Ready for You When You Are Back: Content-Driven Session-Based Recommendation for Continuity of Experience

Brijraj Singh, Sonal, Niranjan Pedanekar

Data Science, Sony Research India
brijraj.singh@sony.com, sonal.dabral@sony.com, niranjan.pedanekar@sony.com

Abstract

Recommender systems used in online platforms can drive users to consume content continuously in an attempt to maximize satisfaction. Such engagement is invariably broken due to more pressing work, alternate pursuits, distractions or fatigue. Recommender systems need to ensure the continuity of experience when the user joins back. Session-based recommender systems typically create different sessions based on a fixed time interval (θ), often resulting in creation of a separate session when the user gets off the platform temporarily. When the user joins back, session-based recommender systems are likely to recommend content different than what they would have in case the earlier session had continued. This may cause dissatisfaction given that there is a difference in the predicted world model of the user, i.e. the expectation from the last session, and the observed one, i.e. the recommendations.

To handle this problem, we propose the creation of content-driven sessions instead of time-driven sessions. In our setting, a session continues while a single item category dominates in the user-item interactions. A new session is created when a different item category begins to dominate. The proposed content-driven method also solves the long-standing problem of deciding the optimal value of time threshold (θ) for defining the time-based session. We report that the proposed method outperforms existing SOTA methodologies set by time-based sessions by a large margin in terms of recommendation performance on multiple datasets.

Extended version —
https://github.com/brijraj08/Ready_Recommendation

Introduction

Though recommender systems are technical in nature, they essentially deal with human behavior. This work is motivated by following behavioral and technical considerations. **Behavioral Motivation:** According to the *theory of predictive coding* and the *free energy principle* in neuroscience, the human brain works towards reducing the difference between the predicted world model and the observed world state through sensory input (Friston and Kiebel 2009), where the difference is known as free energy. The reduction in free energy gives a sense of satisfaction through the release of Dopamine and motivates the brain to optimize it even further

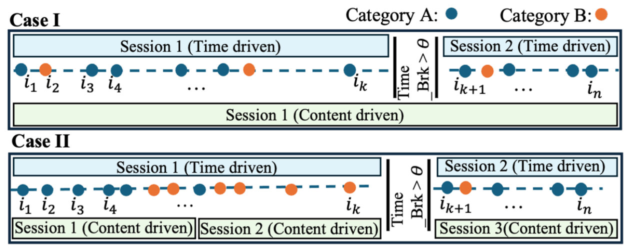


Figure 1: Cases when CD-SBRS differs from TS-SBRS

(Friston et al. 2014) by continuing the engagement. In the case of online services, we realize that it is the recommendation model that provides more relevant content to the user which the brain finds closer to the predicted world model (based on prior experience of engagement) and is one of the reasons for Dopamine release. This often results in a continuous engagement with the platform, which is invariably broken at some time, due to more pressing work, alternate pursuits, distractions, fatigue, or simply wanting to get out of a *rabbit hole* [1].

Session-based recommendation models have been tremendously successful in keeping users engaged with online platforms (de Souza Pereira Moreira et al. 2021), (Dabral, Singh, and Onoe 2023), (Liu et al. 2018), (Li et al. 2017). Yet, such a recommendation model needs to take into account that the user has temporarily left the platform so that when the user comes back again, she should have the same experience as before. In a typical case of a time-driven session-based recommendation system (SBRS), if the user is inactive or off the platform for a period longer than a threshold θ, recommendations can get altered when she joins back, as compared to what she would have got in continued interaction. This may lead to a difference between the predicted world model (i.e. expectation from the last session) and the observed one (i.e. new recommendations) and can trigger a sense of dissatisfaction.

This work proposes using a content-driven session for session-based recommendation, which alters recommendations only if the user drifts toward new choices of content,

[1] https://hbr.org/2022/01/the-psychology-of-your-scrolling-addiction

but otherwise continues recommendations considering earlier choice, irrespective of the time elapsed. This can result in a continuity of experience, and aid in positive engagement.

Technical Motivation: Each individual shows a different behavior while using online services. Even if the content provided by recommendation is highly relevant to a group of users, only a few users can afford to be in continuous engagement (e.g. binge-watching). Most other users can consume content only in an intermittent manner because of other tasks, distractions, or fatigue. Time-driven SBRS considers an idle time threshold θ as a waiting period and if the user does not come back in this duration, it creates new session. Along with this, the insight behind changing the session is that user's activities might vary after a certain duration of inactivity (Bernardis et al. 2022). However, this approach assumes a simplified scenario, as it overlooks two critical factors: (a). Each user shows different behavior so a common constant θ can not correctly identify the session boundaries for all users, (b). Inactivity (of duration θ) does not always infer the user's lack of interest or change in preference. To address these limitations, we propose a methodology for determining session breaks based on assessing the homogeneity in content interaction to establish session boundaries, utilizing time stamps solely for ordering interactions. A typical case of user navigation across multiple items is shown in Figure 1 where in case-1, user continues the same preference even after a long time break (or shifts to a different preference without a break at all, in case-2). This behavior can be observed in datasets listed in Table 1.

In this work, the proposed method considers drift in the user's choice for changing the session. For example, a user watches (or clicks) more $action$ movies in her recent interaction and then gradually switches towards watching more $comedy$ movies in the next couple of interactions. In this case, the proposed method will identify the point when the dominance of the $action$ genre fades and the dominance of the $comedy$ genre intensifies. The chunk of items under a single dominating category is named a Homogeneous Session. We have performed multiple experiments to test the efficacy of homogeneous content-based sessions on existing SBRS and witnessed improved performance in all the cases. To the best of our knowledge in the field of recommendation systems, it is the first study that considers content-driven session creation and observes improvement in recommendation performance. Our contribution can be summarized as follows:

- We propose a novel methodology for session creation based on content homogeneity.
- We propose a method of assigning a new label to each item on the basis of the category represented in the embedding space, if such labels are absent.

We applied our proposed method to create sessions and used them with SOTA session-based recommendation systems to observe their performance. Our findings suggest that content-based sessions consistently outperform time-drive sessions across all recommendation models.

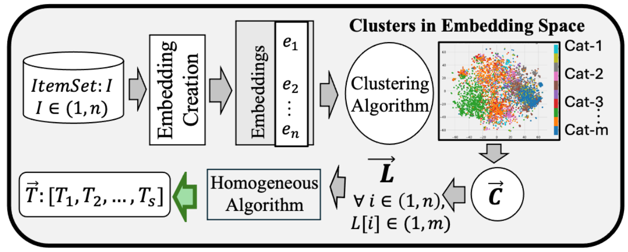


Figure 2: Homogeneous Session Creator Architecture

Related Work

In order to provide recommendations of unused or unseen items to users, there have been multiple solutions to fill up the sparsity of $user, item$ interaction matrix. The initial step to achieve this was Collaborative Filtering (CF), which takes similar users into consideration and shares their likings among them. CF is of types like a) Memory-based CF b) Model-based CF. Memory-based CF tries to find similar users and then considers the items consumed by one user to recommend others. Whereas model-based CF provides more scalability by considering a model such as Matrix Factorization-based method (Guo et al. 2017) (Wang, De Vries, and Reinders 2008), (Su et al. 2007), (DeCoste 2006), (Marlin and Zemel 2004), (Pazzani 1999). CF has also been implemented leveraging neural networks and named as NCF (Chen et al. 2021).

Session-based recommendation system (SBRS) serves the motivation of providing the recommendation based on the user's most recent activity on the platform. It has been observed that recommendations given on the basis of sessions, help in achieving higher CTR (Click Through Rate). It has also been observed that SBRS completes the recommendation cycle by considering Recall, MRR, and hit ratio as their evaluation metrics, unlike other methods, which only serve up to the prediction of ratings (Hidasi et al. 2015), (Devooght and Bersini 2017). In the RMSE prediction task, even if an item has been predicted high on the rating there would be multiple items that users could have found interesting but they all can not be recommended (Mondal et al. 2023). Also, the timing of the recommendation also matters to get high ratings from the users. This way RMSE prediction-based recommendation system fails to fulfill the motivation of recommendation.

Research Gap

In all the previous studies of SBRS, it is the $time$ that was considered for defining the session. In general, 30 minutes is the time period that has been extensively used to limit the length (Bernardis et al. 2022) of the break (or considered as hyperparameter but once decided is fixed for all the users). Considering a constant time period as a threshold forces any of the session-based models to summarize the user activities, even if they are heterogeneous in nature. This confuses the recommendation models, which expect homogeneous behavior within a session so that it can clearly capture short-term preferences and summarize them.

Methodology

This section first introduces the problem statement and then a solution of the stated problem. The proposed architecture can be followed from Figure 2 and the solution is given through the subsections referred as: Data Accumulation, Embedding Creation, Embedding Clustering, Label Assignment, and Homogeneous Session Creation, followed by Model Training.

Basic Notations: User ID is defined as U_Id, item Id is defined as I_Id, time-stamp driven as TS, content driven as CD, rating as $\mathcal{R}$. The description of movies is defined as $\vec{\mathcal{D}}$. BERT embedding of a movie is defined as $\vec{\mathcal{E}}$. Movie, Book, Song are used to refer to the item.

Problem Formulation

In a system of N items, a session ($\mathcal{S} = i_1, i_2, \cdots, i_{\mathcal{W}} : \mathcal{W} \in (1, N)$ where each item has an associated category) will continue including interactions in same session until $\Delta(\mathcal{S})$ is singleton where $\Delta = max(\nu(j))$, and $\nu(j)$ is the frequency of category of item j such that $j \in (1, N)$, ($|\mathcal{S}| \geq 2$) and $\Delta(\mathcal{S}) \neq \Delta(\mathcal{S}')$ and $\mathcal{S}, \mathcal{S}'$ are adjacent sessions.

A session (having at least 2 items) needs to continue including items while the point it finds only a single dominating category and splits the session otherwise.

Creating Item Representation

Explicit representation of the items through their attributes is the essential requirement of the proposed method. The attributes of the items vary with the domain, for instance in the domain of movies, attributes like plots of the movies (including the cast, directors, storylines, genres, etc.) serve the purpose. In the domain of books, descriptions of the books contain enough information for their unique representation. In the music domain, the artist, genre, music type, etc. are features that can be considered. In the case of e-commerce, a product category is usually created when an item is placed for sale and that category can be considered for representing the item. CD sessions are created as follows:

Data Accumulation

The dataset, for this work is expected to have $item\_Id$, $user\_Id$, and $time\_stamp$ as primary attributes. In order to collect more discriminatory features from the item-set, a textual description of each item is also collected. In the domain of movie recommendation, the description (or plot) of the movie is considered from IMDB like movie repository or can also be acquired using LLM (large language model) (Acharya, Singh, and Onoe 2023) or through web scrapping from any online sources like blogs. In the domain of books, the description of the book can be acquired using web scrapping (if not available). In the music domain, the genre of the song is acquired using LLM. In the case of e-commerce, item categories are available as part of the dataset and are considered directly as labels. Note: The description of the item (movie, book, song) is expected to be large enough to cover the uniqueness of the item.

Embedding Creation

A description of the item acquired in the previous subsection is used to represent the items in the feature space. This work considers BERT for creating the embedding using text-based descriptions. Since each item brings a new property, hence each item can be placed uniquely in the feature space created by BERT embeddings. Therefore, the pre-trained BERT model is exploited to formulate the embedding $\mathcal{E}$ corresponding to each $item\_Id$ and item description ($\vec{\mathcal{D}}$) as shown in Eq. 1.

$$\vec{\mathcal{E}} = BERT(I\_Id_1||\mathcal{D}[1], I\_Id_2||\mathcal{D}[2] \cdots ||\mathcal{D}[n]) \qquad (1)$$

Embedding Clustering

Item Id I_Id (Movie Id, Book Id, Song Id etc.) is a unique number that has no pattern; even two similar items are not supposed to have anything common in I_Id. This motivates us to consider the natural language description of the items to improve their representation. Since similar kinds of items are positioned close to each other in the embedding space, we can cluster similar items together (description-based clusters: Figures 3, 4). We tried multiple clustering algorithms and selected one on the basis of clustering performance as shown in Figure 3, 4.

$$K = Elbow\_method(\vec{\mathcal{E}}); \qquad \vec{\mathcal{C}} = kMean(\vec{\mathcal{E}}, K) \qquad (2)$$

$\vec{C}$ is the collection of K cluster centers. K is selected using Elbow method (shown in Eq. 2, Figures 3, 4 (Thorndike 1953), (Onumanyi et al. 2022)), which produces the optimal number of clusters. Each cluster represents a collection of similar kinds of items. As clustering is performed in embedding space, it is expected to have considered all the manually crafted features like genre, casting, story, etc.

Label Assignment

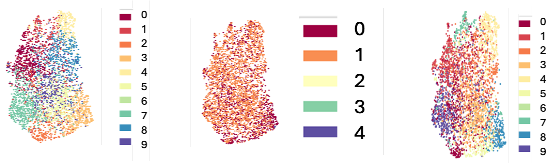
The clusters developed in the previous subsection are used to assign a new label to each item. For instance, if an item belongs to cluster # $C1$, the item will be labeled as: $C1$.

$$Label(item_i) = Label(min(\mathrm{d}(item_i, \vec{C}_j))) \qquad (3)$$

where $i \in (1, N)$ and $j \in (1, |\vec{C}|)$ and d is the distance between $item_i$ and centers of all the clusters. In the session-based recommendation setting, we consider these new labels for preparing the homogeneous session window, and all the interactions covered in a session window are allotted with the same $session\_Id$. Whereas, after creating the session, the dataset got enriched with one additional attribute named as $session\_Id$.

Definition 1. *A Homogeneous session is a sequence I_{seq} of items with a single dominating item category in the presence of other categories in the sequence of interaction.*

Definition 2. *A dominating category is one which occurs more than any other items category in users' interaction in the user-item interaction sequence such that $\#(I_d, I_{seq}) > \#(I_i, I_{seq}), \forall I_i \in I_{seq}$ and $I_i \neq I_d$ where $\#$ represents the number of times item I_i or I_d is found in I_{seq} (user-item interactions) and I_d is the dominating item category.*



Clustering	Silhouette Coefficient
K-Means	**0.02433**
DBSCAN	0.01745
Agglomerative	0.00573

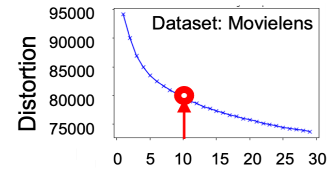
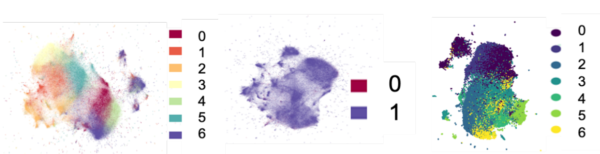


Figure 3: Clusters of movies created using BERT embeddings on movies plot, #clusters are found using Elbow method



Clustering	Silhouette Coefficient
K-Means	**0.03882**
DBSCAN	-0.22598
Agglomerative	0.02681

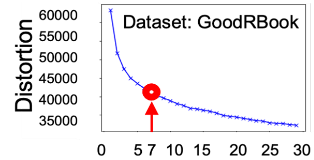


Figure 4: Clusters of books created using BERT embeddings on book description, #clusters are found using Elbow method

Homogeneous Session Creation

Session-creation method is applied over the sequence of items. Sessions are created in a way so that every single session maintains the homogeneous content at any point in time. As shown in Algorithm 1, a session continues to consider the items in the window while a single category of the items dominates. As the user behavior is dynamic, it keeps drifting from one category of content to another. Therefore, in the response; a new session starts right after the last occurrence of the dominating category in the ongoing session. The method ensures that each session window keeps a-like content and curates the sessions accordingly.

As shown in Algorithm 1, it considers the sequence of items $\mathcal{I}$ as input and keeps adding the items in the window (Φ is a hashmap and counts the occurrences of each category) when it finds more than 2 occurrences of the same category, it is considered as dominating category, and a marker keeps monitoring index of every occurrence of this dominating category (line 3-5, algorithm.1). When it finds 2 such dominating categories, then it considers the previously set marker as the split point and starts a new session for next dominating category and then resets all the markers and hashmap (line 6-12, Algorithm 1).

Theorem 1. *A dominating item of the session is the item likely to be clicked next.*

Proof *Let the sequence of items in the interaction data of user history is $I_{seq} = I_1, I_2, \cdots, I_n$. If training is performed in a user (or session) parallel fashion where an item to be clicked next is predicted on the basis of the current item in the sequence. As per Definition 1, $\exists I_d \in I_{seq}$ as the dominating item. Suppose the user clicks on item x as the first action in the session and $x \in I_{seq}$ is the first item in the session.*

$$\mu_i = \Phi(P(I_i|x = I_1), P(I_i|x = I_2), \cdots, P(I_i|x = I_n)) \tag{4}$$

where $i \in (1, n)$ and

$$P(I_i|x = I_1) = \frac{\#(I_i||I_1)}{\sum \#(I_i||I_l)} : I_l \in I_{seq} \tag{5}$$

Where $I_i||I_1$ represents the adjacent occurrence of the items keeping the relative position. If I_d is the dominating item in the session then

$$\mu_d > \mu_j, \forall j \in (1, n), j \neq d \tag{6}$$

using Eq. 5 and Φ being the aggregator. The above relation holds true also for the cases when a previous sequence of items (instead of one item) is considered to predict the next item. If $\mathcal{S}$ denotes the sequence of items such that $|\mathcal{S}| \geq 1$.

$$\mu\_seq_i = \Phi(P(I_i|x = \mathcal{S}_1), \cdots, P(I_i|x = \mathcal{S}_n)) \tag{7}$$

where $|\mathcal{S}_1| = 1, |\mathcal{S}_2| = 2, \cdots, |\mathcal{S}_n| = |I_{seq}|$ (A case of training when each upcoming item becomes part of the sequence) In a similar manner, If I_d is the dominating item in the session then

$$\mu\_seq_d > \mu\_seq_j \forall j \in (1, n), j \neq d. \tag{8}$$

The above equations (Eq. 6, 8) verify the idea that a dominating item I_d of a sequence has higher probability of occurrence and is likely to be clicked next.

Algorithm 1: *Homogeneous Content Window Creation*

Input:$\mathcal{I}$ **Output**:$\vec{\mathcal{H}}$

1: **for** each i $\in (0, len(\mathcal{I}))$ **do**
2: $\quad\Phi[\mathcal{I}[i]] \leftarrow \Phi[\mathcal{I}[i]] + 1$
3: $\quad$**if** $\Phi[item] > 2$ and $\mathcal{S} < 2$ **then**
4: $\quad\quad\mathcal{M} = i$
5: $\quad$**end if**
6: $\quad$**if** $max(\Phi) > 2$ and $\mathcal{S} == 2$ **then**
7: $\quad\quad\S.append(\mathcal{M})$
8: $\quad\quad$**if** $\mathcal{M} + 1 < len(\mathcal{I} - 1)$ **then**
9: $\quad\quad\quad i = \mathcal{M}$
10: $\quad\quad$**end if**
11: $\quad\quad\mathcal{M} = 0$
12: $\quad\quad\Phi = zeros(len(item\_groups))$
13: $\quad$**end if**
14: $\quad$i=i+1
15: **end for**
16: **for** i in $(0, len(\S))$ **do**
17: $\quad\mathcal{H}.append(\mathcal{I}[\S[i] + 1 : \S[i+1] + 1])$
18: **end for**

$d \in \mathcal{S}$ (%)	$d \in \mathcal{S}_{N[-1]}$		$d \in \mathcal{S}_{(N+1)[1]}$		$d \in \mathcal{S}_{(N+1)}$	
	TS	CD	TS	CD	TS	CD
Movielens	43.88	96.69	25.85	1.14	69.12	59.30
Book	47.87	95.40	29.51	1.61	83.16	68.86
Amazon	62.98	97.77	34.99	1.04	35.89	10.93
LastFM	49.13	99.85	25.37	2.70	81.11	40.24

Table 1: Existence of dominating item as the last item of the same session (col 2, 3), as the first item in next session (col 4, 5), is present in next session (col 6, 7)

Experimental Setup

This section provides details about datasets, baseline methods, and other details considered in the experiments.

Dataset Analysis: Criteria of selection This work needs $time\_stamp$ for arranging the items in order of occurrence. Since $time\_stamp$ is a subject-dependent attribute hence the best of the dataset contains $user\_id, item\_id, time\_stamp, item\_category$ as well. Datasets like $LastFM$ (and Movielens) doesn't contain a category in general so we leveraged LLM (or online sources) to acquire genre of the track and considered it as a category. This work includes Movielens (Harper and Konstan 2015) dataset for movie domain, Goodreads Book data for the book domain (Wan and McAuley 2018), (Wan et al. 2019), LastFM for the music domain (Celma 2010), and Amazon for e-commerce[2].

Baseline Analysis: Criteria of selection We have selected the papers which are based on the idea of using sessions for recommendation systems. Many popular papers that are based on sequential recommendation (like (Kang and McAuley 2018)) do not qualify the criteria and are not included. This work includes STAMP (Liu et al. 2018), NARM (Li et al. 2017), GRU4Rec (Hidasi et al. 2015), (Hidasi and Karatzoglou 2018), HRNN (Quadrana et al. 2017), Cd-HRNN (Dabral, Singh, and Onoe 2023), Tr4Rec (de Souza Pereira Moreira et al. 2021) models as time-driven session-based methods as baseline.

In general, the dataset is sorted in increasing order of $user\_Id$ and then in increasing order of time-stamp. In a few of the SBRS models where $user\_Id$ is not considered, the dataset is arranged only in increasing order of $time\_stamp$.

To evaluate the comparative performance of the proposed methodology, we plugged in content-driven, time-driven sessions to multiple existing methods. With the experiments, our method can address the following research questions.

- **RQ1:** Can homogeneous sessions be applied to existing session-based models? How do they perform?
- **RQ2:** Can the method work with non-Movie datasets?
- **RQ3:** Can the homogeneous session be prioritized over time-driven session?

[2]https://www.kaggle.com/datasets/mkechinov/ecommerce-behavior-data-from-multi-category-store : April

Experiments

In this work, we have performed multiple experiments considering existing models of time-driven session-based methods. The experimental settings are kept intact for comparing TS and CD performance on recommendation tasks. The testing data is the same in both cases, and the only thing that differs is the length of sessions represented by $session\_Id$. The application of the proposed method on existing models (considered as baselines) shows the adaptability of the proposed method. Content-driven sessions outperform the existing time-based sessions tested on multiple SBRS models (Table 2) across multiple datasets to answer the $RQ1$.

To answer the $RQ2$, it is true that movie/video recommendation is the favorable domain for this research because of the easy availability of all the required attributes for representing the items in feature space. The best case of applying the proposed method is in the presence of attributes like $user\_Id$, $item\_Id$, $time\_Stamp$, $Item\_Description$. However, it is less common to find attributes like $item\_description$, to be available in every domain of recsys. Therefore, we advise employing LLM to acquire the description of the items using item name (Acharya, Singh, and Onoe 2023). In this work, we acquired the genre of song through LLM for LastFM dataset.

In order to show the application of the proposed method for additional domains, we applied it to the Book, E-commerce, and Music domains. In our experiments, we realized that the evaluation mechanism of video/ music recommendation might not be the same as e-commerce. As in e-commerce, users buy the product mostly on a need basis, and a pattern manifests mostly a need rather than a general behavior. For instance, if the user needs a mobile phone s/he will search for the phone in the electronics and related category, whereas the category *electronics* in the interaction will disappear when they complete the purchase and may shift towards completing other needs like clothing. We have observed that most of the recommendation models are evaluating the performance on a few of the last interactions in time-ordered data. In the case of e-commerce, there is a high chance that the interactions (need) covered during testing are completely different than the interactions covered in training. Hence, evaluation methods are best when they evaluate the model in the contemporary period. eg. evaluating the last few interactions of each session. However, in the case of Movies/ Music the behavior of the user sustains, so even the last interactions of the data could represent the best distribution of the overall user behavior.

$RQ3$ is about the relative performance of the proposed method. As shown in Figure 5, it can be realized that CD sessions cover a larger switching time. These figures show the average time difference between any two interactions in a user session in both cases (TS, CD; where y-axis shows the number of sessions). In the case of CD, skewness towards smaller time implies that several users change their preferences in less time, however, it is still quite larger than time-driven on all the datasets. In the case of CD-SBRS empirical performance (Table 2) supports the claim that CD sessions outperform TS sessions in capturing personalized engagement and overall recommendation performance.

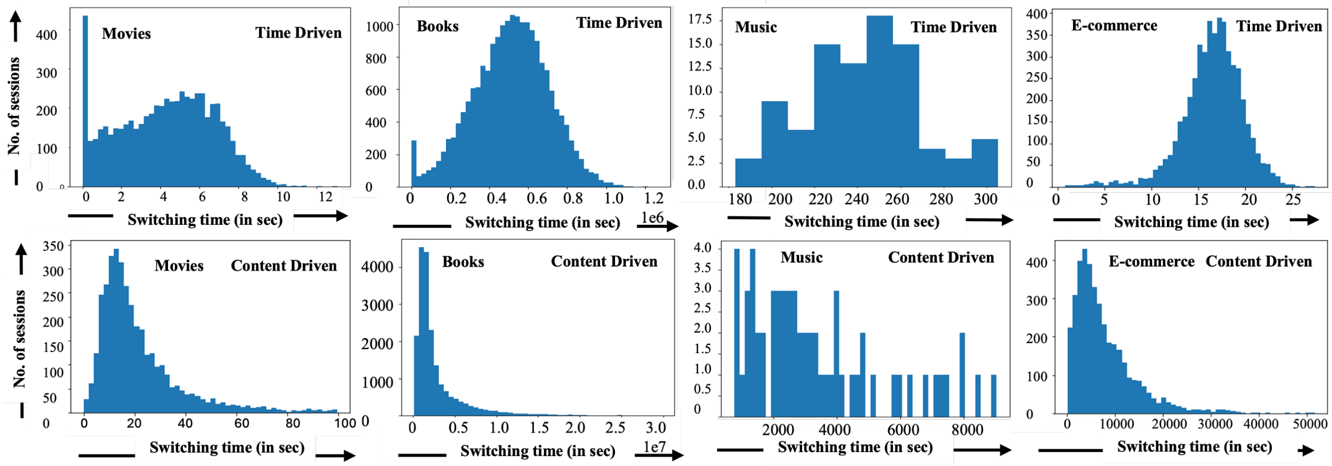


Figure 5: Time-Driven (row-1), Content-Driven (row-2) sessions analysis

Diversity and Practicality Analysis

The proposed CD recommendation model identifies the user's interest as the dominating category. When observed interaction is found converging towards one particular category, it considers all such interactions in the same session. On the contrary, TS sessions may have multiple dominating categories in the same session (because of considering the time). One point to note here is that data sequence is identically same in both cases it's just the boundary of the session that varies. Therefore, the proposed method helps in converging toward user's choice by controlling the unintended exploration. In particular, a category covers $10\%(370/3706)$ of items in video, $14.3\%(44531/311721)$ books in book data, $0.7\%(725/102998)$ in e-commerce, and $0.079\%(87/110117)$ in LastFM music dataset. These statistics show a larger coverage of items in each session after deciding the category in CD sessions. Item's *category* facilitates exploration in music and e-commerce domains.

It has been realized that the same training data with different $session\_id$ can easily replace the existing time-based $session\_id$ and can be considered for training to serve in user products. It is worth mentioning here that the difference between time-driven and content-driven sessions affects only the training of the model and when a model is in the product, it performs the task of next item prediction. There is no functional difference between both the cases at testing time.

Results and Discussion

The proposed method of session creation is applied on popular session-based recommendation methods such as STAMP, NARM, GRU4Rec, HRNN, CD-HRNN, and Tr4Rec. The comparative performance can be observed in Table 2. We can realize the improvement in the recommendation performance on all four datasets across all baselines. In the movie domain, IMDB data is considered for the movie plot, and genres were acquired through LLM for LastFM songs and a total of *1255* unique genres were considered to tag the song tracks (using Gemini Pro).

Comparison: We observed that content-driven (CD) session performs better than time-driven (TS) session because it fulfills the motivation of SBRS more precisely. CD session makes sure that a session is dominated by a single category of items, which helps the recommendation engine to learn the category of the item in which the user is most interested. This way of creating sessions gives training to the model in such a way that it identifies the dominating category in the sequences. Hence, at the inference time model stands at a better state for identifying the category of the item that is likely to be clicked next. As shown in Table 1, columns 2 and 3 represent the total number of cases (in %) in TS, CD scenarios, when the dominating item is present at the last interaction of the session. Columns 4 and 5 represent the case when the dominating item is present at the first interaction of next session. Whereas CD session creation algorithm ensures that the dominating item can not come as the first item in the upcoming session. Columns 6 and 7 represent the cases when dominating item is present in the next session. We should know a few corner cases while reading the Table 1 eg. The upcoming session may belong to a different user. The session could be the last session with no dominating category. As CD-HRNN works in the presence of item description, its results are available only for Movie and book datasets (Table 2). One thing to note here is that CD-HRNN considers the description along with user-item interaction attributes even then CD sessions outperform on recommendation metrics, which rules out the case of improvement in CD sessions because of content description. In the experiments, we found that STAMP model originally works in the absence of $user\_id$. In our case, we eliminated $user\_id$ from the datasets to apply the model, but performance was found lower on both time-driven and content-driven cases with respect to other models. In case of HRNN, CD-HRNN, when we followed the strategy given in (Quadrana et al. 2017), we got the performance of HRNN as (Recall@20) 0.43, 0.51 and of CD-HRNN 0.50, 0.55 on TS, CD respectively. But we noticed that most of the user sessions got eliminated

Dataset	Eval Metrices	Type	@K	**STAMP**	**NARM**	**GRU4Rec**	**HRNN**	**Cd-HRNN**	**Tr4Rec**
Movielens	**Recall**	TS-SBRS	@20	0.0662	0.2616	0.2733	0.2769	0.2779	0.3142
			@10	0.0394	0.1696	0.2026	0.2016	0.1973	0.2176
		CD-SBRS	@20	**0.0761**	**0.3121**	**0.2835**	**0.2847**	**0.283**	**0.3231**
			@10	**0.0507**	**0.2121**	**0.2113**	**0.2089**	**0.2017**	**0.2181**
	MRR/ NDCG	TS-SBRS	@20	0.0168	0.0683	0.0974	0.0925	0.0884	N: 0.0942
			@10	0.0146	0.0617	0.0925	0.0873	0.0829	N: 0.0877
		CD-SBRS	@20	**0.0198**	**0.0884**	**0.0999**	**0.0949**	**0.091**	**N: 0.1371**
			@10	**0.0197**	**0.0813**	**0.0949**	**0.0896**	**0.085**	**N: 0.1107**
GoodRBook	**Recall**	TS-SBRS	@20	0.0583	0.1113	0.1731	0.0771	0.0854	0.0466
			@10	0.0259	0.0848	0.1483	0.0642	0.0725	0.0333
		CD-SBRS	@20	**0.0639**	**0.1305**	**0.1817**	**0.0961**	**0.0961**	**0.0556**
			@10	**0.0281**	**0.1037**	**0.1540**	**0.0870**	**0.0833**	**0.0392**
	MRR/ NDCG	TS-SBRS	@20	**0.0151**	0.0480	0.1032	0.0574	0.0463	N: 0.0218
			@10	**0.0117**	0.0465	0.1015	0.0498	0.0415	N: 0.0185
		CD-SBRS	@20	0.0122	**0.0685**	**0.1075**	**0.0722**	**0.0569**	**N: 0.0245**
			@10	0.0111	**0.0663**	**0.1064**	**0.0691**	**0.0561**	**N: 0.0203**
LastFM	**Recall**	TS-SBRS	@20	0.0105	0.3722	**0.5045**	0.3463	N/A	0.0689
			@10	0.0087	0.3194	0.4530	0.3285	N/A	0.0538
		CD-SBRS	@20	**0.0112**	**0.4255**	0.5043	**0.3583**	N/A	**0.1060**
			@10	**0.0095**	**0.3650**	**0.4543**	**0.3364**	N/A	**0.0866**
	MRR/ NDCG	TS-SBRS	@20	0.0044	0.2068	0.2905	0.2447	N/A	N: 0.0354
			@10	0.0040	0.2026	0.2868	0.2442	N/A	N: 0.0316
		CD-SBRS	@20	**0.0062**	**0.2404**	**0.2944**	**0.2515**	N/A	**N: 0.0559**
			@10	**0.0061**	**0.2491**	**0.2909**	**0.2499**	N/A	**N: 0.0510**
Amazon	**Recall**	TS-SBRS	@20	0.0948	0.3464	0.4034	0.3987	N/A	0.0399
			@10	0.0791	0.2951	0.3584	0.3356	N/A	0.0238
		CD-SBRS	@20	**0.0997**	**0.4702**	**0.5466**	**0.427**	N/A	**0.0479**
			@10	**0.08513**	**0.3984**	**0.4787**	**0.3685**	N/A	**0.0314**
	MRR/ NDCG	TS-SBRS	@20	**0.0562**	0.1856	0.2594	0.2316	N/A	N:0.0165
			@10	**0.0457**	0.1855	0.2563	0.1818	N/A	N:0.0125
		CD-SBRS	@20	0.0529	**0.2464**	**0.3316**	**0.246**	N/A	**N:0.0211**
			@10	0.0432	**0.2465**	**0.3269**	**0.1926**	N/A	**N:0.0169**

Table 2: Comparative performance of the proposed CD-SBRS against TS-SBRS on multiple datasets.

from the testing and so many of the users were not evaluated. We adapted a way of appending the single-item testing session with the previous session to include them in the evaluation. The process helped to include all the users in testing and making it a fairly comparable setting but overall performance got reduced on both TS, CD as shown in Table 2. Tr4Rec performed best on Movielens. Tr4Rec is evaluated on its original metric (Recall, NDCG). N : in the table (last column) refers NDCG otherwise it's MRR.

Conclusion

In this paper, we propose a novel method of creating a session leveraging content and named it as content-driven session. We proposed that a content-driven session captures the user behavior more precisely therefore it improves the recommendation performance. Content-driven sessions are created on the basis of homogeneity in the content. We proposed to cluster the items in the feature space to know a) Total number of categories available b) The category of each item. The category of each item is leveraged to decide the boundary of each session. The proposed method creates new $session\_Id$ in the existing dataset and keeps the rest of the experimental setting intact. We evaluate the performance of the proposed method on four benchmark datasets on 6 session-based recommendation models. The proposed method outperformed the time-driven session on all the baseline models. This way, content-driven method a). Helps users in continuing the same experience after returning back b). Provides a solution to the problem of deciding threshold θ in TS-SBRS c). Better captures the user interaction pattern and improves the recommendation performance. The proposed method performs clustering to create labels. In domains (such as e-commerce) where the category is already known, the clustering step can be skipped.

References

Acharya, A.; Singh, B.; and Onoe, N. 2023. LLM Based Generation of Item-Description for Recommendation System. In *Proceedings of the 17th ACM Conference on Recommender Systems*.

Bernardis, C.; Dacrema, M. F.; Maurera, F. B. P.; Quadrana, M.; Scriminaci, M.; and Cremonesi, P. 2022. From Data Analysis to Intent-Based Recommendation: An Industrial Case Study in the Video Domain. *IEEE Access*, 10: 14779–14796.

Celma, O. 2010. *Music Recommendation and Discovery in the Long Tail*. Springer.

Chen, H.; Shi, S.; Li, Y.; and Zhang, Y. 2021. Neural collaborative reasoning. In *Proceedings of the Web Conference 2021*, 1516–1527.

Dabral, S.; Singh, B.; and Onoe, N. 2023. Cd-HRNN: Content-Driven HRNN to Improve Session-Based Recommendation System. In *2023 International Joint Conference on Neural Networks (IJCNN)*, 1–8. IEEE.

de Souza Pereira Moreira, G.; Rabhi, S.; Lee, J. M.; Ak, R.; and Oldridge, E. 2021. Transformers4rec: Bridging the gap between nlp and sequential/session-based recommendation. In *Proceedings of the 15th ACM Conference on Recommender Systems*, 143–153.

DeCoste, D. 2006. Collaborative prediction using ensembles of maximum margin matrix factorizations. In *Proceedings of the 23rd international conference on Machine learning*, 249–256.

Devooght, R.; and Bersini, H. 2017. Long and short-term recommendations with recurrent neural networks. In *Proceedings of the 25th conference on user modeling, adaptation and personalization*, 13–21.

Friston, K.; and Kiebel, S. 2009. Predictive coding under the free-energy principle. *Philosophical transactions of the Royal Society B: Biological sciences*, 364(1521): 1211–1221.

Friston, K.; Schwartenbeck, P.; FitzGerald, T.; Moutoussis, M.; Behrens, T.; and Dolan, R. J. 2014. The anatomy of choice: dopamine and decision-making. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 369(1655): 20130481.

Guo, H.; Tang, R.; Ye, Y.; Li, Z.; and He, X. 2017. DeepFM: a factorization-machine based neural network for CTR prediction. *arXiv preprint arXiv:1703.04247*.

Harper, F. M.; and Konstan, J. A. 2015. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)*, 5(4): 1–19.

Hidasi, B.; and Karatzoglou, A. 2018. Recurrent neural networks with top-k gains for session-based recommendations. In *Proceedings of the 27th ACM international conference on information and knowledge management*, 843–852.

Hidasi, B.; Karatzoglou, A.; Baltrunas, L.; and Tikk, D. 2015. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939*.

Kang, W.-C.; and McAuley, J. 2018. Self-attentive sequential recommendation. In *2018 IEEE international conference on data mining (ICDM)*, 197–206. IEEE.

Li, J.; Ren, P.; Chen, Z.; Ren, Z.; Lian, T.; and Ma, J. 2017. Neural attentive session-based recommendation. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, 1419–1428.

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*, 1831–1839.

Marlin, B.; and Zemel, R. S. 2004. The multiple multiplicative factor model for collaborative filtering. In *Proceedings of the twenty-first international conference on Machine learning*, 73.

Mondal, P.; Kapoor, P.; Singh, S.; Saha, S.; Onoe, N.; and Singh, B. 2023. Impulsion of Movie's Content-Based Factors in Multi-modal Movie Recommendation System. In *International Conference on Neural Information Processing*, 230–242. Springer.

Onumanyi, A. J.; Molokomme, D. N.; Isaac, S. J.; and Abu-Mahfouz, A. M. 2022. AutoElbow: An automatic elbow detection method for estimating the number of clusters in a dataset. *Applied Sciences*, 12(15): 7515.

Pazzani, M. J. 1999. A framework for collaborative, content-based and demographic filtering. *Artificial intelligence review*, 13: 393–408.

Quadrana, M.; Karatzoglou, A.; Hidasi, B.; and Cremonesi, P. 2017. Personalizing session-based recommendations with hierarchical recurrent neural networks. In *proceedings of the Eleventh ACM Conference on Recommender Systems*, 130–137.

Su, X.; Greiner, R.; Khoshgoftaar, T. M.; and Zhu, X. 2007. Hybrid collaborative filtering algorithms using a mixture of experts. In *IEEE/WIC/ACM International Conference on Web Intelligence (WI'07)*, 645–649. IEEE.

Thorndike, R. L. 1953. Who belongs in the family? *Psychometrika*, 18(4): 267–276.

Wan, M.; and McAuley, J. 2018. Item recommendation on monotonic behavior chains. In *Proceedings of the 12th ACM conference on recommender systems*, 86–94.

Wan, M.; Misra, R.; Nakashole, N.; and McAuley, J. 2019. Fine-grained spoiler detection from large-scale review corpora. *arXiv preprint arXiv:1905.13416*.

Wang, J.; De Vries, A. P.; and Reinders, M. J. 2008. Unified relevance models for rating prediction in collaborative filtering. *ACM Transactions on Information Systems (TOIS)*, 26(3): 1–42.