochose Recsys	180,512	35,233	3,167,484

## Phase 1: Data Collection, Preparation and Analysis

This phase consists of data collection from the company, data preparation and dataset analysing steps. In the data preparation step, we apply label encoding<sup>2</sup> to categorical IDs, and we refine items which are viewed only one time in the whole dataset and some sessions consist of one viewed item, in which they do not provide enough information to build connections with other items and the sessions. We use two datasets in these work. The first dataset is the Fresh relevance dataset. This dataset covers for a two weeks period from a real-world e-commerce website.<sup>3</sup> The second dataset is the Yoochose RecSys dataset<sup>4</sup> which stores click events from an e-commerce website and covers one month of period. Table 1 shows the statistics about the number of unique items, sessions and total interactions in each dataset. In this section, we analyse the details of our datasets. As seen from Table 1, the dataset density is very low. Thus, RS models are effected from cold-start sessions as mostly, in the sessions a few items are browsed.

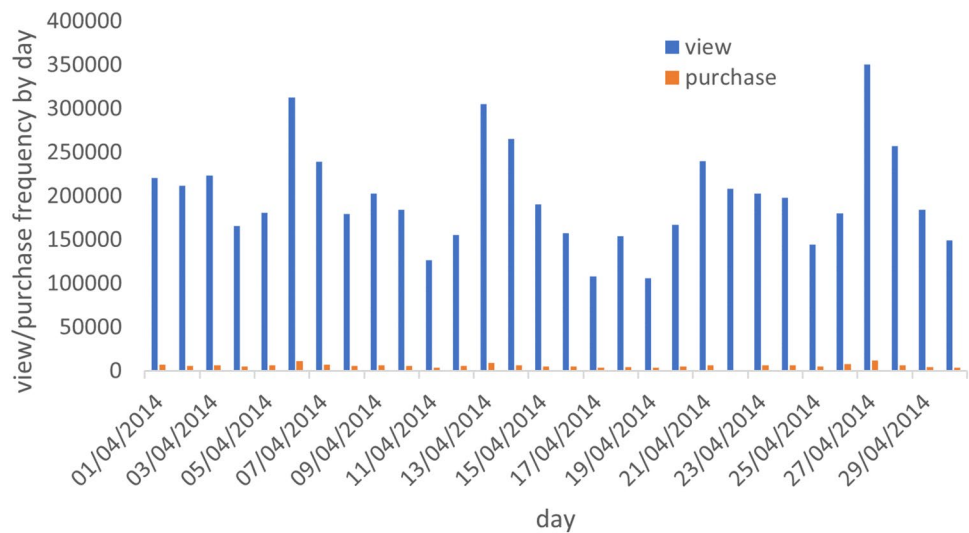
<sup>2</sup> <https://scikit-learn.org/>.

<sup>3</sup> <https://www.freshrelevance.com>.

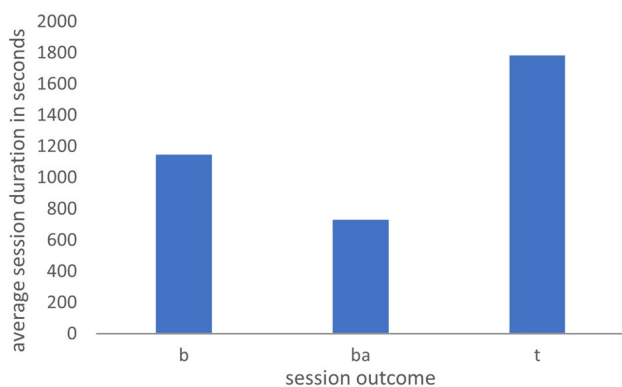
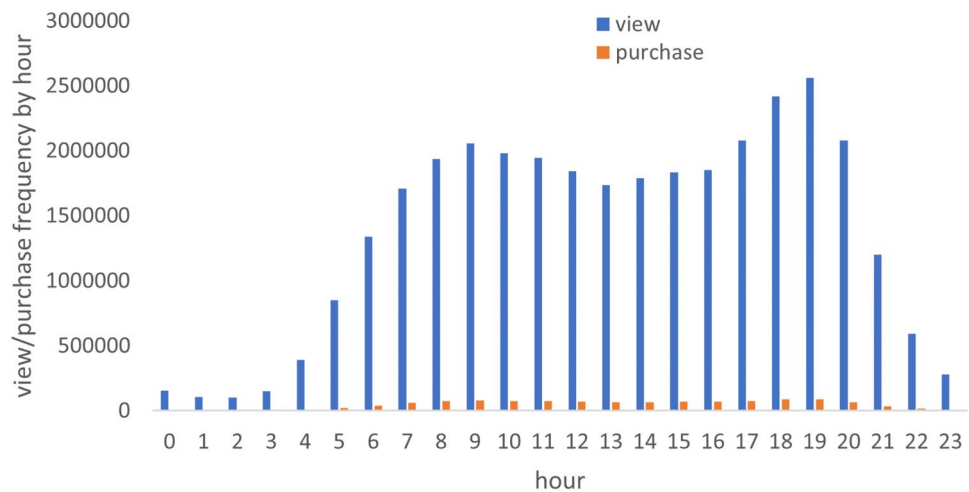
<sup>4</sup> <https://2015.recsyschallenge.com/challenge.html>.



**Fig. 2** Outcome analyses by days for one month in Yoochose RecSys dataset



**Fig. 3** Outcome analyses by hours in Yoochose RecSys dataset

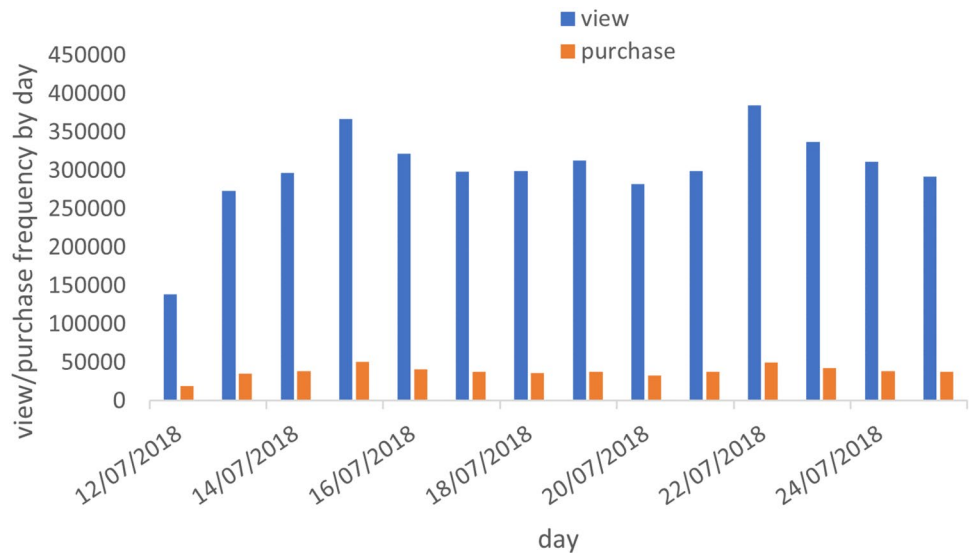
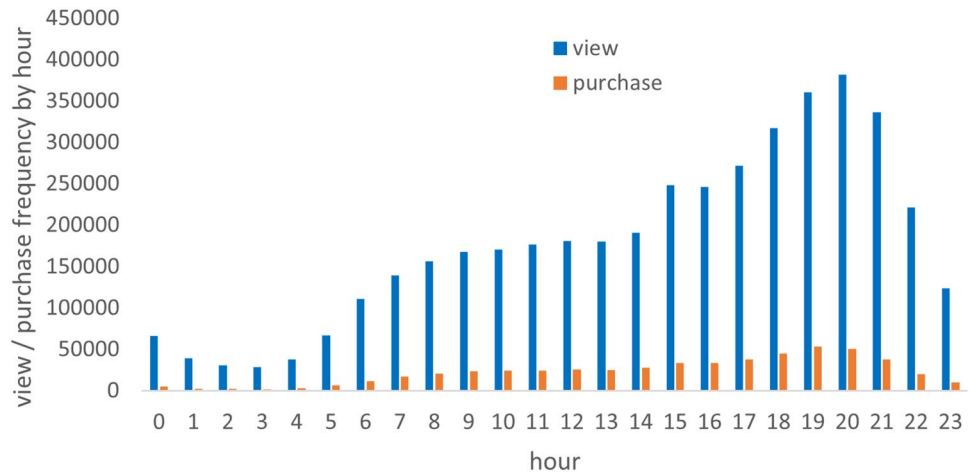


**Fig. 4** Relation of basket outcome and session duration

### Yoochose RecSys Dataset

Figure 2 shows the analysis of user interaction for 1 month. It can be seen in Fig. 2 that users likely choose to visit the website on weekends or closer days to weekends such as Fridays and Mondays. Also, users prefer to visit the website after working hours around 19:00, as seen in Fig. 3.

We examine the item frequencies in sessions. We find that most sessions include two items interactions which is very low number for RS to create correlations between items and sessions due to sparsity and cold start challenges. Thus, we eliminate some sessions which have less than 10 items interactions. After the pre-processing stage, we have 35,233 item, 180,512 sessions and 3,167,484 total interactions. Before

**Fig. 5** Day outcome frequency**Fig. 6** Session outcome distribution in Fresh relevance dataset

pre-processing, the dataset has 50,345 item, 8,324,658 sessions and 29,567,481 interactions.

### Fresh Relevance Dataset

We analyse Fresh relevance dataset in terms of the relation between basket outcome and session duration, as seen in Fig. 4. *ba* the sessions end up browsing only, *b* the sessions end up with adding to basket, *t* the sessions end up with the purchase.

It can be seen from Fig. 4 that if the outcome of the session is purchasing, the total duration is longer than other types of outcome. Also, as expected if users' intention solely to browse products, the session has the shortest duration among the other types of outcome.

Moreover, we look at the daily user interaction frequency with the session outcome to find if there is any correlation between weekdays and user interactions (Fig 5).

Interestingly, it seems from the analyses of the result that users are keen on buying or browsing products on weekends, especially on Sunday. In contrast, in the middle of the week, they are less likely to visit the website to purchase or browse items. Also, we analyse the most interacted hour in the days and Fig. 6 shows that users are more likely to visit the website after working hours.

We look at the session-item interaction frequency for each session. Our dataset analysis results show that more than 450,000 sessions have viewed only one item which means that RS algorithms will suffer from cold start problem and sparsity problem since there is not enough data to establish similarity relation between sessions and items as in Yoochose dataset. To alleviate sparsity and cold start drawbacks for RS models, we delete sessions which have less than 10 item interactions.

**Table 2** Most common actions that define the users' behaviour in an e-commerce platform

ID	Name	Weight
F1	Browse behaviour	w1
F2	Basket behaviour	w2
F3	View duration	w3
F4	Purchase behaviour	w4

## Phase 2: Interest Level Prediction

In this phase, we analyse the users' behaviours and their contributions to calculate the final users' interest level on items.

A user can be directed to a website from different sources, for example, from Google search or an advertisement link shown on a website. The first item that the user look for can be considered as the most relevant item for the user initial intention. After visiting a product, the user will get recommendations based on the item's content or other users' tastes. The point in RS is to get user attention to visit items in the recommendation list. If recommended items are interesting for the user, he/she will click and will look at the detail of the suggested product. If the user is happy with the item user browsed, user can add this item to cart. Otherwise, the user will keep searching until he finds his favoured items or user will leave the system. Sometimes, the user can have some uncertainties about buying products added to cart. In that case, the user will not proceed to purchase the item added to cart, or the user can give up browsing products and leave the system.

### Simple Implicit Feedback

Simple implicit feedback can be considered as positive feedback if a user views an item. Since we do not have explicitly given ratings by users, we have two indicators  $U_{ui}$  for the simple implicit feedback, in which if the user  $ueU$  interacted with the product  $ieI$  or not in a session  $seS$ . For the proposed framework, we used Eq. 8 for the simple implicit feedback (one class) rating representation.

$$f(x) = \begin{cases} 1, & \text{if } U_{ui} \geq 1 \\ 0, & \text{otherwise} \end{cases} \quad (8)$$

For any interactions, regardless users' basket outcome, purchasing behaviour or click behaviour, if there is an interaction with a product, this can be considered as positive feedback otherwise 0, means a user has not seen the items yet. Also, as mentioned in [40], implicit feedback is a relative indication that shows if a user likes an item or not. However, having an interaction on an item can be assumed a minimum interest level [33, 46].

## Behaviour Mapping

Mapping implicit feedback (behaviour) to numerical implicit rating can help better represent user interest on the items. However, implicit mapping feedback is not trivial work since each different domain has different factors to be considered [32, 40]. The motivation of the UIA framework is to see whether there is an improvement on RS performance by analysing user activities in terms of their different behaviours on e-commerce websites, and deriving users' interest level on items as the numerical implicit rating.

Other researchers [30, 40] proposed methods to convert implicit feedback to numerical implicit rating by giving weights to users' behaviours on an e-book application, and job domain, respectively. We follow a similar way to construct the numerical implicit ratings(interest level), and we define actions a user can have on e-commerce system and their weights (see Table 2). If any of these actions have not appeared in the dataset, their contribution will be 0.

To understand the process of explication converting process, Table 2 is explained in detail. *ID* indicates different behaviours and used in mathematical notation defining stage, *Name* explains the behaviour that the user showed in the system. *Weight* shows the contribution of a given behaviour on the numerical implicit rating conversion process. For example, if a user did not like an item he browsed, he may have the intention to click another item in a minute.

### Mathematical Model to Convert Implicit Feedback to Numerical Value of Implicit Feedback (Interest Level)

We define different mathematical equations to indicate user  $ueU$  interest level on item  $ieI$ . The aim of using mathematical equations is to interpret users' actions to have a numerical value of implicit feedback which we call explicit rating of user behaviour or numerical implicit rating. After having users' explicit rating, they can be utilised in different RS methods to analyse explicitly modelled user-item interactions. Our final rating score will be between 0 and 4, which means 0 shows that the user has not interacted with item yet and 4 means item took user's attraction at the highest level.

As mentioned in [20], each domain has different implicit feedback modelling method, even for similar domains but in different e-commerce applications, the interpretation method for the implicit feedback changes. Thus, we may have different weights and their contributions for final explicit rating calculation for each dataset.

*F1*: This indicates the click count contribution to the numerical implicit rating.



This indicates the click count contribution to the numerical implicit rating for each item for a session. To have a normalised value for this indicator, we formulate the calculation of this indicators contribution in Eq. 9. In this equation  $t_c$  shows total click count in a session  $s$ , and  $c_i$  shows the click count for the item  $i$  in the session  $s$ . In this equation, we will get a value between 0 and 1 as item's click contribution to implicit rating based on total click and item's click in the session

$$F1(i, s) = \frac{c_i}{t_c}. \quad (9)$$

$F2$  : indicates level, in which if item  $i$  is added to basket in session  $s$  (Eq. 10) The contribution of adding an item to basket shows an interest level for the item but this depends on users' habit. For example, if a user adds more than one item to the basket, the interest level for the each item can be different comparing to adding one item to the basket. Therefore, in the Equation,  $t_a$  shows total number of added items to basket, and  $t_i$  shows how many item  $i$  is added to basket in a session  $s$ . The user's interest contribution of adding to basket for each item is restricted between 0 and 1. This equation is valid only if there is any item is added to the basket in the session ( $t_a > 0$ ).

$$F2 = \frac{t_i}{t_a} \quad (10)$$

Basket outcome has three categories:  $b$  means item only browsed.  $ba$  means item added to the basket but not purchased.  $t$  means item is purchased. We assume item purchasing is strong interest indicator that we calculate its contribution in  $F4$ , adding to the basket is high-interest indicator however it is relatively less than purchasing, and browsing is minimum interest indicator however we already calculated its contribution in click count indicator  $F1$ ; thus we will not give any interest level contribution for browsing the items.

$F3$  : This represents the duration factor. We can think that if a user spends more time on an item, this means the user has more interest level than less time spend. The Eq. 11 is used to calculate the user's interest level on an item using duration factor. In this equation, total session duration represented as  $t_d$ , and  $i_d$  duration spend on an item  $i$  in the session  $s$ . Calculated  $F3$  value as the consequence of duration factor for interest level calculation on an item is in range between 0 and 1

$$F3 = \frac{i_d}{t_d} \quad (11)$$

$F4$ : shows if the item  $i$  is purchased in a session  $s$  or not. This will have an important interest level indicator for user

on an item in the session. It is calculated using Eq. 12. In this equation,  $t_p$  is the number of total purchased item in the session  $s$ , and  $i_p$  is the number that shows how many of item  $i$  is purchased in the session  $s$ . This interest level has a score between 0 and 1 for the each item in a session  $s$ . This implicit factor is valid when at least one item is purchased in the session  $s$  ( $t_p > 0$ ).

$$F4 = \frac{i_p}{t_p} \quad (12)$$

### Final Numerical Implicit Rating Calculation

We use weights for final score calculation, these weights are showing importance levels of the factors. The sum of these weights is equals to 1 (Eq. 13).

$$\sum_{n=1}^4 w_n = 1 \quad (13)$$

After we have numerical equivalents of implicit feedback using factor equivalence of user behaviours, we create the final numerical rating score by applying aggregation of each numerical equivalents with considering their weights (Eq. 14). The best weight combination in this equation learnt by applying a cross validation method, in which in each cross validation the performance of RS models are evaluated and the weights for the best found performance are selected.

$$\text{score} = w1 \times F1 + w2 \times F2 + w3 \times F3 + w4 \times F4. \quad (14)$$

### Phase 3: Evaluation

In this phase, we evaluate the proposed framework with different metrics. First, we split datasets as test and train. For testing, we allocate one month and one day for Yoochoso RecSys dataset and Fresh relevance dataset respectively. Secondly, RS models are trained with two different types of datasets; the dataset consisting of simple implicit feedback, and new dataset with explicit feedback. In the model evaluation step, models are evaluated by giving interacted items in the sessions to trained RS models and getting recommendations from the models. Ground truth items will be hidden, and the recommendations from the model and items in ground truth will be compared to evaluate the performance. Note that we do not use derived numerical implicit rating for the ongoing item interaction but previous items since in practice we cannot utilise numerical implicit rating concurrently, in which after user view next item we can utilise numerical implicit rating for the previous item.

## Experimental Setup and Results

In this section, we explain the experimental setup details, evaluation metrics, evaluation methods lastly, we discuss the results of the experiments.

### Experimental Setup

We use in this work two different RS models which are the FR model and Item–Item similarity CF model. To run the experiments, we use Graphlab machine learning tool.<sup>5</sup>

#### FR Model

If any side data are not presented in the FR model, it acts as a standard MF model. We used two different types of FR. The first one is for implicit feedback which is one class implicit feedback and the second one is for derived numerical implicit rating.

#### Item–Item Similarity CF Model

We use the Item–Item similarity CF model to compare the result of one class implicit rating data and derived numerical implicit rating. For evaluation, Jaccard and Cosine similarity metrics are used in the Item–Item similarity CF model. In the Jaccard similarity metric, only interacted items are important regardless of ratings on items. On the other hand, Cosine similarity takes into account the user ratings on items.

### Evaluation Metrics

In the literature, accuracy, precision, recall and coverage are some metrics used in RS [14]. recall@n (Eq. 15) and precision@n (Eq. 16) have been used widely in the top-n ranked list RS [13, 14, 42]. Since RS can only recommend a few items at a time, users are expecting to see relevant items in the first page. Thus, we prefer recall@n as an evaluation metric to measure the performance of our method on top n recommendations. recall@n metric shows how the model is good to predict the items in ground truth, precision@n describes how our model's recommendations are good to predict items in ground truth. Also, we employ user coverage metric to see the ratio of the number of users that get at least one correct prediction.

**Table 3** Best weights for the user behaviours

Contributing factor	Weight indicator	Value
F1	$w_1$	0.1
F2	$w_2$	0.5
F3	$w_3$	0.2
F4	$w_4$	0.7

$$\text{recall@n} = \frac{|\text{Recommended Items} \cap \text{Ground Truth}|}{|\text{Ground Truth}|} \quad (15)$$

$$\text{prec@n} = \frac{|\text{Recommended Items} \cap \text{Ground Truth}|}{|\text{Recommended Items}|} \quad (16)$$

coverage@n (Eq. 17) describes the ratio of the number of the users retrieved at least one correct recommendation  $U_{mi}$  to all number of the user  $U$  in test data [28].

$$\text{coverage@n} = \frac{|U_{mi}|}{|U|} \quad (17)$$

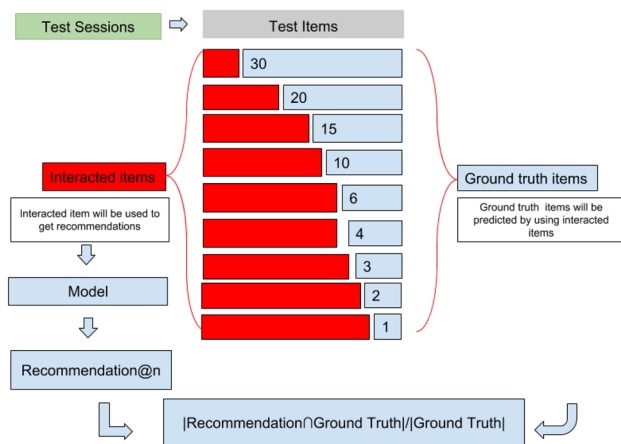
### Dataset Splitting

For dataset splitting, we apply 10 fold cross-validation to have reliable performance results. In each validation loop, we split sessions as train and test. For test dataset, we select 10% of whole sessions in each fold. We do not add any session-item interaction to train dataset from test sessions. In other words, our models are blind to the test sessions. The experiment results show the average values of the 10 fold cross-validation.

### Calculating the Weights

The values of the weights of the behaviours  $w_n, n \in \{1, 2, 3, 4\}$  are decided by the experiments' results. The main approach followed is to assume that, browsing ( $w_1$ ) an item is the weakest level for users' interest indicator. If an item is added to cart ( $w_2$ ), it is presumed that the user has an intention to buy this product and that, thus, he/she has a higher interest level than when just viewing the item. Also, the duration ( $w_3$ ) that the user spent on the item shows an interest level, if it is more than a certain level, as explained in Sect. 3. Lastly, if an item is purchased ( $w_4$ ) in the session, it is taken that the user explicitly indicated he/she liked it and is interested in it. In fact, purchase action has the highest interest rate among the other action types. After experimenting with different weight values by considering the above assumption which inspired by [18], the best ones are identified, as shown

<sup>5</sup> <https://turi.com/products/create/docs/graphlab.toolkits.recommender.html>.



**Fig. 7** Test method followed in sequence aware recommendation [10]

in Table 3. The search space for the weights are restricted between 0 and 1.

## Experiments

We choose two different recommendation models which are Item–Item similarity CF [22] and FR models [36]. We analyse Item–Item similarity CF with two different similarity

measurements, which are Cosine and Jaccard. Cosine similarity is applied to numerical implicit rating data while the Jaccard similarity metric is applied to one-class implicit ratings. In Item–Item similarity CF, 64 most similar items are selected for each item as neighbour since experiment results showed above 64 nearest neighbour does not make a difference in performance.

For the FR model, we select stochastic gradient descent (SGD) [44] as the optimisation method. In Graphlab tool, we can adjust if our dataset consists of implicit feedback or explicit feedback by defining the target attribute. If the model is trained with a target attribute, it means we are using explicit feedback and model will be trained with the standard SGD optimisation method. Otherwise, when the ratings are not available, the ranking will be done by SGD optimiser that SGD will optimise logistic loss function such as observed items is pushed to 1, and the unobserved sample is pushed to 0. In the FR model, since the dimension of latent factors is an important parameter to represent item latent factors and user latent factors, we set this parameter 100 as training FR model is computationally expensive and experiments shows that above 100 for the dimension of the latent factors does not show enough improvement on the performance.

**Table 4** Performance comparison on different length of user interaction of Yoochose RecSys dataset with Item–Item similarity CF model

$G$	rec@5		prec@5		cov@5	
	Implicit	Explicit	Implicit	Explicit	Implicit	Explicit
1	0.2230	0.2200	0.0446	0.0440	0.2230	0.2200
2	0.2075	0.2060	0.0830	0.0824	0.3230	0.3355
3	0.1952	0.1947	0.1171	0.1168	0.3815	0.4165
4	0.1743	0.1848	0.1394	0.1478	0.4195	0.4880
6	0.1428	0.1573	0.1714	0.1887	0.4905	0.5625
10	0.0782	0.0958	0.1563	0.1915	0.4360	0.5125
15	0.0573	0.0737	0.1720	0.2211	0.4449	0.5595
20	0.0481	0.0651	0.1924	0.2605	0.4880	0.6220
30	0.0320	0.0484	0.1920	0.2907	0.5200	0.8933

**Table 5** Performance comparison on different length of user interaction of Yoochose RecSys dataset with FR model

$G$	rec@5		prec@5		cov@5	
	Implicit	Explicit	Implicit	Explicit	Implicit	Explicit
1	0.0000	0.0050	0.0000	0.0010	0.0000	0.0050
2	0.0013	0.0065	0.0005	0.0026	0.0025	0.0130
3	0.0022	0.0075	0.0013	0.0045	0.0065	0.0225
4	0.0025	0.0081	0.0020	0.0065	0.0100	0.0325
6	0.0029	0.0083	0.0035	0.0100	0.0175	0.0490
10	0.0065	0.0094	0.0130	0.0187	0.0600	0.0835
15	0.0056	0.0080	0.0167	0.0241	0.0759	0.1012
20	0.0055	0.0077	0.0220	0.0309	0.0962	0.1271
30	0.0071	0.0049	0.0427	0.0293	0.1733	0.1200

**Table 6** Performance comparison on different length of user interaction of Item–Item similarity CF model on Fresh relevance dataset

$G$	rec@5		prec@5		cov@5	
	Explicit	Implicit	Implicit	Explicit	Implicit	Explicit
1	0.1445	0.1450	0.0257	0.0500	0.1250	0.1285
2	0.1153	0.1125	0.0667	0.0825	0.2323	0.2408
3	0.0965	0.0962	0.0864	0.1241	0.2938	0.3040
4	0.0876	0.0870	0.0999	0.1439	0.3227	0.3299
6	0.0703	0.0713	0.1002	0.1488	0.3070	0.3174
10	0.0450	0.0444	0.1002	0.1398	0.2958	0.3099
15	0.0465	0.0471	0.0825	0.1048	0.2440	0.2491
20	0.0334	0.0339	0.1126	0.1503	0.3671	0.3671
30	0.0298	0.0309	0.1252	0.1553	0.3883	0.3738

**Table 7** Performance comparison on different length of user interaction of FR model on Fresh relevance dataset

$G$	rec@5		prec@5		cov@5	
	Implicit	Explicit	Implicit	Explicit	Implicit	Explicit
1	0.0015	0.0090	0.0028	0.0024	0.0015	0.0090
2	0.0033	0.0073	0.0066	0.0066	0.0065	0.0136
3	0.0045	0.0079	0.0086	0.0088	0.0127	0.0228
4	0.0040	0.0072	0.0106	0.0115	0.0144	0.0272
6	0.0037	0.0065	0.0117	0.0134	0.0193	0.0349
10	0.0032	0.0055	0.0135	0.0145	0.0513	0.0895
15	0.0040	0.0047	0.0144	0.0192	0.0842	0.1082
20	0.0033	0.0035	0.0196	0.0210	0.1189	0.1224
30	0.0037	0.0028	0.0252	0.0194	0.0825	0.0631

For each interacted item(item sequence), we retrieve top@n  $n \in [5]$  recommendation, and we evaluate top@n recommendation with recall@n, precision@n and coverage@n metrics. Length of item sequence changes regarding the length of hidden items to predict. Our aims in this experiment are two-fold. Firstly, we investigate the performance of RS on numerical implicit rating data which is derived from user behaviours and one class implicit rating data. Secondly, we analyse the effect of the sequence length, which has been used as interacted items on RS performance.

The overview of our experiment method for sequence aware recommendation is simulated in Fig. 7. As shown in Fig. 7, at the beginning of a session, the number of the items in ground truth is 30, the items in the ground truth will be predicted by the model base on interacted item(s). For a given interacted item/items in a session, the recommendation outcome is ranked based on the similarity scores of the given item/items. Over time, the length of interacted items increases until the ground truth length reduces to 1.

## Result and Analysis

We report the results of experiments in Tables 4, 5, 6 and 7. In Tables,  $G$  shows the number of items in the ground truth.

As seen in Fig. 7, the aim is recommending accurately these items. For example, if  $G$  is 1, it means that except last interacted item, all previously interacted items in the session are used for getting recommendations, and the target is predicting correctly this hidden item. Also, impl. and exp. indicate evaluation results of the implicit (baseline) and estimated numerical implicit rating, respectively. As users are more likely interested in top items in the recommendation list, we chose recall@n and precision@n evaluation metrics. For model effectiveness, the experiment results are also analysed in terms of coverage@n metric. We compare the proposed framework on Item–Item similarity CF and FR models on each dataset. Tables 4 and 5 show the performance results for the Item–Item CF and FR RS models on Yoochase RecSys dataset, while Tables 6 and 7 show the evaluation results for Item–Item CF and the FR RS models on Fresh relevance dataset.

Tables 5 and 7 show that the FR model has performed better for all evaluation metrics when the models trained on derived numerical ratings. Also, the results of Tables 5 and 7 show that when the FR model knows more interacted items from the sessions, it performed better in terms of recall. Interestingly, when the length of interacted items are decreased (length of ground truth increased), the performance difference between the FR model trained on two

different datasets reduces. The reason for this result could be that when a few items are available in the interacted items, FR model which is trained with derived implicit ratings may not be able to create a correlation between ongoing session and other items. However, after a certain length of interacted items, the FR model trained with derived implicit rating performs better in recall since using ratings derived from user's behaviour through the session may help to create better session-item connections.

Furthermore, results of Tables 4 and 6 indicate that when the Item–Item similarity CF model know more item interactions about the sessions, similar to the FR model, the Item–Item similarity CF model's performance has improved in both rating type (one class rating, derived numerical rating). Nevertheless, the performance difference of the models trained on the different type of ratings for Fresh relevance dataset does not show constant superiority.

Moreover, the coverage rate of the Item–Item similarity CF model for both datasets has performed better than the FR model. Also, the models trained on derived implicit ratings showed robust coverage rates compared to one class rating dataset. This result has supported that taking into consideration users' actions in the sessions may help create well session-item correlations.

Overall, we can confirm from the results that when the model knows more interaction about an ongoing session, it performs better in terms of recall metric. Also, when we train the models with derived ratings, the models have better performance on all metrics due to taking into account users' preferences on the items in the sessions. Lastly, the overall results show that the Item–Item similarity CF model fit better than the FR model in SBRS domain.

## Discussion

The literature examines next item recommendations in a number of ways. Item KNN [25], for example, estimates transition probability after an item interaction by calculating item similarity from the sessions, however simply examining items in the sessions might ignore some of the aspects of the user intention in the session. Later, the Markov chain [38] approach is created, which also depends on the transition probability between items and the item sequences in the session. Recently, neural network [6], recurrent neural network [15], graph neural network [12] and deep neural network-based [48] algorithms have been applied to session-based recommendations. These models, on the other hand, seek to discover item transition relations only via the use of item sequences. Existing approaches have achieved good results in capturing item relations in sessions, but they still have limitations. These strategies take into account the clicked items in the sessions on the same level in terms of the user's interest. On the other hand, each item may capture the user's

interest at a different degree [18]. For example, a user may spend more time on one item when browsing than on others, or a user may visit the same item many times during the session. As a result, in addition to utilising item sequence to calculate the transition probability between items, users' implicit feedback may also be included. To the best of our knowledge, this is the first effort that takes into account implicit feedback during session-based recommendation in the e-commerce domain by converting users' implicit feedback into numerical implicit ratings. Our findings add to the body of knowledge by demonstrating that when users' implicit feedback is used to generate recommendations, RS performance significantly improves.

## Conclusion and Future Works

In this study, we proposed a user behaviour aware framework called UIA to integrate user behaviour awareness to the SBRS models. In this framework, we derived numerical implicit ratings from users' behaviours in the sessions, and we utilised derived numerical implicit ratings as the context factor in the two different RS models namely FR and Item–Item CF models and compared the models' performances which are trained on derived numerical implicit rating dataset and one class implicit ratings dataset. Also, we analysed the effect of sequential awareness on the models' performance. We evaluated the UIA framework on two real-world datasets and three evaluation metrics to see how the proposed framework performs. We believe that our study has several important results:

1. Integrating users' behaviours besides other context factors in the sessions help to improve SBRS quality.
2. SBRS models are performing better when the sessions have more item interactions.
3. Using derived numerical implicit ratings enhanced FR model more than Item–Item similarity CF model. However, evaluation results for all metrics showed that Item–Item similarity CF models have better performance on SBRS. This results support why the Item–Item similarity CF models are mostly preferred in SBRS [27, 35].

The suggested UIA framework has some limitations. To begin, the estimated weights are affected by the dataset utilised in the experiments. As a result, it may be preferable to calculate these weights independently for each dataset. Second, the features considered for the weight factors must be present in the datasets. For future works, different approaches can be applied for deriving implicit numerical feedback from user behaviours. Also, different user behaviours such as only viewing, adding to cart and time spending on items can be integrated RNN based recommendation in



addition to item feature embedding and user feature embedding. In this work, we split the sessions from different levels, and we used all items in the first side as interaction, as seen in the experiment section. However, instead of using all items as interacted items, one can design a different method to analyse the effect of inputs one by one or different input combinations as the interacted items from the first part of the split session.

**Funding** No funding was received.

**Data availability** Interested parties can obtain the anonymised datasets that support the findings of this study from the corresponding author upon reasonable request.

## Declarations

**Conflict of interest** The authors declare that they have no conflict of interest.

**Ethical standards** This article does not contain any studies with human participants or animals performed by any of the authors.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

- Adomavicius G, Tuzhilin A. Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *IEEE Trans Knowl Data Eng.* 2005;6:734–49.
- Al Farani K, Nafis F, Aghoutane B, Yahyaouy A, Riffi J, Sabri A. Hybrid recommender system for tourism based on big data and ai: a conceptual framework. *Big Data Min Anal.* 2021;4(1):47–55.
- Bottou L. Stochastic gradient descent tricks. In: *Neural networks: tricks of the trade*. Springer; 2012. p. 421–436.
- Cao Y, Zhang W, Song B, Pan W, Xu C. Position-aware context attention for session-based recommendation. *Neurocomputing.* 2020;376:65–72.
- Cheng LC, Lin M-C. A hybrid recommender system for the mining of consumer preferences from their reviews. *J Inf Sci.* 2020;46(5):664–82.
- Choudhury SS, Mohanty SN, Jagadev AK. Multimodal trust based recommender system with machine learning approaches for movie recommendation. *Int J Inf Technol.* 2021;13(2):475–82.
- Duan R, Jiang C, Jain HK. Combining review-based collaborative filtering and matrix factorization: a solution to rating's sparsity problem. *Decis Support Syst.* 2022:113748.
- Esmeli R, Bader-El-Den M, Abdullahi H. Improving session based recommendation by diversity awareness. In: *UK workshop on computational intelligence*. Springer; 2019. p. 319–330.
- Esmeli R, Bader-El-Den M, Abdullahi H, Henderson D. Improving session-based recommendation adopting linear regression-based re-ranking. In: *2020 international joint conference on neural networks (IJCNN)*. IEEE; 2020. p. 1–8.
- Esmeli R, Bader-El-Den M, Abdullahi H, Henderson D. Improved session-based recommender system by context awareness in e-commerce domain. In: *13th international conference on knowledge discovery and information retrieval: KDIR 2021*. SciTePress; 2021.
- Esmeli R, Bader-El-Den M, Mohasseb A. Context and short term user intention aware hybrid session based recommendation system. In: *2019 IEEE International Symposium on INnovations in Intelligent SysTems and Applications (INISTA)*. IEEE; 2019. p. 1–6.
- Feng C, Shi C, Liu C, Zhang Q, Hao S, Jiang X. Context-aware item attraction model for session-based recommendation. *Expert Syst Appl.* 2021;176: 114834.
- Gunawardana A, Shani G. A survey of accuracy evaluation metrics of recommendation tasks. *J Mach Learn Res.* 2009;10(Dec):2935–62.
- Herlocker JL, Konstan JA, Terveen LG, Riedl JT. Evaluating collaborative filtering recommender systems. *ACM Trans Inf Syst (TOIS).* 2004;22(1):5–53.
- Hidasi B, Karatzoglou A, Baltrunas L, Tikk D. Session-based recommendations with recurrent neural networks. In: *4th international conference on learning representations, ICLR 2016*. 2016.
- Hidasi B, Quadrana M, Karatzoglou A, Tikk D. Parallel recurrent neural network architectures for feature-rich session-based recommendations. In: *Proceedings of the 10th ACM conference on recommender systems*. ACM; 2016. p. 241–248.
- Isinkaye F, Folajimi Y, Ojokoh B. Recommendation systems: principles, methods and evaluation. *Egypt Inform J.* 2015;16(3):261–73.
- Jannach D, Leriche L, Zanker M. Recommending based on implicit feedback. In: *Social information access*. Springer; 2018. p. 510–569.
- Jannach D, Ludewig M, Leriche L. Session-based item recommendation in e-commerce: on short-term intents, reminders, trends and discounts. *User Model User-Adapt Interact.* 2017;27(3–5):351–92.
- Jawaheer G, Szomszor M, Kostkova P. Comparison of implicit and explicit feedback from an online music recommendation service. In: *Proceedings of the 1st international workshop on information heterogeneity and fusion in recommender systems*. ACM; 2010. p. 47–51.
- Jawaheer G, Weller P, Kostkova P. Modeling user preferences in recommender systems: a classification framework for explicit and implicit user feedback. *ACM Trans Interact Intell Syst (TiiS).* 2014;4(2):8.
- Kaššák O, Kompan M, Bieliková M. Personalized hybrid recommendation for group of users: top-n multimedia recommender. *Inf Process Manag.* 2016;52(3):459–77.
- Ko H, Lee S, Park Y, Choi A. A survey of recommendation systems: recommendation models, techniques, and application fields. *Electronics.* 2022;11(1):141.
- Koren Y, Bell R, Volinsky C. Matrix factorization techniques for recommender systems. *Computer.* 2009;8:30–7.
- Linden G, Smith B, York J. Amazon. com recommendations: item-to-item collaborative filtering. *IEEE Internet Comput.* 2003;1:76–80.
- Liu Q, Zeng Y, Mokhosi R, Zhang H. Stamp: short-term attention/memory priority model for session-based recommendation.

- In: Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining. ACM; 2018. p. 1831–1839.
27. Ludewig M, Mauro N, Latifi S, Jannach D. Performance comparison of neural and non-neural approaches to session-based recommendation. In: Proceedings of the 13th ACM conference on recommender systems. 2019. p. 462–466.
  28. Mesas RM, Bellogín A. Evaluating decision-aware recommender systems. In: Proceedings of the eleventh ACM conference on recommender systems. ACM; 2017. p. 74–78.
  29. Montgomery AL, Smith MD. Prospects for personalization on the internet. *J Interact Mark*. 2009;23(2):130–7.
  30. Núñez-Valdez ER, Lovelle JMC, Hernández GI, Fuente AJ, Labra-Gayo JE. Creating recommendations on electronic books: a collaborative learning implicit approach. *Comput Hum Behav*. 2015;51:1320–30.
  31. Núñez-Valdéz ER, Lovelle JMC, Martínez OS, García-Díaz V, De Pablos PO, Marín CEM. Implicit feedback techniques on recommender systems applied to electronic books. *Comput Hum Behav*. 2012;28(4):1186–93.
  32. Núñez-Valdez ER, Quintana D, Crespo RG, Isasi P, Herrera-Viedma E. A recommender system based on implicit feedback for selective dissemination of ebooks. *Inf Sci*. 2018;467:87–98.
  33. Pan R, Scholz M. Mind the gaps: weighting the unknown in large-scale one-class collaborative filtering. In: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM; 2009. p. 667–676.
  34. Parra D, Karatzoglou A, Amatriain X, Yavuz I. Implicit feedback recommendation via implicit-to-explicit ordinal logistic regression mapping. In: Proceedings of the CARS-2011. 2011. p. 5.
  35. Quadrana M, Cremonesi P, Jannach D. Sequence-aware recommender systems. *ACM Comput Surv (CSUR)*. 2018;51(4):1–36.
  36. Rendle S. Factorization machines. In: 2010 IEEE 10th international conference on data mining (ICDM). IEEE; 2010. p. 995–1000.
  37. Rendle S. Factorization machines with libfm. *ACM Trans Intell Syst Technol (TIST)*. 2012;3(3):57.
  38. Rendle S, Freudenthaler C, Schmidt-Thieme L. Factorizing personalized Markov chains for next-basket recommendation. In: Proceedings of the 19th international conference on World wide web. ACM; 2010. p. 811–820.
  39. Renjith S, Sreekumar A, Jathavedan M. An extensive study on the evolution of context-aware personalized travel recommender systems. *Inf Process Manag*. 2020;57(1): 102078.
  40. Reusens M, Lemahieu W, Baesens B, Sels L. A note on explicit versus implicit information for job recommendation. *Decis Support Syst*. 2017;98:26–35.
  41. Ricci F, Rokach L, Shapira B. Introduction to recommender systems handbook. In: *Recommender systems handbook*. Springer; 2011. p. 1–35.
  42. Ricci F, Rokach L, Shapira B. Recommender systems: introduction and challenges. In: *Recommender systems handbook*. Springer; 2015. p. 1–34.
  43. Sarwar B, Karypis G, Konstan J, Riedl J. Item-based collaborative filtering recommendation algorithms. In: Proceedings of the 10th international conference on World Wide Web. ACM; 2001. p. 285–295.
  44. Shi X, He Q, Luo X, Bai Y, Shang M. Large-scale and scalable latent factor analysis via distributed alternative stochastic gradient descent for recommender systems. *IEEE Trans Big Data*. 1(1); 2020.
  45. Silva N, Carvalho D, Pereira AC, Mourão F, Rocha L. The pure cold-start problem: a deep study about how to conquer first-time users in recommendations domains. *Inf Syst*. 2019;80:1–12.
  46. Wan M, McAuley J. One-class recommendation with asymmetric textual feedback. In: Proceedings of the 2018 SIAM international conference on data mining. SIAM; 2018. p. 648–656.
  47. Wu C, Yan M. Session-aware information embedding for e-commerce product recommendation. In: Proceedings of the 2017 ACM on conference on information and knowledge management. ACM; 2017. p. 2379–2382.
  48. Zhang S, Yao L, Sun A, Tay Y. Deep learning based recommender system: a survey and new perspectives. *ACM Comput Surv (CSUR)*. 2019;52(1):1–38.
  49. Ziarani RJ, Ravanmehr R. Serendipity in recommender systems: a systematic literature review. *J Comput Sci Technol*. 2021;36(2):375–96.

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.