

Improved Session-based Recommender System by Context Awareness in e-Commerce Domain

Ramazan Esmeli¹ ^a, Mohamed Bader-El-Den¹ ^b, Hassana Abdullahi² and David Henderson³

¹*School of Computing, University of Portsmouth, Lion Terrace, Portsmouth, U.K.*

²*School of Mathematics and Physics, University of Portsmouth, Lion Terrace, Portsmouth, U.K.*

³*Fresh Relevance Ltd., Southampton Science Park, Southampton, U.K.*

Keywords: Context Awareness, Recommender Systems, e-Commerce, User Behaviour Modelling.

Abstract: Over the past two decades, there has been a rapid increase in the number of online sales. This has resulted in an increase in the data collected about the users' behaviour which has aided the development of several novel Recommender System (RS) methods. One of the main problem in RS is the lack of "explicit rating"; many customers do not rate the items they buy or view. However, the user behaviour (e.g. session duration, number of items, item duration view, etc.) during an online session could give an indication about what the user preferences "implicit rating". In this paper, we present a method to derive numerical implicit ratings from user browsing behaviour. Also, we examine the impact of using the derived numerical implicit ratings as context factors on some of the main RS methods, i.e. Factorisation Recommender and Item-Item Collaborative Filtering models. We evaluate the performance of the proposed framework on two large e-commerce datasets. The computational experiments show that in the absence of user explicit rating, the use of the user behaviour data to generate numerical implicit ratings could significantly improve the several RS methods.

1 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 (Esmeli et al., 2019b). Also, many businesses depend on RS as powerful personalised marketing tools (Montgomery and Smith, 2009) to achieve business goals and boost sales. Thus, several methods have been developed to recommend most relevant items to users (Esmeli et al., 2019a) including Content-Based Filtering (CBF) (Isinkaye et al., 2015), Collaborative Filtering (Kaššák et al., 2016) (CF) and Hybrid RS (Al Fararni et al., 2021).


One important type of RS is Session-Based Recommendation Systems (SBRS) (Esmeli et al., 2020), where the main target is to predict what the next item a particular user is likely to view. Most SBRS make use of the current browsing history only, i.e. the items the user has viewed in the session so far. This is because past browsing/purchase history is not always available and may not be relevant to the user's current intention

(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 (Hidasi et al., 2016a; Hidasi et al., 2016b; Liu et al., 2018; Wu and Yan, 2017) 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 paper is motivated by the idea 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 per-

^a  <https://orcid.org/0000-0002-2634-6224>

^b  <https://orcid.org/0000-0002-1123-6823>

formance, 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 (Jawaheer et al., 2014; Parra et al., 2011). 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 (Núñez-Valdez et al., 2018; Jawaheer et al., 2010; Reusens et al., 2017) 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. A novel users' behaviour attention model is proposed to implement User Interest Aware (UIA) SBRS in which, user behaviours are mapped to numerical implicit ratings and ratings are integrated into RS models.
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. Section 2 reviews similar works to this work. Section 3 addresses the overall proposed framework. Computational experiments and results are presented in Section 4. Finally, in Section 5, concluding remarks and future direction are reported.

2 BACKGROUND

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 by 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, feedback types and previous works related to user

behaviours mapping to numerical rating.

2.1 RS Types and Methods

In (Kaššák et al., 2016), 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 (De Gemmis et al., 2015), CF RS is hindered from adding a new item or new user (Isinkaye et al., 2015; Lika et al., 2014; Silva et al., 2019) in other mean cold start problems and sparsity problems (Isinkaye et al., 2015) and Hybrid RS (Kaššák et al., 2016) 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. The Factorisation Recommender (FR) model (Rendle, 2012) tries to learn the latent factors for the users, items and side features (context factors). 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 Item-Item similarity CF model (Kaššák et al., 2016; Sánchez and Bellogín, 2020), 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 (Domingues et al., 2013). In Jaccard similarity (Domingues et al., 2013), user ratings on items are not taken into account. While, in Cosine similarity (Domingues et al., 2013), the ratings on items are considered.

In e-commerce, generally, users are not registered, and they are anonymous. Thus, users do not have long term preference history. Also, they provide ratings rarely to show their preference explicitly to interacted items (Hidasi et al., 2016a; Hidasi et al., 2016b). However, users create sequential logs (clicked, purchased, time spent on an item, etc.) while they are browsing the system. These logs can be utilised in RS algorithms. Session-based recommendation (Hidasi et al., 2016a; Hidasi et al., 2016b) is one of the ways to adapt short term user sequential logs (preferences), missing long term user logs and anonymous users to get recommendations since User-based CF models cannot give recommendation when they do not have trained latent factor vector for a new user.

Recent works show that the improvement in the performance of the SBRS implemented on deep learning-based approaches are questioned (Jannach and Ludewig, 2017; Ludewig et al., 2019; Dacrema et al., 2019). The main concerns of the deep-learning-based SBRS are reproducibility, scalability, and performance (Zhang et al., 2019). For example, in

(Ludewig et al., 2019; Dacrema et al., 2019) only on one dataset deep-learning-based SBRS model outperformed modified Item-Item similarity CF RS on different domains such as music and e-commerce domain. Therefore, in this work, we prefer to use Item-Item similarity CF RS model. However, the proposed method can be applied to any deep-learning-based approach by feeding the model with calculated interest level score as a context factor.

2.2 Context Awareness in Session-based Recommender Systems

The recommendation list can be influenced by the context the user is in. The works (Cao et al., 2020; Jannach et al., 2017; Renjith et al., 2020) investigated the context factor on the performance of the recommendation model. (Cao et al., 2020) 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, (Jannach et al., 2017) investigated the role of discounts, the effects of adopting users' short term intention and popularity trends of the products on RS performance.

2.3 Explicit and Implicit Feedback Correlation

In (Jawaheer et al., 2010), they proposed a work to show the correlation between implicit feedback and explicit feedback. 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.

In (Reusens et al., 2017), a method to compare performance measurement of two types of feedback applied to the job domain is designed. 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 (Núñez-Valdéz et al., 2012), 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.

2.4 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 (Koren et al., 2009; Jannach et al., 2017) 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 (Hidasi et al., 2016a; Jannach et al., 2017). 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 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.

3 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.

3.1 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 encoding¹ 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 was collected for a 15 day period from a real-world e-commerce website². The second dataset is the Yoochoso RecSys dataset³ which stores click events from an e-commerce website. Table 1 shows the statistics about the number of unique items, sessions and total interactions in each dataset.

¹<https://scikit-learn.org/>

²<https://www.freshrelevance.com>

³<https://2015.recsyschallenge.com/challenge.html>

Table 1: Dataset statistics used in this work.

dataset	Sessions	Items	Interactions
fresh relevance	71596	36605	1917985
yoochoso	197601	37220	3291424

3.2 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.

3.2.1 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 $u \in U$ interacted with the product $i \in I$ or not in a session $s \in S$. For the proposed framework, we used Equation 1 for the simple implicit feedback (One Class) rating representation.

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

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 (Reusens et al., 2017), 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 (Wan and McAuley, 2018; Pan and Scholz, 2009).

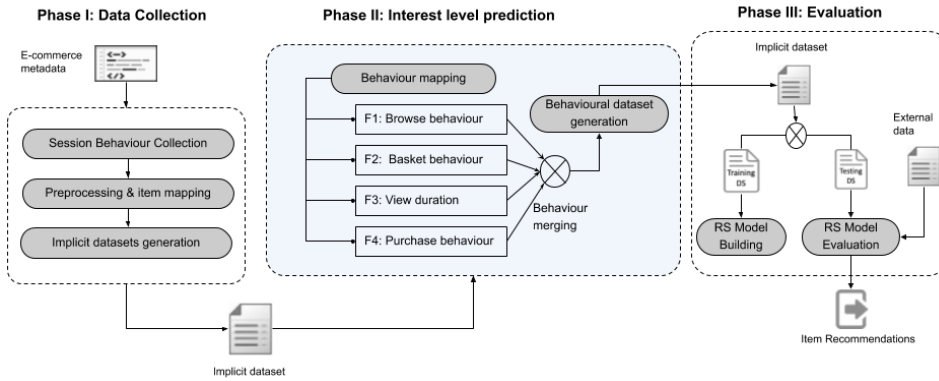


Figure 1: User Interest Aware Framework(UIA).

3.2.2 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 (Reusens et al., 2017; Núñez-Valdez et al., 2018). 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 (Núñez-Valdez et al., 2015; Reusens et al., 2017) 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.

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

In order 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 exam-

ple, if a user did not like an item he browsed, he may have the intention to click another item in a minute.

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

We define different mathematical equations to indicate user $u \in U$ interest level on item $i \in I$. 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 (Jawaheer et al., 2010), 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. In order to have a normalised value for this indicator, we formulate the calculation of this indicators contribution in Equation 2. 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

$$F1(i, s) = \frac{c_i}{t_c} \quad (2)$$

$F2$: indicates level, in which if item i is added to basket in session s (Eq. 3) 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 ($t_a > 0$).

$$F2 = \frac{t_i}{t_a} \quad (3)$$

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 Equation 4 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

$$F3 = \frac{i_d}{t_d} \quad (4)$$

$F4$: 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 Equation 5. 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 ($t_p > 0$).

$$F4 = \frac{i_p}{t_p} \quad (5)$$

3.2.4 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. 6).

$$\sum_{n=1}^4 w_n = 1 \quad (6)$$

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. 7). 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.

$$score = w1 * F1 + w2 * F2 + w3 * F3 + w4 * F4 \quad (7)$$

3.3 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 Yoochose 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.

4 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.

4.1 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 tool⁴.

4.1.1 FR Model

If any side data is 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.

4.1.2 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.

4.2 Evaluation Metrics

In the literature, accuracy, precision, recall and coverage are some metrics used in RS (Herlocker et al., 2004). $recall@n$ (Eq. 8) and $precision@n$ (Eq. 9) have been used widely in the top-n ranked list RS (Ricci et al., 2015; Herlocker et al., 2004; Gunawardana and Shani, 2009). 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.

$$recall@n = \frac{|Recommended\ Items \cap Ground\ Truth|}{|Ground\ Truth|} \quad (8)$$

$$prec@n = \frac{|Recommended\ Items \cap Ground\ Truth|}{|Recommended\ Items|} \quad (9)$$

$coverage@n$ (Eq. 10) describes the ratio of the number of the users retrieved at least one correct recommendation U_{mi} to all user U in test data (Mesas and Bellogín, 2017).

$$coverage@n = \frac{|U_{mi}|}{|U|} \quad (10)$$

⁴<https://turi.com/products/create/docs/graphlab.toolkits.recommender.html>

4.2.1 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.

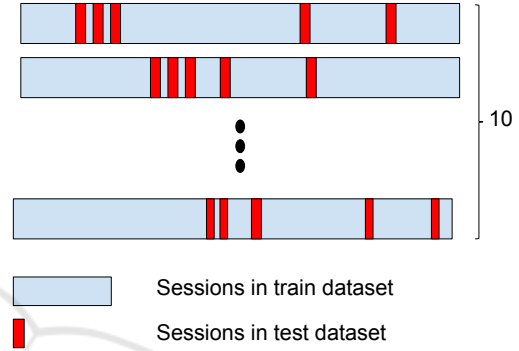


Figure 2: Dataset Splitting method for Sequence Aware model.

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 Section 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 (Jannach et al., 2018), the best ones are identified, as shown in Table 3. The search space for the weights are restricted between 0 and 1.

Table 3: Best weights for the user behaviours.

Contributing Factor	weight indicator	value
F1	w_1	0.2
F2	w_2	0.4
F3	w_3	0.3
F4	w_4	0.9

4.2.2 Experiments

We choose two different recommendation models which are Item-Item similarity CF (Kaššák et al., 2016) and FR models (Rendle, 2010). 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)(Shi et al., 2020) 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.

For each interacted item(item sequence), we retrieve $top@n$ $n \in [20]$ 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. 3. As shown in Fig. 3, 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.

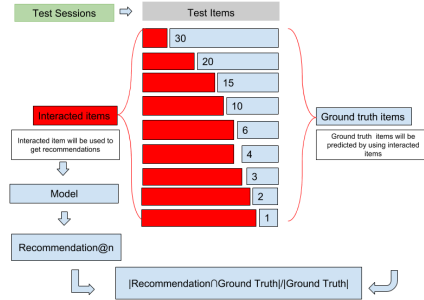


Figure 3: Test method followed in Sequence Aware Recommendation.

4.3 Result and Analysis

We report the results of experiments in Tables 5, 4, 6 and 7. In Tables, G shows the number of items in the ground truth. As seen in Figure 3, 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 Yoochose RecSys dataset, while Tables 6 and 7 show the evaluation results for Item-Item CF and the FR RS models on Fresh relevance dataset.

Table 4: Performance comparison on different length of user interaction of Yoochose RecSys dataset with Item-Item similarity CF model.

G	recall@20		precision@20		coverage@20	
	impl.	exp.	impl.	exp.	impl.	exp.
1	0.32450	0.31900	0.02170	0.02160	0.43400	0.43200
2	0.30725	0.30125	0.04172	0.04122	0.56800	0.57700
3	0.29016	0.28966	0.05912	0.05990	0.63150	0.65750
4	0.26437	0.27337	0.07445	0.07692	0.68400	0.72200
6	0.22991	0.24558	0.10075	0.10617	0.74050	0.79900
10	0.12950	0.15535	0.09957	0.11510	0.63750	0.68400
15	0.09861	0.12629	0.12187	0.14330	0.69047	0.74404
20	0.08402	0.10910	0.14312	0.17302	0.75601	0.81786
30	0.05200	0.08222	0.13800	0.19400	0.74666	0.89333

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

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

G	recall@20		precision@20		coverage@20	
	impl.	exp.	impl.	exp.	impl.	exp.
1	0.01350	0.03450	0.00067	0.00172	0.01350	0.03450
2	0.01650	0.03475	0.00165	0.00347	0.03150	0.06300
3	0.01700	0.03400	0.00255	0.00510	0.04350	0.08650
4	0.01762	0.03387	0.00352	0.00677	0.05700	0.10450
6	0.01716	0.03250	0.00515	0.00975	0.07800	0.13750
10	0.02330	0.03105	0.01165	0.01552	0.15550	0.19500
15	0.02212	0.02748	0.01659	0.02061	0.20089	0.22767
20	0.02061	0.02543	0.02061	0.02543	0.22680	0.25429
30	0.02222	0.02488	0.03333	0.03733	0.29333	0.33333

Table 6: Performance comparison on different length of user interaction of Item-Item similarity CF model on Fresh relevance dataset.

G	recall@20		precision@20		coverage@20	
	impl.	exp.	impl.	exp.	impl.	exp.
1	0.25150	0.24850	0.01277	0.01180	0.25550	0.23600
2	0.21875	0.21700	0.02101	0.03474	0.42031	0.41327
3	0.19466	0.19466	0.02502	0.04691	0.50050	0.48277
4	0.18050	0.18050	0.02682	0.05552	0.53648	0.51798
6	0.15741	0.15766	0.02471	0.05491	0.49427	0.47450
10	0.10500	0.10530	0.02640	0.05955	0.52816	0.50201
15	0.10709	0.10762	0.02139	0.04888	0.42783	0.41408
20	0.07374	0.07537	0.03042	0.06818	0.60839	0.59790
30	0.07365	0.07419	0.03179	0.07135	0.63592	0.59708

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

G	recall@20		precision@20		coverage@20	
	impl.	exp.	impl.	exp.	impl.	exp.
1	0.01900	0.02700	0.00147	0.0017	0.01900	0.02700
2	0.01729	0.02155	0.00482	0.0053	0.03318	0.03971
3	0.01876	0.02378	0.00658	0.00704	0.05167	0.06028
4	0.01752	0.02181	0.00845	0.00889	0.06166	0.07040
6	0.01604	0.01978	0.01012	0.01040	0.08168	0.09105
10	0.01400	0.01695	0.01222	0.01247	0.21629	0.23340
15	0.01247	0.01478	0.01391	0.01451	0.25773	0.26460
20	0.01070	0.01209	0.01486	0.01643	0.31118	0.32517
30	0.01068	0.00976	0.01553	0.01771	0.18446	0.16504

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.

5 CONCLUSION

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 (Quadrona et al., 2018; Ludewig et al., 2019).

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.

REFERENCES

- Al Farani, K., Nafis, F., Aghoutane, B., Yahyaouy, A., Riffi, J., and Sabri, A. (2021). Hybrid recommender system for tourism based on big data and ai: A conceptual framework. *Big Data Mining and Analytics*, 4(1):47–55.
- Cao, Y., Zhang, W., Song, B., Pan, W., and Xu, C. (2020). Position-aware context attention for session-based recommendation. *Neurocomputing*, 376:65–72.
- Dacrema, M. F., Cremonesi, P., and Jannach, D. (2019). Are we really making much progress? a worrying analysis of recent neural recommendation approaches. In *Proceedings of the 13th ACM Conference on Recommender Systems*, pages 101–109.
- De Gemmis, M., Lops, P., Semeraro, G., and Musto, C. (2015). An investigation on the serendipity problem in recommender systems. *Information Processing & Management*, 51(5):695–717.
- Domingues, M. A., Jorge, A. M., and Soares, C. (2013). Dimensions as virtual items: Improving the predictive ability of top-n recommender systems. *Information Processing & Management*, 49(3):698–720.
- Esmeli, R., Bader-El-Den, M., and Abdullahi, H. (2019a). Improving session based recommendation by diversity awareness. In *UK Workshop on computational intelligence*, pages 319–330. Springer.
- Esmeli, R., Bader-El-Den, M., Abdullahi, H., and Henderson, D. (2020). Improving session-based recommendation adopting linear regression-based re-ranking. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.
- Esmeli, R., Bader-El-Den, M., and Mohasseb, A. (2019b). 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)*, pages 1–6. IEEE.
- Gunawardana, A. and Shani, G. (2009). A survey of accuracy evaluation metrics of recommendation tasks. *Journal of Machine Learning Research*, 10(Dec):2935–2962.
- Herlocker, J. L., Konstan, J. A., Terveen, L. G., and Riedl, J. T. (2004). Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems (TOIS)*, 22(1):5–53.
- Hidasi, B., Karatzoglou, A., Baltrunas, L., and Tikk, D. (2016a). Session-based recommendations with recurrent neural networks. In *4th International Conference on Learning Representations, ICLR 2016*.
- Hidasi, B., Quadrana, M., Karatzoglou, A., and Tikk, D. (2016b). Parallel recurrent neural network architectures for feature-rich session-based recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems*, pages 241–248. ACM.
- Isinkaye, F., Folajimi, Y., and Ojokoh, B. (2015). Recommendation systems: Principles, methods and evaluation. *Egyptian Informatics Journal*, 16(3):261–273.
- Jannach, D., Lerche, L., and Zanker, M. (2018). Recommending based on implicit feedback. In *Social information access*, pages 510–569. Springer.
- Jannach, D. and Ludewig, M. (2017). When recurrent neural networks meet the neighborhood for session-based recommendation. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*, pages 306–310.
- Jannach, D., Ludewig, M., and Lerche, L. (2017). Session-based item recommendation in e-commerce: on short-term intents, reminders, trends and discounts. *User Modeling and User-Adapted Interaction*, 27(3-5):351–392.
- Jawaheer, G., Szomszor, M., and Kostkova, P. (2010). 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*, pages 47–51. ACM.
- Jawaheer, G., Weller, P., and Kostkova, P. (2014). Modeling user preferences in recommender systems: A classification framework for explicit and implicit user feedback. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 4(2):8.
- Kaššák, O., Kompan, M., and Bieliková, M. (2016). Personalized hybrid recommendation for group of users: Top-n multimedia recommender. *Information Processing & Management*, 52(3):459–477.
- Koren, Y., Bell, R., and Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer*, (8):30–37.
- Lika, B., Kolomvatsos, K., and Hadjiefthymiades, S. (2014). Facing the cold start problem in recommender systems. *Expert Systems with Applications*, 41(4):2065–2073.
- Liu, Q., Zeng, Y., Mokhosi, R., and Zhang, H. (2018). Stamp: short-term attention/memory priority model for session-based recommendation. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1831–1839. ACM.
- Ludewig, M., Mauro, N., Latifi, S., and Jannach, D. (2019). Performance comparison of neural and non-neural approaches to session-based recommendation. In *Proceedings of the 13th ACM Conference on Recommender Systems*, pages 462–466.

- Mesas, R. M. and Bellogín, A. (2017). Evaluating decision-aware recommender systems. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*, pages 74–78. ACM.
- Montgomery, A. L. and Smith, M. D. (2009). Prospects for personalization on the internet. *Journal of Interactive Marketing*, 23(2):130–137.
- Núñez-Valdez, E. R., Lovelle, J. M. C., Hernández, G. I., Fuente, A. J., and Labra-Gayo, J. E. (2015). Creating recommendations on electronic books: A collaborative learning implicit approach. *Computers in Human Behavior*, 51:1320–1330.
- Núñez-Valdéz, E. R., Lovelle, J. M. C., Martínez, O. S., García-Díaz, V., De Pablos, P. O., and Marín, C. E. M. (2012). Implicit feedback techniques on recommender systems applied to electronic books. *Computers in Human Behavior*, 28(4):1186–1193.
- Núñez-Valdez, E. R., Quintana, D., Crespo, R. G., Isasi, P., and Herrera-Viedma, E. (2018). A recommender system based on implicit feedback for selective dissemination of ebooks. *Information Sciences*, 467:87–98.
- Pan, R. and Scholz, M. (2009). 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*, pages 667–676. ACM.
- Parra, D., Karatzoglou, A., Amatriain, X., and Yavuz, I. (2011). Implicit feedback recommendation via implicit-to-explicit ordinal logistic regression mapping. *Proceedings of the CARS-2011*, page 5.
- Quadrana, M., Cremonesi, P., and Jannach, D. (2018). Sequence-aware recommender systems. *ACM Computing Surveys (CSUR)*, 51(4):1–36.
- Rendle, S. (2010). Factorization machines. In *Data Mining (ICDM), 2010 IEEE 10th International Conference on*, pages 995–1000. IEEE.
- Rendle, S. (2012). Factorization machines with libfm. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 3(3):57.
- Renjith, S., Sreekumar, A., and Jathavedan, M. (2020). An extensive study on the evolution of context-aware personalized travel recommender systems. *Information Processing & Management*, 57(1):102078.
- Reusens, M., Lemahieu, W., Baesens, B., and Sels, L. (2017). A note on explicit versus implicit information for job recommendation. *Decision Support Systems*, 98:26–35.
- Ricci, F., Rokach, L., and Shapira, B. (2015). Recommender systems: introduction and challenges. In *Recommender systems handbook*, pages 1–34. Springer.
- Sánchez, P. and Bellogín, A. (2020). Time and sequence awareness in similarity metrics for recommendation. *Information Processing & Management*, 57(3):102228.
- Shi, X., He, Q., Luo, X., Bai, Y., and Shang, M. (2020). Large-scale and scalable latent factor analysis via distributed alternative stochastic gradient descent for recommender systems. *IEEE Transactions on Big Data*, 1(1).
- Silva, N., Carvalho, D., Pereira, A. C., Mourão, F., and Rocha, L. (2019). The pure cold-start problem: A deep study about how to conquer first-time users in recommendations domains. *Information Systems*, 80:1–12.
- Wan, M. and McAuley, J. (2018). One-class recommendation with asymmetric textual feedback. In *Proceedings of the 2018 SIAM International Conference on Data Mining*, pages 648–656. SIAM.
- Wu, C. and Yan, M. (2017). Session-aware information embedding for e-commerce product recommendation. In *Proceedings of the 2017 ACM on conference on information and knowledge management*, pages 2379–2382. ACM.
- Zhang, S., Yao, L., Sun, A., and Tay, Y. (2019). Deep learning based recommender system: A survey and new perspectives. *ACM Computing Surveys (CSUR)*, 52(1):1–38.